



PROCEEDINGS

ICTIS '24



INTERNATIONAL CONFERENCE ON
**INNOVATIONS IN COMPUTING
TECHNOLOGIES AND INFORMATION
SCIENCES (ICTIS)**

29TH & 30TH APRIL, 2024.



Preface

With immense pleasure, I am honored to present the Proceedings for the 1st International Conference on Innovations in Computing Technologies and Information Sciences (ICTIS) 2024 as the Conference Chief Organizer. Hosted by the esteemed UET Peshawar, this conference marks a significant milestone in the realm of computing and information sciences.

The primary objective of this conference is to provide a dynamic platform for the convergence of scholars, researchers, academicians, and industry professionals from around the globe. Our focus on cutting-edge innovations in computing technologies and information sciences aims to foster interdisciplinary collaborations and propel advancements in these rapidly evolving fields.

The conference themes span a wide spectrum, including but not limited to Artificial Intelligence, Machine Learning, Data Science, Cybersecurity, Internet of Things (IoT), and Blockchain Technology. By facilitating dialogue and knowledge exchange on these topics, ICTIS 2024 seeks to catalyze innovation and drive positive impact across academia, industry, and society.

UET Peshawar, renowned for its academic excellence and commitment to research, serves as an ideal host for this prestigious event. With dedicated qualified faculty, and enthusiastic students, UET Peshawar continues to be a beacon of innovation in the region.

I extend my heartfelt gratitude to all the keynote speakers, session chairs, and authors whose invaluable contributions have enriched the conference proceedings. I would also like to express sincere appreciation to our technical and financial sponsors for their generous support, without which this event would not have been possible.

Lastly, I commend the diligent efforts of all my colleagues on the organizing committees for their unwavering commitment and dedication to the successful execution of ICTIS 2024. Together, we have created a platform that not only fosters intellectual discourse but also paves the way for groundbreaking innovations in computing technologies and information sciences.

Thank you to everyone who has contributed to making ICTIS 2024 a resounding success. Your passion and dedication inspire us to continue pushing the boundaries of knowledge and innovation.



Dr. Nasru Minallah
Chief Organizer

Organizing Committee

Honorary Chairs

Prof. Qaiser Ali, Vice Chancellor, University of Engineering and Technology, Peshawar, Pakistan

Prof. Syed Waqar Shah, Dean, Faculty of Electrical and Computer Engineering, University of Engineering and Technology, Peshawar, Pakistan

Program Chair

Prof. Laiq Hasan, University of Engineering and Technology, Peshawar, Pakistan

Organizing Chair

Dr. Nasru Minallah, University of Engineering and Technology, Peshawar, Pakistan

Technical Chairs

Dr. Nasir Ahmad, University of Engineering and Technology, Peshawar, Pakistan

Dr. Sohail Imran, IQRA National University, Peshawar, Pakistan

Dr. Khurram Khattak, University of Engineering and Technology, Peshawar, Pakistan

Dr. Safdar Marwat, University of Engineering and Technology, Peshawar, Pakistan

Dr. Sheeraz Ahmed, IQRA National University, Peshawar, Pakistan

Dr. Ihsan Ul Haq, University of Engineering and Technology, Peshawar, Pakistan

Dr. Arbab Masood, University of Engineering and Technology, Peshawar, Pakistan

Dr. Amjad Ali, University of Engineering and Technology, Peshawar, Pakistan

Dr. Amaad Khalil, University of Engineering and Technology, Peshawar, Pakistan

Dr. Muniba Ashfaq, University of Engineering and Technology, Peshawar, Pakistan

Dr. Yasir Afridi, University of Engineering and Technology, Peshawar, Pakistan

Dr. Abeer Irfan, University of Engineering and Technology, Peshawar, Pakistan

Dr. Athar Javed Sethi, University of Engineering and Technology, Peshawar, Pakistan

Dr. Madiha Sher, University of Engineering and Technology, Peshawar, Pakistan

Dr. Rehmat Ullah Khattak, University of Engineering and Technology, Peshawar, Pakistan

Engr. Sumayyea Salahuddin, University of Engineering and Technology, Peshawar, Pakistan

Industry Chairs

Engr. Mohsin Tariq, Machine Learning Engineer, Turing

Engr. Muhammad Zeeshan, Web & App Developer, NCBC UET Peshawar

Mr. Tufail Ahmad, Blockchain Developer, NCBC UET Peshawar

Keynote Speakers Sessions

Session1:

Title: Landscape of Developer Ecosystem in Pakistan
Keynote Speaker: Huzaifa Habib, Community Manager at Google

Session2:

Title: Flutter: Your Bridge to Seamless Multi-Platform Apps
Keynote Speaker: Muhammad Zeeshan, Web & App Developer

Session3:

Title: Game Development: Pioneering Technological Innovation
Keynote Speaker: Engr. Hameed Ullah Jan, CEO AptechMedia

Session4:

Title: Attracting Top Talent in the AI Era: "Insights from Afiniti "
Keynote Speaker: Imran Khan, Digital Transformation Professional

CPD Sessions:

Session1:

Title: Use of GIS in Water resources Engineering: Pitfalls of Auto-delineation
Resource Person: Dr. Asif Khan, Climate Change Expert & Member of European Geophysical Union

Session2:

Title: Energy saving opportunities through Energy Audits.
Resource Person: Mr. Muhammad Ali, Senior Assistant Manager, CIBEA Pakistan



International Conference on Innovations in Computing Technologies and Information Sciences (ICTIS'24)

APRIL 29-30, 2024 | UET PESHAWAR

Contents

Paper ID: 01.....	4
Classification of Medical Images Through Convolutional Neural Network Modification Method.....	4
Paper ID: 02.....	10
On Evaluation of discrete RL agents for Traffic Scheduling and Trajectory Optimization of UAV-based IoT Network with multiple RIS unit.....	10
Paper ID: 03.....	16
Machine Learning Based Estimation of End Effector position in Three Dimension Robotic.....	16
Paper ID: 04.....	22
Hybrid Approach to solve Thermal Power plants Fuel cost Optimization.....	22
using Ant Lion Optimizer with Newton based local search technique.....	22
Paper ID: 05.....	27
Smart Fire Safety: Real-Time Segmentation and Alerts using Deep Learning.....	27
Paper ID: 06.....	33
Modeling of Post-Myocardial Infarction and Its Solution Through Artificial Neural Network.....	33
Paper ID: 07.....	39
Honey Adulteration Detection through Hyperspectral Imaging And Machine Learning.....	39
Paper ID: 08.....	43
Optimized Coverage and Capacity Planning of WiFi Network based on Radio Frequency Modeling & Propagation Simulation.....	43
Paper ID: 09.....	49
Breast Masses Detection Using YOLOv8.....	49
Paper ID: 10.....	55
AI-driven Prediction of Electricity Production and Consumption in Micro-Hydropower Plant.....	55
Paper ID: 11.....	63
Meta-Space: Pioneering Education in the Metaverse.....	63
Paper ID: 12.....	68
Use of Artificial Intelligence in Ethereum Forecasting: The Deep Learning Models RNN And CNN With Ensemble Averaging Technique.....	68
Paper ID: 13.....	74



International Conference on Innovations in Computing Technologies and Information Sciences (ICTIS'24)

APRIL 29-30, 2024 | UET PESHAWAR

Model-Based vs Model Free Deep Reinforcement Learning Models for Cancer Treatment: A Critical Analysis with a Solution DRL Model.....	74
Paper ID: 14.....	80
Investigating Threats to ICS and SCADA Systems via Honeypot Data Analysis and SIEM	80
Paper ID: 15.....	86
SkinScan: Cutting-edge AI-Powered Skin Cancer Classification App for Early Diagnosis and Prevention ..	86
Paper ID: 16.....	93
Combatting Illegal Logging with AI Powered IoT Devices for Forest Monitoring	93
Paper ID: 17.....	99
Exploring Hardware Security: Ring Oscillator Based PUF Implementation on Xilinx 7 Series FPGAs	99
Paper ID: 18.....	103
GAMEBRAINS: NPCs Intelligence Using Neural Network Brains.....	103
Paper ID: 19.....	109
Artificial Intelligence Based Seed Germination, Prediction, and Quality Assessment	109
Paper ID: 20.....	114
Drinking Water Monitoring: Computer Vision Kit for Early E.coli Detection.....	114
Paper ID: 21.....	120
Low Cost Smart Metering Using Deep Learning	120
Paper ID: 22.....	126
Visually: Assisting the Visually Impaired People Through AI-Assisted Mobility	126
Paper ID: 23.....	132
An Aggregated Approach Towards NILM on ACS-F2 Using Machine Learning.....	132
Paper ID: 24.....	138
Relevance Classification of Flood-Related Tweets using XLNET Deep Learning Model	138
Paper ID: 25.....	142
Exploring Political Emotions: Sentiment Analysis of Urdu Tweets	142
Paper ID: 26.....	152
Stress Detection and Prediction Using CNNs from Electrocardiogram Signals	152
Paper ID: 27.....	158
Beyond CNNs: Encoded Context for Image Inpainting with LSTMs and Pixel CNNs.....	158



International Conference on Innovations in Computing Technologies and Information Sciences (ICTIS'24)

APRIL 29-30, 2024 | UET PESHAWAR

Paper ID: 28.....	167
Comparative Study On Harassment Stance Detection Using Social Media Texts Through Deep Learning Methods.....	167
Paper ID: 29.....	173
Delineation of Crop Fields in Satellite Imagery Using SAM	173
Paper ID: 30.....	179
Sherpa: Implementing a Hybrid Recommendation System for Next Gen Tourist Experience	179
Paper ID: 31.....	183
PERFORMANCE ANALYSIS OF A HYBRID RECOMMENDER SYSTEM.....	183
Paper ID: 32.....	191
NEUROSCAN: Revolutionizing Brain Tumor Detection using Vision-Transformer.....	191
Paper ID: 33.....	197
Detection of Holes in Point Clouds using Statistical Technique.....	197
Paper ID: 34.....	202
CUSTOMER REVIEWS ANALYSIS PLATFORM BY CORRELATING SENTIMENT ANALYSIS AND TEXT CLUSTERING	202
Paper ID: 35.....	213
A Deep Learning based Mobile Enabled Application for Wheat Disease Diagnosis	213
Paper ID: 36.....	219
Home Automation Using Internet of Things and Machine Learning.....	219
Paper ID: 37	222
Deep Learning based Skin Lesion Segmentation and Classification	222
Paper ID: 38.....	228
Stock Market Analysis and Prediction.....	228
Paper ID: 39.....	234
Land Subsidence in Quetta Valley: A Study using PyGMTSAR Library and SBAS Technique	234

Classification of Medical Images Through Convolutional Neural Network Modification Method

Syed Kashif Badshah

Department of Basic Sciences and Islamiat,
University of Engineering and Technology,
Peshawar, Pakistan
skashif.bsi@uetpeshawar.edu.pk

Noor Badshah

Department of Basic Sciences and Islamiat
University of Engineering and Technology,
Peshawar, Pakistan
noorbadshah@uetpeshawar.edu.pk

Abstract— The COVID-19 positive, tuberculosis and pneumonia, share the trait of being able to be identified using radiological investigations, such as Chest X-Ray (CXR) images. This paper aims to distinguish between four classes, including tuberculosis (TB), COVID-19 positive, healthy, and pneumonia using CXR images. Many deep-learning models such as a Convolutional Neural Network (CNN) have been developed for the Classification of CXR images. Deep learning-based models such as CNN offer significant advantages over traditional methods in the classification of diseases like TB, COVID-19, pneumonia, and healthy states. They provide higher accuracy, automation, early detection, reduced subjectivity and resource efficiency, ultimately leading to improved patient care and outcomes. However, well liked CNNs are massive models that require a lot of data to achieve optimal accuracy. In this paper, we propose a new CNN model that can be used to distinguish between different classes of CXR images. This model proves to be effective in classifying between different diseases such as pneumonia, COVID-19, and tuberculosis. This study has used 6326 CXR images dataset containing COVID-19 positive, tuberculosis, pneumonia and has normal images. In this dataset 80% of the CXR images are taken for the training purpose and 20% are taken for the validation purpose, of the proposed CNN model. The proposed CNN modified model with parameter adjustment as well as using of categorical cross-entropy as a loss function obtains the highest classification accuracy of 98.51% with a precision, recall and F1 score of 0.90, 0.89 and 0.90 respectively.

Keywords: Image Classification; Fuzzy Membership; VGG-19 Modified Model;

I. INTRODUCTION

The COVID-19 has set historical records on a global scale. More than 116 million confirmed cases and more than 2.5 million fatalities had been recorded by the World Health Organization (WHO) as of 8 March 2021. The SARS-CoV2 virus, which causes the infectious disease COVID-19, is easily spread by contact and the air and has a serious impact on the lungs of those who contract it. The COVID-19 virus can develop consequences, including pneumonia, as well as other symptoms that may be mistaken for those of other infections [1].

In contrast, the infectious disease tuberculosis, which is caused by *Mycobacterium tuberculosis*, also results in antibiotic resistance and the death of tissue in various sections of the body, primarily the lungs. The WHO estimates that tuberculosis kills over 1.5 million people worldwide each year, making it the most lethal infectious

disease. An estimated 10 million people contracted TB in 2019 alone [1].

As a result, COVID-19, pneumonia and tuberculosis all have the ability to be identified by radiological procedures like CXR images. Prior to the development of deep learning (DL) frameworks, feature extraction and classification methods were used to classify medical images.

Medical image classification through Convolutional Neural Networks (CNNs) is an application of deep learning that has revolutionized the field of medical imaging. With the increase in availability and quality of medical imaging data, there has been a growing demand for automated and accurate image classification tools that can assist clinicians in their diagnoses and treatment decisions. CNN is a neural network that are designed specifically for image analysis tasks. They are made up of several layers of interlinked nodes able to recognize and extract features from pictures. In medical image classification, CNN is typically trained on large datasets of labeled images, where the labels represent the different diagnostic categories or pathologies of interest. The process of medical image classification through CNN involves several steps. First, the input medical images are preprocessed to ensure that they are in a suitable format for analysis. Next, the CNN is trained on a dataset of labeled medical images, using a process called back propagation to adjust the weights of the network to optimize its performance on the training data. Once the CNN has been trained, it can be used to classify new medical images by passing them through the network and observing the output of the final layer, which represents the predicted class label.

CNN-based models have been successfully applied and used to create dependable, quick and accurate detection methods against COVID-19 and other respiratory diseases, demonstrating their potential for transforming medical diagnostic procedures. This is due to the models' deep learning capabilities and intricate architectures. [2–6, 11]. This is why the current study's objective is to determine whether it is possible to distinguish between healthy patients and those with COVID-19 positive, pneumonia, and TB using early automated classification of CXR images.

To determine if a patient is normal or has a lung illness, we have built a deep transfer learning pipeline called the VGG-19 modified model. The VGG-19 are ideally modified in the suggested network. Transfer learning techniques are implemented using pre-trained networks on the VGG-19 model. We put our suggested network to the test for four-

class classification problems (TB, pneumonia, healthy and COVID-19 positive).

To the best of our current understanding, this work is an important attempt to investigate the viability and effectiveness of using early automatic identification and differentiation methods with a particular focus on differentiating between people who are COVID-19 positive, suffering from pneumonia or tuberculosis, and those who are considered healthy using only CXR images as the primary diagnostic modality. The suggested model has demonstrated robust and enhanced performance over the state-of-the-art methods for the classification of lung disorders in all of our datasets and has been able to perform optimally in a variety of multi-class tasks. Moreover, it has demonstrated its versatility and efficacy in intricate diagnostic circumstances by continuously outperforming a wide range of multi-class tasks. This work represents a major step forward in using computational approaches to support early and precise identification of respiratory diseases, especially when it comes to using CXR imaging to differentiate COVID-19 from other related disorders.

The following are this paper's major contributions:

1. The building of a new convolutional neural network, named VGG-19 modified model, for robustness and more precise classification.
2. Evaluating the VGG-19 modified model's precision and robustness on CXR image datasets with many classes of labels (TB, pneumonia, COVID-19 positive, and healthy).
3. The comparison of VGG-19 modified model with the other state of the art architectures such as VGG-16, DenseNet-121, and ResNet-50.

The pattern of the paper is organized as follows:

- Section I introduce the specified title.
- The associated works are briefly summarized in Section II.
- The preprocessing approach and the suggested methodology are presented in Section III and IV respectively.
- The numerical outcomes of our methodology on CXR dataset are shown in Section V.
- Section VI presents the final conclusion.

II. LITERATURE REVIEW

The classification of CXR images has been the subject of extensive research in recent years. A brief summary of these research initiatives is provided as below:

Jaiswal et al. [7] introduced a Mask-Region-based CNN model, aimed at automating the classification process of pneumonia cases utilizing CXR images. Bharati et al. [8] concentrated on making use of a hybrid deep learning system that blends several modern techniques, such as data augmentation approaches, spatial transformer networks (STNs) and Convolutional Neural Networks (CNNs) with VGG architecture. Bharati et al. sought to create a complete framework that may improve the resilience and accuracy of

classification tasks by combining these disparate components, especially when it came to the diagnosis of medical disorders from CXR pictures.

CNNs are well known for their ability to extract features and recognize patterns; this helped to provide the groundwork for the proposed system's learning capabilities. Additionally, the utilization of VGG architecture, known for its depth and effectiveness in image classification tasks, further bolstered the model's performance. To address potential spatial variations and distortions within the CXR images, Bharati et al. integrated a spatial transformer network (STN), enabling the network to dynamically adapt and rectify spatial transformations to enhance enhancing its adaptability and resilience to image variations. Furthermore, to augment the training dataset and mitigate overfitting, data augmentation techniques were employed, facilitating the generation of diverse training samples by applying transformations such as rotations, translations, and scaling. Through extensive experimentation and training on the National Institutes of Health (NIH) CXR dataset, Bharati et al. reported notable results, achieving an accuracy rate of 73%. They trained their network in NIH CXR dataset and achieved 73% accuracy.

Pereira et al. [9] introduced the RYDLS-20 network model, representing a significant advancement in the field of medical image analysis. The primary focus of their study was on the diagnosis of COVID-19 utilizing deep learning techniques. The RYDLS-20 model, meticulously designed and optimized by Pereira et al., achieved an impressive F1 value of 89%, underscoring its efficacy and reliability in accurately identifying COVID-19 cases. Notably, the dataset utilized in their study exhibited a considerable imbalance, consisting of 2000 healthy cases compared with only 180 patients afflicted by COVID-19. Without using any cross-validation steps, the RYDLS-20 model's classification performance was shown, demonstrating the model's capacity to attain high accuracy even in the absence of such validation processes. This omission may cause questions about how well the model generalizes to new data, but the stated F1 value indicates that there is a good chance that it will be useful in everyday life.

Song et al. [4] built a COVID-19 patient identification system for computed tomography (CT) scans called Deep-Pneumonia that is based on deep learning. After manually segmenting the lung area using a DL network, they classified healthy or COVID-19 patients. They combined ResNet50 with a feature pyramid network (FPN) and an attention module to create their own network, which they called DRE-Net. The investigation's primary strengths were highlighted by the study, which used multi-vendor datasets from three different hospitals and showed impressive sensitivity (95%) and specificity (96%), as well as a quick diagnostic time of only 30 seconds per patient. Several drawbacks can be identified in this study. Firstly, the reliance on semi-automatic lung segmentation raises concerns about the consistency and accuracy of the segmentation process, potentially introducing variability in the analysis. Secondly, the classification of datasets solely based on CT images without stratification according to

factors such as advanced age, underlying diseases, or the presence of pleural effusions could lead to biased results.

Chen et al. [10] applied 46, 095 anonymised images of 106 hospitalized patients at Renmin Hospital of the University of Wuhan to train their deep network. Of them, 51 patients had COVID-19 pneumonia confirmed by a laboratory, and the remaining 55 patients had various illnesses. The lungs were divided into sections, and any scar tissue was located using a U-net++ network. The time difference between the radiologist and the model was compared using a two-tailed paired Student's t-test with a significance level of 0.05. The main strengths of this work are the huge and equally distributed training dataset, the good classification accuracy (above 95%), and the use of three experienced radiologists who considered inter-observer variability to obtain the ground truth.

Ozturk et al. [11] leveraged the Dark CovidNet model as a tool to assist radiologists and medical professionals in diagnosing COVID-19. This model demonstrated a remarkable accuracy in binary classification, achieving an impressive 98.08% accuracy rate in distinguishing between COVID-19 cases and healthy individuals. Additionally, in more complex multi-class classification scenarios where the model had to differentiate between pneumonia, COVID-19, and healthy cases, it maintained a respectable accuracy of 87.02%.

A brand-new transfer learning-based model for the categorization and detection of pneumonia (both viral and bacterial) was put out by Rahman et al. [12]. To determine which pre-designed CNN architecture had the greatest performance, they suggested comparing the various ones. The results demonstrated that, of all the utilized architectures, DenseNet201 demonstrated remarkable accuracy rates, with a noteworthy 98% accuracy in differentiating between chest X-ray that showed no symptoms and those that showed indicators of pneumonia (bacterial or viral). In addition, when the model was required to distinguish between cases of bacterial, viral, and normal pneumonia, it performed admirably, scoring 93.3%. In this particular subset, the model had a robust accuracy rate of 95% when it came to differentiating between pneumonia caused by bacteria and viruses.

Michail et al. [13] presented the DenResCov-19 deep transfer learning model to identify patients with Pneumonia, TB, COVID-19, or healthy based on CXR images dataset. Their method combines the previous ResNet-50 and DenseNet-121 architectures with an extra layer of convolutional neural network (CNN) building blocks to improve the model's functionality. They were able to leverage the advantages of both networks by combining these topologies, using ResNet's residual connections to mitigate vanishing gradient problems and DenseNet's dense connections for feature reuse. Their network was tested on the CXR image dataset, which included several classifications such as COVID-19, pneumonia, TB, and normal cases. With careful tweaking and training, the network demonstrated an amazing accuracy rate of 96.40%. This significant accuracy highlights the potential of deep transfer learning approaches in the field of medical image

analysis, providing promising paths for the early and accurate diagnosis of a variety of respiratory conditions, including COVID-19, and enabling prompt intervention and treatment plans.

III. DATASET DESCRIPTION

We have collected a large set of CXR images and apply data augmentation techniques to increase data diversity in an attempt to improve classification accuracy. The study employed a dataset consisting of 6326 CXR images that were obtained from the free software platform Kaggle. This dataset contains pictures of a variety of diseases, such as pneumonia, tuberculosis, and COVID-19 positive patients, in addition to typical cases.

A. Preprocessing

Fuzzy set theory and Gaussian kernel-based enhancement are used to improve the performance of CNN-based image classification. Fuzzy set theory is a mathematical framework that deals with uncertainty and imprecision. It allows for the representation of a concept with degrees of membership instead of a binary yes/no value. In the context of image classification, this means that instead of assigning a single label to an image, we can assign multiple labels with different degrees of certainty. This approach can be particularly useful when dealing with images that contain ambiguous or overlapping features.

Fuzzy set theory involves the use of membership functions to assign degrees of membership to elements of a set. Let X be the set of possible image features, and let A be a fuzzy subset of X . The membership function for A is denoted by $\mu_A(x)$, where x is an element of X , and $\mu_A(x)$ is a value between 0 and 1 that represents the degree to which x belongs to A . For each image feature x , we compute the membership function values $\mu_{A_1}(x)$, $\mu_{A_2}(x)$, ..., $\mu_{A_n}(x)$ for each of the fuzzy subsets A_1, A_2, \dots, A_n . Then we apply the Gaussian kernel function to the image to smooth it and reduce noise.

To improve the quality of the images, we experimented with a range of image enhancement methods, such as Histogram Equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE), and Fuzzy Contrast Enhance (FCE). Each of these methods was applied to a single dataset image.

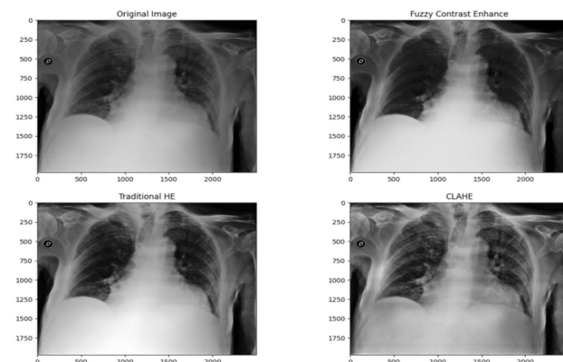


Figure 1: Comparison of different image enhancement techniques

Figure 1 displays the outcomes of various methods. HE and CLAHE have demonstrated impressive results in denoising and improving the images, as illustrated in Figure 1. But it may also possible to see that using CLAHE intensifies the color of the bones, which could have an impact on how well the classification model performs because the rib and sternum bones may be identified by the neural network as the main X-Ray detection features [16].

Thus in our suggested framework, histogram equalization was chosen as the image enhancing method. Our second step is to split the dataset. The dataset of CXR images is divided into two groups. 80% of the CXR images (5061 images) are used for training the proposed model, and 20% (1265 images) are reserved for validating the model. The images in the dataset originally have different sizes (512×512) pixels. To make them uniform and suitable for deep learning, they are resized to a smaller size of 112×112 pixels. This resizing step ensures that all images are in the same format and size, making it easier for the model to analyze them. Resizing techniques' underlying theory is covered in [14].

IV. METHODOLOGY

In this section we will discuss the methodology of the proposed network. We started with the VGG-19 model, which is CNN architecture, designed for image classification. VGG-19 is a well-known CNN architecture with 19 layers, consisting of 16 convolutional layers followed by 3 fully connected layers. The model VGG-19 is known for its effective feature extraction. It uses 16 convolutional layers organized into 5 groups. After each group of convolutional layer, there is a max-pooling layer. These convolutional layers are designed to capture the important features from the input images. They use (3, 3) filters with Rectified Linear Units (ReLU) as activation function. Max-pooling is employed with a (2, 2) kernel and a stride of 2 pixels for downscaling.

We have changed some layers in VGG-19 architecture to adapt it to our X-ray image classification task. One of the convolutional layers in the VGG-19 was replaced with a dropout layer and also skipped one or more than one convolutional layer in each group. Dropout is a regularization technique that helps to prevent overfitting by randomly deactivating a fraction of neurons during training. We added one more group for feature extraction, making a total of six groups, each one followed by a max-pooling layer. These layers are responsible for extracting important features from the input images. In the first layer of our modified model, we used a 2D convolution operation with a (5, 5) kernel size, applying the same padding, followed by max-pooling with a (2, 2) kernel for reducing the spatial dimensions. ReLU activation function and dropout are applied in this layer. We continued to build the model with similar convolutional layers for features extraction. The final layer in the features extraction process uses a 2D convolution with a (3, 3) kernel size, (2, 2) stride, and additional (2, 2) max-pooling which is shown in Figure 2.

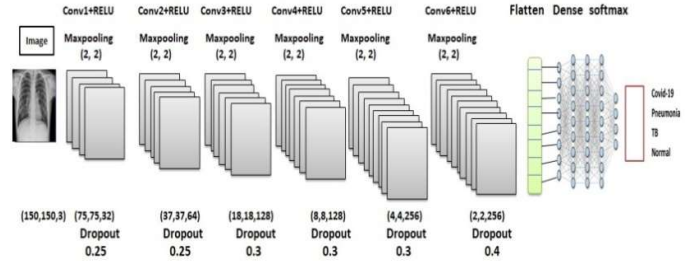
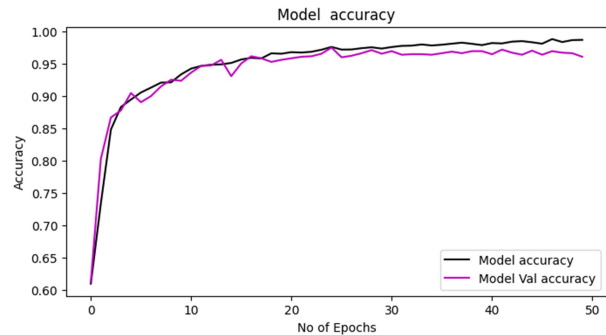


Figure 2: Architecture of the proposed model

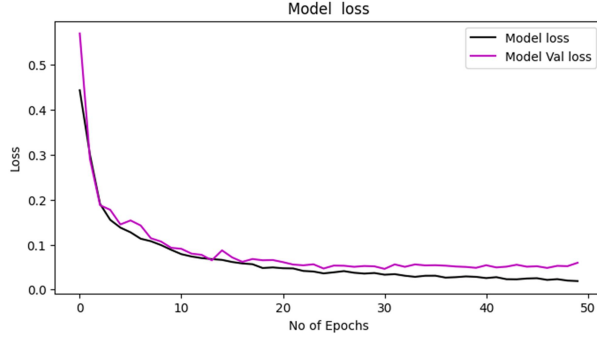
After these convolutional and max-pooling layers, we have used a flatten layer to convert the extracted features into one-dimensional vector. Following the flatten layer, we added dense (fully connected) layers for the classification task. The first dense layer has 1024 features and the final dense layer has only 4 features, indicating the number of classes in our X-ray image classification. Following the last dense layer, there is a softmax layer with the same number of outputs as the classes. The softmax layer computes class probabilities. In this work categorical cross-entropy is used as the loss function, which is common for multi-class classification problems. We specified a learning rate of 0.00150 for the optimizer. The learning rate controls how quickly the model adjusts its weights during training. Hyperparameter tuning was performed to optimize parameters like step size, kernel size, number of dropouts and number of channels. We obtained results as output in the output layer and achieved optimal accuracy in CXR image classification.

V. EXPERIMENTAL RESULTS

In this study, we classified CXR images using a CNN modified model. Adamax optimizer is used to compile the CNN model after it has been constructed. The experimental results of the proposed model are compared with the existing model [13] and also with other state of the art architectures. It is a very potent optimization method for deep learning networks. The suggested model, which utilized the adamax optimizer with a learning rate of 0.00150, successfully classified CXR images and obtained 98.51% accuracy.



(A). Training and validation accuracy vs epochs



(B). Training and validation loss vs epochs

Figure 3: Training and validation loss and accuracy

Our study’s main training goal is to reduce the average probability error between each pixel’s anticipated and the actual values for CXR images. This was accomplished by using categorical cross-entropy as the loss function [15].

This method works well for multi-class classification problems such as our CXR images classification, as it penalizes discrepancies between predicted and true class labels, making it easier to optimize the CNN model’s parameters. With the help of this loss function, which efficiently measures the difference between the ground truth labels and the projected probability distributions, the network can iteratively modify its weights and biases in order to reduce this difference over the course of subsequent training epochs.

This progression across the 50 epochs is graphically depicted in Figure 3, which provides insights into the convergence behavior and generalizability of the model to new data. We can evaluate the learning dynamics of the proposed CNN model, to spot any overfitting or under fitting scenarios and adjust hyperparameters to maximize performance by examining the trends shown in the graphical output of the model.

A. Confusion Matrix

A confusion matrix used as a fundamental tool in classification tasks, offering a concise summary of the performance of a classification algorithm. In essence, it’s a square matrix that can handle several classes; normally, it’s 2×2 in size for binary classification issues. The results of a classification task are concisely summarized in this matrix, where the rows represent the actual class labels and the columns represent the anticipated class labels. Figure 4 shows the Confusion Matrix of the proposed network. The four fundamental values within the confusion matrix are as follows:

True Positive (TP): This describes situations in which both the expected class and the actual class are positive. For example, in the medical domain, this might mean accurately diagnosing patients with a certain illness.

True Negative (TN): In these cases, the predicted and actual classes are both negative. In the context of medical

diagnostics, this could mean accurately identifying people who do not have a specific illness.

False Positive (FP): This is also referred to as a Type I error and happens when the actual class is negative but the projected class is positive.

False Negative (FN): When the expected class is negative but the actual class is positive, this results in a false negative (Type II mistake). This could indicate, in the context of health care applications, that a patient was not correctly diagnosed with a condition.

Researchers can assess the accuracy, precision, recall, and other performance parameters of the classification algorithm by examining these four variables, which offer crucial insights into the algorithm’s operation. The distribution of examples inside the confusion matrix can be analyzed to spot misclassification trends and gauge how well the algorithm performs generally in differentiating between classes. As a result, the confusion matrix is an essential analytical tool for evaluating the advantages and disadvantages of classification algorithms, which helps to improve and optimize them for a range of practical uses.

Precision: In the context of classification, precision is a metric that calculates the proportion of all cases that were correctly anticipated to be positive, that a classifier predicts as positive. It is calculated as using Eq. (1):

$$\text{Precision} = \frac{TP}{TP+FP} \quad (1)$$

Recall (Sensitivity or true positive rate): In the context of classification, Recall is a performance metric that measures how well the classifier identified and retrieved all instances in the dataset that belong to a particular class. It is calculated as using Eq. (2):

$$\text{Recall or true positive rate} = \frac{TP}{TP+FN} \quad (2)$$

F1-Score: The F1-score is the harmonic mean of precision and recall. This provides a fair evaluation of a classifier’s performance by taking into account both precision and recall. It is calculated as using Eq. (3):

$$\text{F1-score} = \frac{2(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP+FP+FN} \quad (3)$$

The overall training performance of the proposed model derived from the confusion matrix is presented in Table. I while the validation performance is shown in Table. II.

Table. III shows the comparative Analysis of the proposed network with the other deep learning networks performing classification of TB, pneumonia, COVID-19 and healthy.

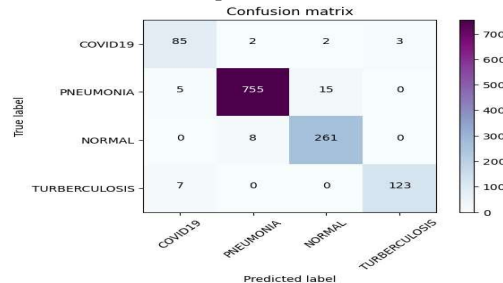


Figure 4: Confusion matrix

Table I. Training results of proposed CNN modification method

Class name	Precision	Recall	F1-score	Support
Covid-19	0.98	0.98	0.98	368
Pneumonia	1.00	0.98	0.99	3100
Normal	0.94	1.00	0.97	1072
Tuberculosis	1.00	0.98	0.99	520

Table II. Validation results of proposed CNN modification method.

Class name	Precision	Recall	F1-score	Support
Covid-19	0.88	0.92	0.90	92
Pneumonia	0.99	0.95	0.97	775
Normal	0.88	1.00	0.94	269
Tuberculosis	0.98	0.92	0.95	130

Table III. Comparative analysis of the proposed network with other deep learning networks performing classification of pneumonia, TB, COVID-19, and healthy.

Model	Precision (%)	Recall (%)	AUC-ROC(%)	F1(%)
Proposed Model	98	98.5	98.51	98.25
DenResCov-19	82.90	69.7	95.00	75.75
DenseNet-121	79.35	62.70	91.00	70.07
ResNet-50	78.60	62.00	93.21	69.51

VI. CONCLUSION

In the field of medical science, classifying CXR images is crucial. This research project aims to create a new type of computer algorithm called a Convolutional Neural Network (CNN) that can distinguish between four different categories of CXR images: Pneumonia, Covid-19, TB, and Normal. The process starts with preparing the images by making them a consistent size and reducing noise. Then, the dataset is divided into two parts: one part for training the algorithm and another part for validating its performance. After that, the CNN is used to automatically categorize the CXR images. Impressively, the model achieved a high accuracy rate of 98.51% in experimental tests using the CXR image dataset. Looking ahead, future research should aim to find new and innovative ways to classify CXR images in the medical field. This work will contribute to efforts to investigate the potential of artificial intelligence (AI) in the future. Additionally, it contains a number of research avenues that could be pursued in the future that are pertinent to the study of medical sciences.

REFERENCES

- [1] J. E. Lujan, Y. Villuendas-Rey, I. Lopez-Yanez, O. Camacho-Nieto and C. Yanez-Marquez, "NanoChest-Net: A Simple Convolutional Network for Radiological Studies Classification", *Diagnostics (Basel)*, 11(5):775, 2021, doi: 10.3390/diagnostics11050775.
- [2] J. W. Song, S. M. Lam, X. Fan, W.J. Cao, S.Y. Wang and H. Tian, "Omics-driven systems interrogation of metabolic dysregulation in COVID-19 pathogenesis", *Cell Metabolism* 32, pp. 188-202.e5, 2020.
- [3] S. Lalmanawma, J. Hussain, L. Chhakchhuak, "Applications of machine learning and artificial intelligence for Covid-19", (SARS-CoV-2) pandemic: a review. *Chaos, Solitons Fractals* 139, 110059, 2020.
- [4] D. Das, K. C. Santosh and U. Pal, "Truncated inception net: COVID-19 outbreak screening using chest x-rays", *Phys. Eng. Sci. Med.* 43, pp. 915-925, 2020.
- [5] L. Sarker, M. M. Islam, T. Hannan and Z. Ahmed, "COVID-DenseNet: a deep learning architecture to detect COVID-19 from chest radiology images", *Preprints*, 2020050151, 2020.
- [6] K. Li, Y. Fang, W. Li, and C. Pan, "CT image visual quantitative evaluation and clinical classification of coronavirus disease (COVID-19)", *Eur. Radiol.* 30, pp. 4407-4416, 2020a.
- [7] A.K. Jaiswal, P. Tiwari, S. Kumar, D. Gupta, A. Khanna and J. J. Rodrigues, "Identifying pneumonia in chest x-rays: a deep learning approach", *Measurement* 145, pp. 511-518, 2019.
- [8] S. Bharati, P. Podder, and M. R. H. Mondal, "Hybrid deep learning for detecting lung diseases from x-ray images", *Inform. Med. Unlocked* 20, 100391, 2020.
- [9] R. M. Pereira, D. Bertolini, L. O. Teixeira, C. N. Silla and Y. M. Costa, "COVID-19 identification in chest x-ray images on flat and hierarchical classification scenarios", *Comput. Methods Progr. Biomed.* 194, 105532, 2020. [12]
- [10] J. Chen, L. Wu, J. Zhang, L. Zhang, D. Gong, Y. Zhao and Q. Chen, "Deep learning-based model for detecting 2019 novel coronavirus pneumonia on high-resolution computed tomography", *Sci. Rep.* 10, 19196, 2020.
- [11] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim and U.R. Acharya, "Automated detection of COVID-19 cases using deep neural networks with x-ray images", *Comput. Biol. Med.* 121, 103792, 2020a.
- [12] T. Rahman, M. E. Chowdhury, A. Khandakar, K. R. Islam, K. F. Islam and S. Kashem, "Transfer Learning with Deep Convolutional Neural Network (CNN) for Pneumonia Detection using Chest X-ray", *Applied Sciences*, 10(9), p.3233, 2020. DOI: 10.3390/app10093233.
- [13] M. Mamalakis, A. Swift, B. Vorselaars, S. Ray, S. Weeks and A. Banerjee, "DenResCov-19: A deep transfer learning network for robust automatic classification of COVID-19, pneumonia, and tuberculosis from X-rays," *Comput Med Imaging Graph.*, 94:102008, 2021. doi: 10.1016/j.compmedimag.2021.102008.
- [14] R. Girshick, J. Donahue, T. Darrell and J. Malik, "DenResCov-19: Rich feature hierarchies for accurate object detection and semantic segmentation," In proceeding of the IEEE conference on computer vision and pattern recognition, pp. 580-587, 2020.
- [15] U. Ruby and V. Yendapalli, "Binary cross-entropy with deep learning technique for image classification," *International Journal of Advanced Trends in Computer Science and Engineering*, 9, 10, 2020.
- [16] A. Swaraj and K. Verma, "Classification of COVID-19 on chest X-Ray images using Deep Learning model with Histogram Equalization and Lungs Segmentation," *arXiv preprint arXiv:2112.0247*, 2021.

On Evaluation of discrete RL agents for Traffic Scheduling and Trajectory Optimization of UAV-based IoT Network with multiple RIS unit

Sheheryar*

Department of Electrical Engineering,
National University of Sciences and Technology
(NUST),
Islamabad, Pakistan
syar.msee19seecs@seecs.edu.pk

Sania Gul*

Department of Electrical Engineering,
University of Engineering and
Technology (UET),
Peshawar, Pakistan
saniagul@hotmail.com

Rizwan Ahmad

Department of Electrical Engineering,
National University of Sciences and
Technology (NUST),
Islamabad, Pakistan
rizwan.ahmad@seecs.edu.pk

*Equal contribution

Abstract: Unmanned Aerial Vehicles (UAVs) have been very effective for data collection from widely spread Internet of Things Devices (IoTDs). However, in case of obstacles, the Line of Sight (LoS) link between the UAV and IoTDs will be blocked. To address this issue, the Reconfigurable Intelligent Surface (RIS) has been used, especially in urban areas, to extend the communication beyond the obstacles, thus enabling efficient data transfer in situations where the LoS link does not exist. In this work, the goal is to jointly optimize the trajectory and minimize the energy consumption of UAV on one hand and satisfying the data throughput requirement of each IoTD on the other hand. As it is a mixed integer non-convex problem, Reinforcement Learning (RL); a class of Machine Learning (ML), is used to solve it, which has proven to be computationally faster than the conventional techniques to solve such problems. In this paper, three discrete RL agents i.e. Double Deep Q Network (DDQN), Proximal Policy Optimization (PPO), and PPO with Recurrent Neural Network (PPOwRNN) are tested with multiple RISs to enhance the data transfer and trajectory optimization in an Internet of Things (IoT) network. The results show that DDQN with multiple RIS is more efficient in saving the communication related energy, while a single RIS system with the PPO agent provides more reduction in the UAV's propulsion energy consumption, when compared to other agents.

Keywords: *Internet of Things Devices (IoTDs); Reconfigurable Intelligent Surfaces (RIS); Reinforcement Learning (RL) agents; Unmanned Aerial Vehicle (UAV).*

I. INTRODUCTION

The concept of 'smart cities', 'smart agriculture', 'smart industries', 'smart transportation systems', and 'smart health care systems' relies heavily on information and communications technologies to gather the information critical for the efficient use of existing assets and resources, for increasing the land and the industrial productivity and for improving the safety and risk monitoring. The 'smart' concept requires smooth data collection from various sensors connected to the 'Internet of Things (IoT)' network, hence requiring integration of information and communication technologies. It is forecasted that by 2025, the number of IoT devices (IoTDs) may exceed 500 billion [1] thus requiring an enabling technology to handle this load. 6G is a potential technology expected to provide an enhanced user experience and better Quality of Service (QoS) for IoT networks due to its superior features over the previous network generations such as 1) ultra-low-latency,

2) extremely high throughput, 3) satellite-based customer services, and 4) massive autonomous networks [2]. Artificial Intelligence (AI), real-time intelligent edge processing, cognitive radio, the use of Unmanned Aerial Vehicles (UAVs), and Reconfigurable Intelligent Surfaces (RISs) enable the 6G networks to provide communication in difficult to reach areas (rural/ mountainous), as well as providing better QoS in densely populated urban areas [3]. RIS can smartly reconfigure the radio environment by incurring some change (amplitude/ phase or both) in the incident signal [4]. Compared to the active relays, RIS does not require power to amplify and transmit the signal. It just reflects the signal, making it a cost-effective solution [4]. Instead of installing multiple antenna towers to collect the data from these power constrained IoTDs, spread over a large geographical area; UAVs have proved to be more economical. However, the UAV itself is an energy-constrained device and requires batteries or solar panels to enhance its flight time. This increases its weight, which in turn would result in more energy consumption.

Keeping in view the great potential of the 6G-IoT networks, many efforts have been put into research in this area. In a RIS-assisted UAV system for the IoT network proposed in [5], the UAV trajectory and the communication channel allocation to multiple IoTDs are jointly optimized by using the PPO agent. The RIS configuration is handled by invoking the Block Coordinate Decent (BCD) algorithm, where a finite set of phase angles are tested for each RIS element to maximize the amount of data collected from each IoTD. The objective was to provide a timely data collection service before the information becomes stale and is of no use. The proposed model has outperformed a similar model without a RIS, a model with randomly configured RIS, and the other two models implementing the random walk UAV and the stationary UAV configurations. In an IoT network of [6], the trajectory of UAV, RIS configuration, and traffic scheduling of IoTDs are jointly optimized by using the Double Deep Q Network (DDQN) and Deep Deterministic Policy Gradient (DDPG) Reinforcement Learning (RL) agents. It was found that the DDPG agent performs better than the DDQN agent, due to its continuous nature. The proposed systems have performed better than the system without RIS and the system with RIS, but without optimal phase shifts configured. In the RIS-assisted UAV system of

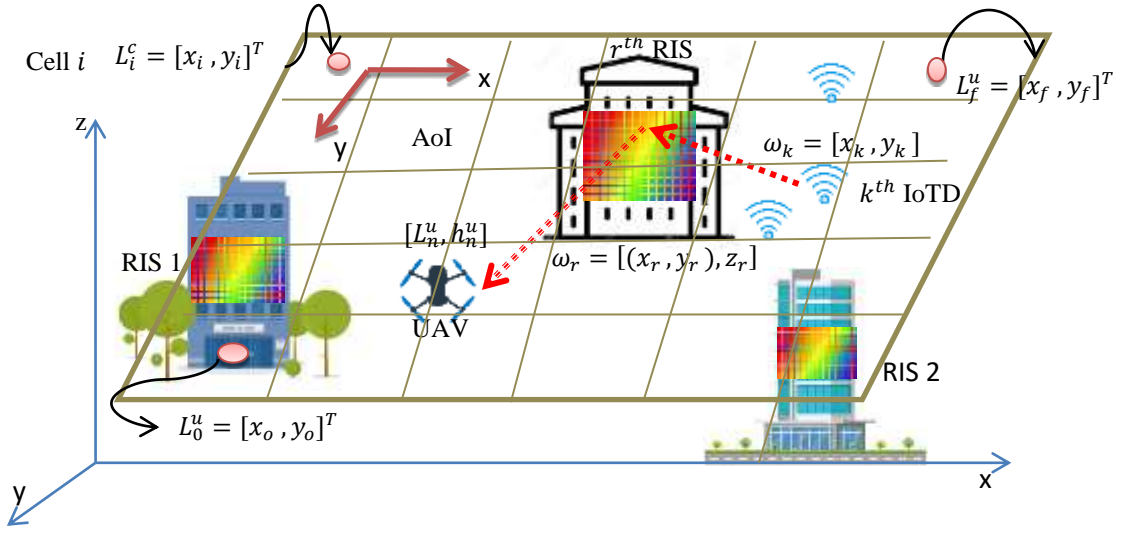


Figure.1: Network architecture with Area of Interest (AoI)

[7], multiple RISs are proposed to serve a single moving user. It was found that using multiple RIS units would result in increasing the communication Energy Efficiency (EE) and reducing the propulsion energy consumption. At any time, only one RIS is assumed to be active, while the others are considered to be in sleep mode to avoid the occurrence of destructive interference at the Ground Terminal (GT). The scheduling of RIS is based on its proximity to GT. In this model, Deep Q Network (DQN) and DDPG agents are used for designing the UAV trajectory and configuring the RIS phase shifts. Again, the DDPG agent outperforms the DQN agent, and also produces better results than the system with fixed UAV and fixed RIS phases and another with randomly moving UAV and random RIS phases.

In this paper, our main objective is to reduce the energy consumption of UAV, by using various discrete RL agents and multiple RIS. Instead of using multiple UAVs (as proposed by [8]), we prefer designing our proposed system with multiple RIS units, because the RIS is a passive device and its energy requirements are far lesser than the UAV. In our proposed IoT model, the UAV motion is allowed in 3D trajectory as in [6] and the system is tested with multiple agents, including PPO (as in [5]) and DDQN (as in [6]) and a newer agent PPO with Recurrent Neural Network (PPOwRNN). A comparison of these agents is made for IoT networks with single and multiple RIS. Our model uses Time Division Multiple Access (TDMA) for traffic scheduling.

The rest of the paper is organized as follows. In section II, we will discuss our proposed methodology. Section III covers the experimental details. Results are given in section IV and finally, the paper is concluded in section V.

II. PROPOSED METHODOLOGY

Figure 1 shows the environment of our proposed model. As shown there, a single UAV is deployed to serve the K IoTDs, spread randomly on ground, in the Area of Interest (AoI). The AoI is partitioned into equal sized L cells. The location of the i^{th} cell in xy coordinates is given by $L_i^c = [x_i, y_i]^T \in \mathbb{R}^{2 \times 1}$, where L_i^c is the center of cell i . The centers of the adjacent cells are separated by a distance of x_s and y_s in x and y coordinate respectively. The horizontal location of the

of k^{th} IoTD is given as $\omega_k = [x_k, y_k]$, while as they are on ground level, so their height $z_k = 0$, and the average amount of data of the k^{th} IoTD is D_k , which needs to be uploaded to the UAV.

The horizontal location of UAV at time n is given as $L_n^u \in \mathcal{L}$ where $\mathcal{L} = \{1, 2, 3, \dots, L\}$, $n = \{1, 2, 3, \dots, N\}$, L is the total number of possible locations, and N is the maximum number of time instants, defined for an agent's training episode. At the end of each episode, the system restarts with the reset conditions, defined for any agent by the designer. Keeping the same nomenclature, L_0^u and L_f^u would be the initial and final locations of the UAV respectively. Then the horizontal trajectory of UAV in N instants is given as $\{L_0^u, L_1^u, L_2^u, L_3^u, \dots, L_n^u, \dots, L_N^u, L_f^u\}$. In the vertical direction, the UAV height level at any time instant n is given as h_n^u , where $h_n^u \in \mathcal{H} = \{1, 2, 3, \dots, H\}$ and H is the total number of height levels. The height is also divided into discrete steps of size h_s , where $h_s = h_{max}/H$. Then the height of UAV at any instant n is given as $H_n^u = h_n^u h_s$, constrained by the condition in eq. (1)

$$h_{min} \leq H_n^u \leq h_{max} \quad (1)$$

with h_{min} and h_{max} being the maximum and the minimum heights respectively. We assume that t_n^u is the duration of time slot n , allocated to an IoTD and is given by eq. (2).

$$t_{min} \leq t_n^u \leq t_{max} \quad (2)$$

The total flight time of UAV is given as in eq. (3).

$$\tau = \sum_{n=1}^N t_n^u \quad (3)$$

The 3D trajectory of UAV is given by N waypoints, where a single waypoint is given as $[L_n^u, H_n^u], \forall n \in N$ along with the time slot duration $t_n^u, \forall n \in N$. With the constraint of maximum horizontal speed V_{max}^h on UAV, the number of time instants N must be large enough that the UAV location within duration t_n^u must remain negligible as compared to the link distances between the UAV and the IoTDs. The horizontal speed of UAV in any time n is given by eq. (4).

$$v_n^h = \frac{|L_{n+1}^u - L_n^u|}{t_n^u} \leq V_{max}^h, \quad \forall n \in N \quad (4)$$

If $v_n^h = 0$, the UAV will hover at time n . Likewise, for the maximum vertical speed V_{max}^v , UAV's vertical speed at any time n is given by eq. (5).

$$v_n^v = \frac{|H_{n+1}^u - H_n^u|}{t_n^u} \leq V_{max}^v, \quad \forall n \in N \quad (5)$$

For rotary wing UAV, the propulsion energy at time n is given in eq. (6) as:

$$e_n^{uav} = t_n^u (P_0 \left(1 + \frac{3(v_n^h)^2}{u_{tip}^2}\right) + \frac{1}{2} d_0 \rho s \delta (v_n^h)^3 + P_1 \left(\sqrt{1 + \frac{(v_n^h)^4}{4v_0^4}} - \frac{(v_n^h)^2}{2v_0^2}\right)^{1/2} + P_2 v_n^v) \quad (6)$$

where P_0 , P_1 and P_2 are constant blade profile power, induced power in hovering status and constant ascending/descending powers respectively. u_{tip} is the tip speed of rotor blade, d_0 is the main body drag ratio, s is the rotor's solidity, v_0 is the mean rotor's induced velocity while hovering, ρ the air density and δ denotes the rotor's disc area.

As shown in Figure 1, there are multiple RISs in the AoI. Each RIS has a large number of Passive Reflecting Units (PRUs). PRUs are arranged in $M_r \times M_c$ uniform planar array and the distance between the adjacent PRUs is d_r meters row-wise and d_c meters column-wise. The reflection coefficient $R_{mr,mc}$ of each PRU is given by eq. (7) as:

$$R_{mr,mc} = \alpha e^{j\theta_{mr,mc}}, \quad \forall m_r \in \{1, \dots, M_r\}, m_c \in \{1, \dots, M_c\} \quad (7)$$

where α is the attenuation loss of PRU and $\theta_{mr,mc}$ is the phase-shift of PRU, adjusted according to the direction of the IoTD being served.

Suppose that there are total R RIS available in the AoI. Then, at any particular time n , only one of them is considered to be switched-on, while the rest are switched-off or in sleep mode. The turn-on schedule decision of the r^{th} RIS is given as $r_n^{RIS} = \{0, 1\}$, where $r_n^{RIS} = 1$ means r^{th} RIS is switched on and $r_n^{RIS} = 0$ indicates that it is switched off at time n . The RIS scheduling scheme is given in eq. (8).

$$r_n^{RIS} = \begin{cases} 1, & r^{RIS} = \text{argmin}(d_n^{uR}) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $d_n^{uR} = \{d_n^{ur}, \forall r \in R\}$ denotes the set of distances between UAV and RIS at time n . Additionally, we have the condition of $\sum_{n=1}^N r_n^{RIS} = 1$, $\forall n \in N$, restricting only a single RIS to be turned-on, which is nearest to the UAV. For the r^{th} RIS, the location of its first element is given by $\omega_r = [x_r, y_r]$ and its height above ground is given by z_r . Although, the horizontal locations of all RISs differ, their height above ground z_r is same.

Assume that the LoS does not exist between the UAV and any of the IoTDs, making it necessary for the IoTD to transmit its data to UAV via RIS. Now the data from the IoTD is uploaded to the UAV via a cascaded channel, consisting of two LoS links (as shown by dotted lines in

Figure 1). The first LoS link is from IoTD to RIS and the other is from RIS to UAV. Then the channel gain between the UAV to r^{th} RIS (switched-on according to eq. (8)) at time n) can be denoted as $g_n^{ur} \in \mathbb{C}^{M_r \times M_c}$ and given in eq. (9) as

$$g_n^{ur} = \frac{\sqrt{\xi}}{d_n^{ur}} \left[1, e^{-j\frac{2\pi}{\lambda} d_r \phi_n^{ur} \psi_n^{ur}}, \dots, e^{-j\frac{2\pi}{\lambda} (M_r - 1) d_r \phi_n^{ur} \psi_n^{ur}} \right]^T \otimes \left[1, e^{-j\frac{2\pi}{\lambda} d_c \phi_n^{ur} \psi_n^{ur}}, \dots, e^{-j\frac{2\pi}{\lambda} (M_c - 1) d_c \phi_n^{ur} \psi_n^{ur}} \right]^T \quad (9)$$

where $[\cdot]^T$ represents the transpose operation, ξ is the path loss at a reference distance $D_o = 1$ m, $d_n^{ur} = \sqrt{(H_n^u - z_r)^2 + (L_n^u - \omega_r)^2}$, λ is the carrier wavelength, $\phi_n^{ur} = \frac{x_i - x_r}{d_n^{ur}}$ is the cosine and $\psi_n^{ur} = \frac{y_i - y_r}{d_n^{ur}}$ is the sine of the horizontal Angle of Arrival (AoA) and $\psi_n^{ur} = \frac{h_n^u - z_r}{d_n^{ur}}$ is the sine of the vertical AoA of signal at the r^{th} RIS. The symbol \otimes shows the tensor product. Far-field array response vector model is assumed at RIS, where $d_n^{ur} \gg \max(M_r d_r, M_c d_c)$.

The channel gain between r^{th} RIS and k^{th} IoTD is denoted as $g_n^{rk} \in \mathbb{C}^{M_r \times M_c}$ and given as in eq. (10).

$$g_n^{rk} = \frac{\sqrt{\xi}}{d_n^{rk}} \left[1, e^{-j\frac{2\pi}{\lambda} d_r \phi_n^{rk} \psi_n^{rk}}, \dots, e^{-j\frac{2\pi}{\lambda} (M_r - 1) d_r \phi_n^{rk} \psi_n^{rk}} \right]^T \otimes \left[1, e^{-j\frac{2\pi}{\lambda} d_c \phi_n^{rk} \psi_n^{rk}}, \dots, e^{-j\frac{2\pi}{\lambda} (M_c - 1) d_c \phi_n^{rk} \psi_n^{rk}} \right]^T \quad (10)$$

where $d_n^{rk} = \sqrt{(z_r)^2 + (\omega_r - \omega_k)^2}$, $\phi_n^{rk} = \frac{x_r - x_k}{d_n^{rk}}$ is the cosine and $\psi_n^{rk} = \frac{y_r - y_k}{d_n^{rk}}$ is the sine of the horizontal AoA and $\psi_n^{rk} = \frac{z_r}{d_n^{rk}}$ is the sine of the vertical AoA of signal at the k^{th} IoTD.

The cascaded channel gain is then given as in eq. (11).

$$g_n^{urk} = g_n^{ur} \cdot \boldsymbol{\theta}_n \cdot g_n^{rk} \quad (11)$$

where $\boldsymbol{\theta}_n = \text{diag}(\theta_n) \in \mathbb{C}^{M_r M_c \times M_r M_c}$ is the r^{th} RIS reflection coefficient matrix, and $\theta_n = [e_n^{j\theta_{1,1}}, \dots, e_n^{j\theta_{mr,mc}}, \dots, e_n^{j\theta_{M_r, M_c}}]^T \in \mathbb{C}^{M_r M_c \times 1}$

According to our assumption of Non-LoS (NLoS) path between UAV and IoTD, the channel gain at the k^{th} IoTD is given by eq. (12) as

$$g_n^k = (1 - p_n^k) \frac{\xi}{(d_n^{uk})^2} + p_n^k g_n^{urk} \quad (12)$$

Where $d_n^{uk} = \sqrt{(H_n^u)^2 + (L_n^u - \omega_k)^2}$ and p_n^k is the blocking probability between UAV and k^{th} IoTD at time n , given by eq. (13) as

$$p_n^k = 1 - \frac{1}{1 + \eta_1 e^{-\eta_2 \tan^{-1}\left(\frac{H_n^u}{d_n^{uk}}\right) - \eta_1}} \quad (13)$$

Where η_1 and η_2 are constants depending on environment. The data rate R_n^k achieved at the k^{th} IoTD is given in eq. (14).

$$R_n^k = c_n^k B \log_2 \left(1 + \frac{P g_n^k}{B \sigma^2} \right) \quad (14)$$

where P is the transmission power of an IoTD, B is the channel bandwidth, σ is the noise variance and $c_n^k \in \{0, 1\}$ is the scheduling decision for the k^{th} IoTD at time n . As TDMA is used, so only a single IoTD is scheduled for service at any time instant, i.e. $\sum_{n=1}^N c_n^k = 1, \forall n \in N$.

The aim of this research is to minimize the propulsion energy and maximize the EE of UAV. Let $L = \{L_n^u, n \in N\}$, $H = \{h_n^u, n \in N\}$, $\mathcal{C} = \{c_n^k, n \in N\}$, $T = \{t_n^u, n \in N\}$ and $\theta = \{\theta_n, n \in N\}$ as already defined in the discussion above. The optimization problem can be formulated in eq. (15) as.

$$P: \quad \min_{L, H, \mathcal{C}, T, \theta} \sum_{n=1}^N e_n^{uav} \quad (15)$$

Due to being a mixed variable problem, eq. (15) is non-convex and would need extensive computational resources to be solved in real time. As already pointed out, RL has been very effective in solving such optimization problems in real time. The goal of eq. (15) is to minimize the fuel consumption of UAV at all times with the constraints of: 1). $c_n^k \in \{0, 1\}$, $\sum_{n=1}^N c_n^k = 1, \forall n \in N$, ensuring a single IoTD links to UAV at any time instant n , 2). $\sum_{n=1}^N t_n^u R_n^k \geq D_k, \forall k \in K$, for making sure that the data uploading from k^{th} IoTD must be completed within the flight time of UAV, 3). $v_n^h \leq V_{max}^h$, 4). $v_n^v \leq V_{max}^v$ and 5). $h_{min} \leq h_n^u \leq h_{max}, \forall n \in N$ for guaranteeing that the UAV horizontal and vertical speeds and its heights does not exceed the maximum limits and finally 6). $t_{min} \leq t_n^u \leq t_{max}$ restricts the time slot duration allocation to an IoTD between t_{min} and t_{max} for data transmission.

For eq. (15), the observation and action spaces and the rewards for RL agents are defined below.

Observation space: The current state $s(n)$ at time n is defined as in eq. (16)

$$s(n) = \{s_u(n)\} \in \mathcal{S} \triangleq \mathcal{L} \times \mathcal{H} \quad (16)$$

Where \mathcal{S} is the overall state space and $s(n) = (L_n^u, h_n^u) \in \mathcal{L} \times \mathcal{H}$ is the current horizontal and vertical location of UAV.

Action space: The action space can be continuous or discrete, depending on the type of agent used. However, as we are using only the discrete agents, so, we will define only the discrete action space here. Ref [6] provides continuous space for a similar problem. The discrete action $a(n)$ at time n is defined as in eq. (17).

$$a(n) = \{l_n, h_n, c_n^k, t_n^u\} \in A \triangleq \mathcal{L}_u \times \mathcal{H}_u \times \mathcal{C} \times T \quad (17)$$

Where A is the overall action space and $a(n) = (l_n, h_n, c_n^k, t_n^u) \in \mathcal{L}_u \times \mathcal{H}_u \times \mathcal{C} \times T$ is the current action chosen from A . $\mathcal{L}_u \times \mathcal{H}_u$ is the action space from which the current UAV flying actions in horizontal and vertical

directions l_n, h_n are chosen. \mathcal{C} is the IoTD scheduling space $\mathcal{C} = \{c_n^k, \forall k, n\}$ and $T = \{t_{min} : 0.1ms : t_{max}\}$ is the action space of discrete flight time durations from which t_n^u is chosen between t_{min} and t_{max} with the step size of 0.1ms. Assume that during one time instant n , the UAV is only allowed to move to one of its adjacent cells from its current cell in the horizontal plane and also it is allowed only to move one step in the vertical direction. Then the horizontal and vertical locations of UAV in next time instant $n + 1$ is given as in eq. (17) and (18):

$$L_{n+1}^u = L_n^u + l_n \quad (17)$$

$$H_{n+1}^u = H_n^u + h_n \quad (18)$$

where l_n is the horizontal flying action of UAV defined as $l_n \in \mathcal{L}_u \triangleq \{(0, y_s), ((0, -y_s), (x_s, 0), (-x_s, 0), (0, 0)\}$ with \mathcal{L}_u being the horizontal action space of UAV comprising 5 actions including moving to north, south, east, west or hover at its current location respectively, and h_n is the vertical flying action of UAV defined as $h_n \in \mathcal{H}_u \triangleq \{h_s, -h_s, 0\}$ with \mathcal{H}_u being the vertical action space of UAV comprising 3 actions including moving upward, downward or stay there respectively.

Step reward: The reward of taking an action $a(n)$, when the system is in state $s(n)$ at time n , is given by eq. (19):

$$rwd(s(n), a(n)) = \sum_{k=1}^K \sum_{n=1}^N \frac{t_n^u R_n^k}{e_n^{uav}} - p_0 \quad (19)$$

where the reward function rwd is the ratio of the total data uploaded by the K IoTDs upto time instant $n + 1$, to the propulsion energy of UAV consumed till that time. However, penalty p_0 is applied to reward, if the data transferred to UAV at time n is less than the average data rate of any IoTD i.e. $t_n^u R_n^k < \frac{D_k}{N}$.

The reward would be highest if the data rate R_n^k is maximized. R_n^k can be maximized by maximizing the device gain g_n^k , which depend on cascaded gain g_n^{urk} , which in turn is maximized, when the phase shift of each PRU of the active RIS (having $r_n^{RIS} = 1$) is adjusted to produce the maximum beamforming towards the IoTD scheduled during the current time n . The maximum reward for the state-actor pair $s(n), a(n)$ is given as in eq. (20).

$$rwd^{max}(s(n), a(n)) = \sum_{k=1}^K \sum_{n=1}^N \frac{t_n^u R_n^{kmax}}{e_n^{uav}} - p_0 \quad (20)$$

The DDQN, PPO and PPOwRNN agents are used for solving the complex optimization problem given in eq. (15). The first agent in the list belongs to the critic and the last 2 belongs to the actor-critic categories of RL agents respectively. We will use the implementations of [9] for DDQN, and [10] for the PPO and PPOwRNN agents.

III. EXPERIMENTAL EVALUATION PARAMETERS

The UAV's trajectory in the presence of multiple RIS is depicted in Figure 2 and the settings for the environment, observation and action spaces, and the hyper-parameters for the Neural Networks (NNs) of RL agents along their training settings are listed in tables 1 to 3 respectively.

Table 1: Environment settings

Environment	Symbol	Value
IoTDs and AoI	AoI	1000m x 1000m
	L	10000
	K	6
	$[\omega_k, z_k]$	[Random, 0]
	h_s	2 m
	x_s	10 m
	y_s	10 m
UAV	u_{tip}	120
	d_0	0.6
	s	0.05
	v_0	4.3
	ρ	1.225
	ϕ	0.503
	P_0	$\frac{12 \times 30^3 \times 0.4^3}{8} \rho s \phi$
	P_1	$\frac{1.1 \times 20^{3/2}}{\sqrt{2\rho\phi}}$
	P_2	11.46
	h_{min}	30m
	h_{max}	100m
	V_{max}^h	10 m/s
	V_{max}^v	10 m/s
	t_{min}	1s
t_{max}	3s	
RIS	R	3
	M	100
	z_r	50m
	ω_r	See the result section
	d_r, d_c	$\lambda/2$
Channel	η_1, η_2	9.61, 0.16
	B	2 MHz
	D_k	1024 bits
	f (carrier frequency)	900MHz
	P	500mW
	σ	169 dBm/Hz
	h_0^u	100m
Reset conditions	$L_0^u = [x_0^u, y_0^u]$	[0,0]
	θ_i	0°
	h_0^u	100m

Table 2: Observation and action space

S	$3 \rightarrow \{[L_n^u, h_n^u] = [x_n^u, y_n^u, h_n^u]\}$
L_n	$5 \rightarrow \{\text{'north', 'east', 'west', 'south', 'hover'}\}$
\mathcal{H}_n	$3 \rightarrow \{\text{'upward', 'downward', 'stay there'}\}$
\mathcal{C}	$6 \rightarrow \{(0,0,0,0,1), (0,0,0,1,0), (0,0,0,1,0,0), (0,0,1,0,0,0), (0,1,0,0,0,0), (1,0,0,0,0,0)\}$
$T = \frac{t_{max} - t_{min}}{0.1ms}$	$21 \rightarrow \{1:0.1:3\}$
A	$5 \times 3 \times 6 \times 21 = 1890$

Table 3: RL agents' NN and training parameters

Evaluation and target networks	DDQN	PPO	PPOwRNN
Total hidden layers in each NN	2	2	2
Total neurons in hidden layer 1	20	200	200
Total neurons in hidden layer 2	20	100	100
Number of LSTM cells inside each hidden neuron	NA	NA	2
Optimizer	RMS prop	Adam	Adam
Weights Initialization	$\mathcal{N}(0,0.003)$	$\mathcal{N}(0,0.003)$	$\mathcal{N}(0,0.003)$
Number of epochs O	NA*	25	25
Number of episodes E	60	120	120
Number of time instants N	600	600	600
Learning rate	0.005	0.001	0.001
Experience buffer F	3200	10000	10000

Mini batch size G	32	8	8
Discount factor γ	0.9	0.9	0.9
Exploration factor ϵ	0.9	NA*	NA*
Clip ratio ϵ_c	NA*	0.2	0.2
Lambda (for advantage estimation function)	NA*	0.95	0.95
Value function loss coefficient	NA*	0.5	0.5

NA* =Not Applicable

PPOwRNN has feedback connections in their NNs, which are composed of Long Short Term Memory (LSTM) cells, which impact the current outputs of the hidden layer's neurons from their previous outputs and the previous outputs of the next layers' neurons.

All agents start with a random policy π , and after every action $a(n)$, the agent stores the current state $s(n)$, action pair $[s(n), a(n)]$, action's reward $rwd(\cdot)$, and the state of the system $s(n+1)$, after the action in its experience buffer F . During the training stage, mini-batches of G random samples from the experience buffer are selected to train the actor and critic's NNs. The discount factor γ determines how the rewards at the individual time steps are weighted. An action's influence over the future states of the environment typically decreases over time. Few parameters listed in Table 3 are specific to an agent and are not applicable for others. For example, the exploration factor ϵ is specific to DDQN and decides how much the agent would explore. Similarly, the number of epochs O , the clip ratio ϵ_c , lambda and the value function loss coefficients are specific to the PPO agents (both PPO and PPOwRNN) and are used for calculating its loss functions.

IV. EXPERIMENTATION RESULTS AND COMPARISON

The results for the DDQN, PPO and PPOwRNN with single and multiple RISs are given in Table 4. The best results and the agents generating them are boldfaced. For multiple RIS, many random locations are tested, but among them, only those which generated the best results for any of the 3 agents, are reported in Table (4).

Table 4: Performance comparison of different agents with multiple RIS

# of RIS	Location $[\omega_r, z_r]$ m	Agent	Propulsion energy (kJ)	(EE) bits/J
1	[50 50 50]	DDQN	135.25	71.29
		PPO	83.21	88.67
		PPOwRNN	94.55	84.69
2	[50 50 50], [900 900 50]	DDQN	147.16	75.42
		PPO	128.05	48.58
		PPOwRNN	112.56	49.80
3	[50 50 50], [250 150 50], [500 500 50]	DDQN	109.49	94.06
		PPO	125.11	44.99
		PPOwRNN	161.94	53.05

As clear from Table 4, the PPO agent with a single RIS provides more propulsion energy savings and the DDQN agent with 3 RIS outperforms other agents in EE.

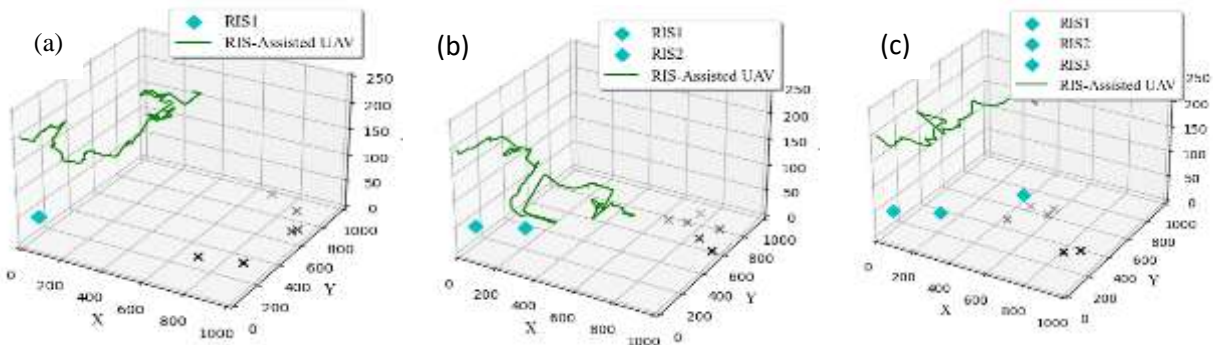


Figure 2: Trajectory of UAV with a) Single RIS, b) Two RIS, c) Three RIS

Among all the RL agents used for our experiments, installing multiple RIS has benefitted the DDQN agent the most, while there is either little or no improvement in the performances of the PPO agent and the PPOwRNN in the presence of multiple RISs. The DDQN agent with 3 RIS surpasses the PPO agent, which is the best available RL agent to date [11].

The computational complexity of the 3 agents is estimated in terms of the average training and evaluation time taken by them on Intel Core i5, 2.81 GHz CPU for multiple RIS unit. On average, the DDQN agent takes 4min and 30s, PPO 10h, 20min and 50s, while the PPOwRNN requires 15h, 40min and 20s for convergence. Although the PPO and PPOwRNN agents save more propulsion energy in single and double RIS cases, they are far more complex than the DDQN agent.

V. CONCLUSION

To collect data from a widely spread IoT network, one solution is to use multiple UAVs, to provide coverage to a large area, as in [8], but owing to the cost of fuel, it would be an expensive option. RIS on the other hand is a low-powered, easy to install, and light weighted device, which can be easily installed on buildings without posing any danger to the public as is caused by the huge antenna structures in case of bad weather conditions or earthquakes. So, we have tested different RL agents for the IoT data collection in the presence of multiple RISs and found that the presence of additional RIS units supports the otherwise weak agents (e.g. DDQN) more than the strong ones (e.g. PPO), which although does not require more RIS units, but are computationally more expensive. It is shown in [6] that the continuous RL agents perform better than the discrete ones. So, in the future, these agents must also be tested with multiple RIS units. Also, more bandwidth efficient access methods e.g. Orthogonal Frequency Division Multiple Access (OFDMA); must be implemented to estimate the maximum potential of the proposed algorithm.

REFERENCES

[1]. M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen and A. Ghrayeb, "UAV Trajectory Planning for Data Collection from Time-Constrained IoT Devices," in *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34-46, January 2020, doi: 10.1109/TWC.2019.2940447.

[2]. D. C. Nguyen et al., "6G Internet of Things: A Comprehensive Survey," in *IEEE Internet of Things*

Journal, vol. 9, no. 1, pp. 359-383, January 2022, doi: 10.1109/JIOT.2021.3103320.

[3]. H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland and F. Tufvesson, "6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities," in *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1166-1199, July 2021, doi: 10.1109/JPROC.2021.3061701.

[4]. Y. Zhao, J. Zhao, W. Zhai, S. Sun, D. Niyato and K.-Y. Lam, "A survey of 6G wireless communications: Emerging technologies" in *arXiv:2004.08549*, July 2020, [online] Available: <http://arxiv.org/abs/2004.08549>.

[5]. A. Al-Hilo, M. Samir, M. Elhattab, C. Assi and S. Sharafeddine, "RIS-Assisted UAV for Timely Data Collection in IoT Networks," in *IEEE Systems Journal*, vol. 17, no. 1, pp. 431-442, March 2023, doi: 10.1109/JSYST.2022.3215279.

[6]. H. Mei, K. Yang, Q. Liu and K. Wang, "3D-Trajectory and Phase-Shift Design for RIS-Assisted UAV Systems Using Deep Reinforcement Learning," in *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3020-3029, March 2022, doi: 10.1109/TVT.2022.3143839.

[7]. L. Wang, K. Wang, C. Pan and N. Aslam, "Joint Trajectory and Passive Beamforming Design for Intelligent Reflecting Surface-Aided UAV Communications: A Deep Reinforcement Learning Approach," in *IEEE Transactions on Mobile Computing (Early Access)*, pp. 1-11, August 2022, doi: 10.1109/TMC.2022.3200998.

[8]. M. Samir, D. Ebrahimi, C. Assi, S. Sharafeddine and A. Ghrayeb, "Leveraging UAVs for Coverage in Cell-Free Vehicular Networks: A Deep Reinforcement Learning Approach," in *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2835-2847, September 2021, doi: 10.1109/TMC.2020.2991326.

[9]. Simulation code for DDQN. <https://github.com/HaiboMei/UAVRIS-DRL.git>

[10] Achiam J Openai spinning up available at: <https://spinningup.openai.com/> (accessed Jun. 7, 2023).

[11]. Schulman, John, et al., "Proximal policy optimization algorithms," in *arXiv preprint*, August 2017, [online] Available: <https://arxiv.org/abs/1707.06347>.

Machine Learning Based Estimation of End Effector position in Three Dimension Robotic Workspace

Hamna Baig¹, Ihtesham Jadoon³
 Department of Electrical and Computer Engineering
 COMSATS University Islamabad Attock Campus
 Attock, Pakistan
 hamnabaig.66@gmail.com¹, jadoon@cuiatk.edu.pk³

Ejaz Ahmed²
 Department of Electrical Engineering Aero Space and Aviation, Air
 University Islamabad Kamra Campus
 Attock, Pakistan
 225258@aack.au.edu.pk²

Abstract — *this research work uses different machine learning algorithms to estimate end effector position in three dimensional robotic work space. Workspace is area around robot where robot can freely move with possible input variations of different joint angles, it is important to exactly know operating area of robot. Conventionally iterative simulation methods were used to find robotic workspace. Which are computationally slow and difficult to model. Our approach utilizes machine-learning algorithms to predict the workspace end effector position. By simulating the robot with Denavit-Hartenberg (D-H) approach in MATLAB, which results in collection of data sets, which are then used in training. Multiple Linear Regression (MLR), Decision-Tree Regression and Artificial Neural Network (ANN) algorithms are trained for prediction. The dataset some portion is used as test data, which is then used for validation step. These methods can be incorporated in real robots. The results from findings show the accuracy of Machine learning algorithms specifically Artificial Neural Network (ANN) performs well then conventional mathematical methods.*

Keywords: Robotics; Simulation; DH-Method; Machine Learning; Kinematics; Prediction.

I. INTRODUCTION

In many crucial cases, it is very difficult, if not impossible, to analytically predict the behavior of physical systems. The necessity to build a physical system prototype drives modelling, which in turn reveals strong motivations to investigate and analyses a system's operation.

Modeling of robot usually carried through kinematics study. That deals with model of robot without any influence of force. The kinematics of a robot deals with the geometric and time-based properties under motion and in particular how various links of a robot move with respect to one another with variation of time. Which is the analytical way of explaining relation between different joint variables. Kinematics modeling can be divided in to two type's forward and inverse kinematics, first one that gives the position and orientation when joint angles and joint positions are known. While the inverse kinematics deals with set of complex equations computed by vector and analytical algebra to compute the joint angles once, end effector position is given [1].

The robot are heart of automation industry used in manufacturing and assembly applications and many applications which required robot to move objects from one place to other using mechanically designed grippers these

robots need to be precise in term of placement of objects. To achieve the precise pick and place robot application it need to be modeled with least error.

Intense research is carried in analytical modeling of robot systems, which are based on line and point transformation. Campos-Macías [2] proposed a geometric model to calculate and find relation between input angles and output position unknown joint angles required for autonomous positioning of a robotic system. In [3] author Bayro-Corrochano uses new algebraic method called quaternion for modeling of different physical systems system. Popovic et al. [4] computed method to model upper extremity movement of the arm of multi leg moving robot inspired by animal's movement. An analytical model based approach to compute kinematics of a humanoid robot was discussed in [5]. In [6] author presented an inverse kinematics model to calculate all the joint variables of a serial arm manipulator. Applications of machine learning in different robots are discussed in [7].

The figure 1 shows the kinematic model's simplified view of robotic arm in an inverted 'L' pose. The first joint S1 is used to move arm claw to pick objects, the joint S2 and S3 is the elbow and shoulder joint respectively to move the arm to the desired position. The S3 joint is base joint to rotate the robot arm. The forward kinematic model of the 4 DOF Robot is presented in Section II and Section III presented the discussion on forward kinematics using machine learning algorithms on MATLAB and Section IV gives the result and Discussion of machine learning algorithms and Section V gives the final Conclusions.

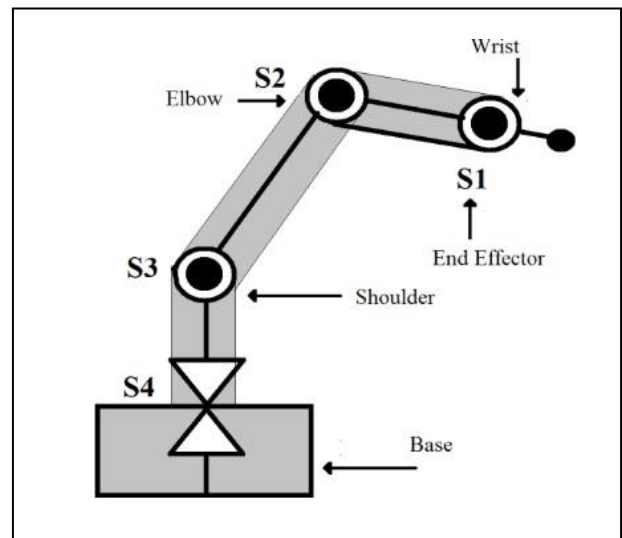


Fig. 1. Kinematic Model Representation of 4-Degree-of-Freedom Robot

A. Aims and Objectives

This paper aims to focus on using machine learning algorithms for estimating the end effector position in three-dimensional robotic workspace. Specifically the paper objectives are as follows;

- Dataset: Generating dataset in MATLAB that contains values of joint angles (θ) and the corresponding end effector positions.
- Model Training: Using the dataset to train machine learning Models MLR (Multiple Linear Regression), DTR (Decision-Tree Regression) and ANN (Artificial Neural Network) for estimating end effector position.
- Prediction Analysis: Using the values from dataset to validate the trained models and access their RMSE (root Mean Squared Error) for prediction.
- Model Comparison: Evaluate the trained MLR, DTR and ANN models on basis of computational time and RMSE.

II. FORWARD KINEMATICS OF 4-DEGREE OF FREEDOM ROBOT

The position and orientation of the end effector calculation of a robotic arm or mechanism based on the joint angles or displacement. The forward kinematics of a 4 Degree-of-Freedom robot usually has four rotational joints that can be calculated by imposing a sequence of transformation from the base frame to the end effector frame. A distinct coordinate system is introduced by every joint, the overall transformation of end effector from the base can be calculated. This process can be implemented using a programming language such as Python and C++ in MATLAB in just a short code. The code will comprise of parameters for each joint for Denavit-Hartenberg approach, resulting in transformation matrices and performing multiplication process on matrices to calculate the overall transformation. On a meanwhile the implementation of the obtained transformation to the end effector original location, the orientation and the final position of the end effector in space can be calculated. This seemed to serve as the essential and indispensable tool for the robots controlling and the movement planning respectively.

B. Denavit-Hartenberg Parameters

Denavit-Hartenberg parameters are employed commonly for the characterization of robots structure. These parameters typically serves as the foundation for conduction of robot kinematics and analysis. There is four DH parameters listed below in Table I to describe the orientation and position of a link. Each parameter for attaching reference frames to robots assembly link is linked to a specific convention. This standardization of coordinate frames across spatial linkages ensures consistency and facilitates analysis.

a_{i-1} = Translation x_i axis

α_{i-1} = Rotation around x_i axis

d_i = Translation around z_{i-1} axis

θ_i = Rotation around z_{i-1} axis

The Denavit-Hartenberg (D-H) decided to use the homogeneous transformation matrix instead. This matrix represents the end effector orientation and position of the robots with respect to the joint angles. Nonetheless, it does not specify the arm arrangement needed to reach this position. Figure (2) depicts the diagram of link coordinate system which is created using DH parameters.

TABLE I.

Symbol	DH Parameters	Description
d_i	Joint Offset	Intersections Length of joint axis to common normal
θ_i	Joint Angles	Angle in between the normal plane to joint axis and orthogonal projections
L	Link Lengths	Axis to the common normal distance
α_{i-1}	Twist Angle	Orthogonal projections of joint axis on the normal plane to common plane Angle.

The vector of joint variables is given by; $\theta = [\theta_1 \theta_2 \theta_3 \theta_4 \theta_5]^T$

The vector of joint distances is given by; $d = [d_1 d_2 d_3 d_4 d_5]^T$

The vector of joint lengths is given by; $a = [a_1 a_2 a_3 a_4 a_5]^T$

The vector of link twist angle is given by; $\alpha = [\alpha_1 \alpha_2 \alpha_3 \alpha_4 \alpha_5]^T$

Denavit-Hartenberg (D-H) parameters are a collection of four parameters as depicted in figure (2) for a 4 DOF robot plays a crucial role in the robot kinematics. With the use of D-H representation a systematic way to express relationships between consecutive links in a robotic serial manipulator is provided. Hence, this method provides the mathematical foundation which is adaptable to a numerous robotic design. It also essentially defines the position and orientation for each link in relation to the other one link that comes before this link [2].

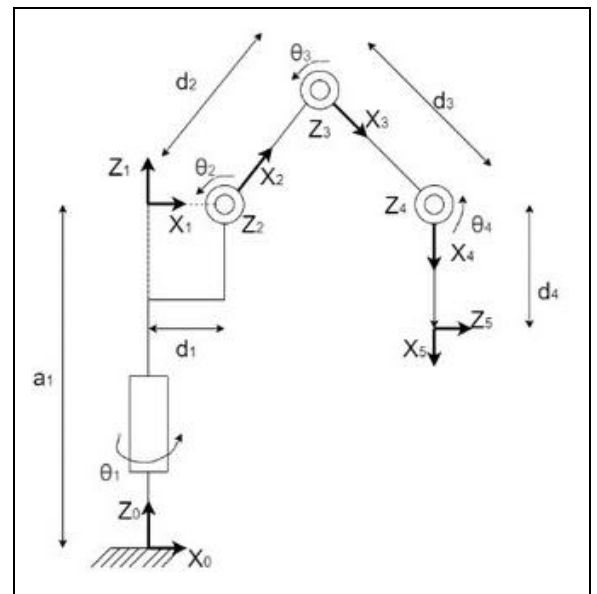


Fig. 2. Denavit-Hartenberg Parameters Labeling of 4-Degree-of-Freedom Robot

C. Model of Forward Kinematics

The 4-Degree-of-Freedom Forward kinematic model uses the Denavit-Hartenberg (DH) parameters in determining the end effector position and orientation based on joint angles for the robot. On an initial level, the DH parameters define each joint geometry including the link length, Joint angles, Joint offset and Twist angles. With the help these settings in place the transformation matrix for each joint explaining the transition between adjacent links as the robot (1) is generated. A single transformation matrix indicating the total effect of each joint movements to the end effector frame from the base frame is generated via the multiplication of matrices. This transformation matrix provides a solution to the problems of forward kinematics by extracting the end effector orientation and location. The matrix for joint number 1 to ith can be calculated as shown in equation (1).

$$A'_{i-1} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A'_{i-1} = \begin{bmatrix} \cos \theta & -\sin \theta \cdot \cos \alpha & \sin \theta \cdot \sin \alpha & a \cdot \cos \alpha \\ \sin \theta & \cos \theta \cdot \cos \alpha & -\cos \theta \cdot \sin \alpha & a \cdot \sin \alpha \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

By multiplication of matrices as depicted in equation (2) gives the transformation matrix.

$${}^0T_4 = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \quad (2)$$

The model of the robot forward kinematic was validated using the MATLAB. A brief understanding of the kinematic behavior of robot was obtained with the use of numerical analysis and also the visualization in the MATLAB environment. The results are contrasted and graphically depicted by giving different sets of angles into both the forward kinematics model established and implemented in MATLAB environment.

TABLE II.

Joint Angles (θ)	Symbol			
	a_{i-1}	d_i	α_{i-1}	θ_i
S1	0	0	0	0
S2	10	10	-90	0
S3	10	10	0	0
S4	10	10	0	0

For example, with a joint angle configuration of [S1 S2 S3 S4] for the values in Table III, the transformation matrix is given below in equation 3 and the visual representation is depicted in figure 2

$$T = \begin{bmatrix} 1.0000 & 0.0000 & -0.0000 & -0.0000 \\ -0.0000 & -0.4481 & -0.8940 & -13.3992 \\ -0.0000 & 0.8940 & -0.4481 & -7.9014 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \quad (3)$$

III. FORWARD KINEMATICS SOLUTION USING MACHINE LEARNING ALGORITHMS

Problems related to machine learning can be classified in to three main types; regression, classification and clustering. In the context of predicting claw positions in mapping with non-linear input poses, the problem at hand falls under regression. An intelligent approach is employed, utilizing machine learning algorithms for obtaining forward kinematics solutions. This Section focuses on predicting the claw position of a 4 Degree-of-Freedom Robot using Multiple Linear Regression (MLR), Artificial Neural Network (ANN) and Decision-Tree Regression Techniques. The algorithms of machine learning are trained and implemented on MATLAB environment. The performance of these three models are being evaluated on the root mean squared error and R squared value basis. The predicted result is then compared with the actual value of the claw position. The result is basically the end effector position on the basis of joint angles.

A. Dataset For Training Machine Learning Model

The dataset is generated through the code in MATLAB through initializing arrays to store workspace points and the link points. The code iterates for all the combinations of joint angles within the specified limits of theta_range (θ) for calculating forward kinematics using DH parameters. For every combination, the end effector position (x, y, z) is calculated and both workspace and the link point are stored. Finally, the dataset is populated with joint angles (θ) and the corresponding end effector positions. Figure 3 shows the robot workspace with links and table III shows the dataset with values of joint angles and the end effector position.

TABLE III. SAMPLE OF THE EXPERIMENTAL DATASET

	S1(θ_1)	S2(θ_2)	S3(θ_3)	S4(θ_4)	X	Y	Z
1	-1.5708	-1.5708	-1.5708	-1.5708	-0.6031	-1.5708	-1.5708
2	-1.5708	-1.5708	-1.5708	-1.2217	-2.3396	-1.5708	-1.5708
3	-1.5708	-1.5708	-1.5708	0.8727	-5.0000	-1.5708	-1.5708
4	-1.5708	-1.5708	-1.5708	-0.5236	-8.2635	-1.5708	-1.5708
5	-1.5708	-1.5708	-1.5708	-0.1745	-11.7363	-1.5708	-1.5708
6	-1.5708	-1.5708	-1.5708	-0.1745	-15.0000	-1.5708	-1.5708
7	-1.5708	-1.5708	-1.5708	0.5236	-17.6604	-1.5708	-1.5708
8	-1.5708	-1.5708	-1.5708	0.8727	-19.9969	-1.5708	-1.5708
9	-1.5708	-1.5708	-1.5708	1.2217	-20.0000	-1.5708	-1.5708
10	-1.5708	-1.5708	-1.5708	1.5708	-4.0233	-1.5708	-1.5708
11	-1.5708	-1.5708	-1.5708	-1.5708	-5.7958	-1.5708	-1.5708
:	:	:	:	:	:	:	:
10000	-2.3701	-3.3498	-1.2217	-3.7892	-3.8495	-4.6590	4.8908

The column 1 to column 4 shows the values of joint angles (S1, S2, S3, S4) and the column 5, 6 and 7 shows the corresponding end effector position variable values (x, y, z). The dataset as shown in Table 3 has 10,000 values. This dataset is used later for training of machine learning algorithms.

B. Machine Learning (ML) Algorithms

For determining the forward kinematic the machine learning algorithms used are Multiple Linear Regression (MLR), Artificial Neural Network (ANN) and Decision-Tree

(DT) Regression techniques. The following machine learning algorithms are trained and implemented of MATLAB.

1) Multiple Linear Regression (MLR)

The Multiple Linear Regression is defined as a straight forward regression technique for employing multiple variables in predicting the response or output variable. A connection between each joint angles and the end effector position variables is used in this technique. The equation general form for Multiple Linear Regression of multiple independent variables and single dependent variables is expressed as follows in equation (4).

$$y_i = b_{0i} + b_{1i} \cdot x_{1i} + \dots + b_{4i} \cdot x_{4i} \quad (4)$$

Here for our work the “ y_i ” represents the estimated end effector position of the 4 DOF robotic claw and x_1 to x_4 denote the four joint angles (θ) of the robot S1, S2, S3 and S4. The first joint S1 is of arm claw, the joint S2 and S3 is the elbow and shoulder joint respectively to move the arm to the desired position. The S3 joint is base joint to rotate the robot arm.

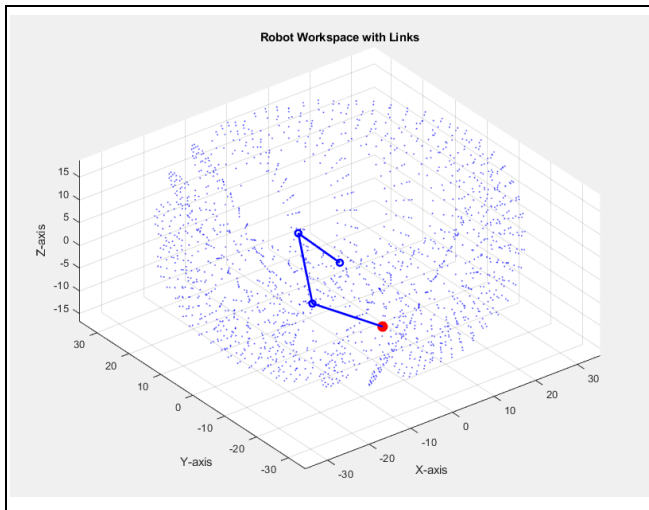


Fig. 3. Graphical Representation of the 4-Degree-of-Freedom Robot Workspace with Links demonstration on MATLAB

2) Decision-Tree (DT) Regression

It is the machine learning algorithm that is being used in predicting the target variable value by imposing the learning simple decision rules inferred from the data features. This technique works by recursively partitioning the feature values (parent/root node) and then fitting a simple model specifically a constant value within each subset (Decision Node). Figure (4) illustrates the simple representation of Decision Tree Regression in general for this work with root node, Decision node and leaf nodes.

The decision tree regression predicts the continuous target variable by taking the average of the target values of all the training instances within each and every leaf node.

To illustrate how decision tree regression can be used to predict the end effector position (x, y, z) of a 4 DOF robot with 4 joint angles as input, we will train the Decision Tree

Regression Model where the input features will be the joint angles ($(\theta_1, \theta_2, \theta_3, \theta_4)$) and the output variable is the end effector position (x, y, z). The Decision Tree algorithm will learn to predict the position of end effector on the basis of each joint angles. After that the model performance will be evaluated on the root mean squared error (MSE) and R2 (R-squared error) basis. Hence in the work the trained model can be used for the prediction. This prediction capability can be used in various robotics applications [8].

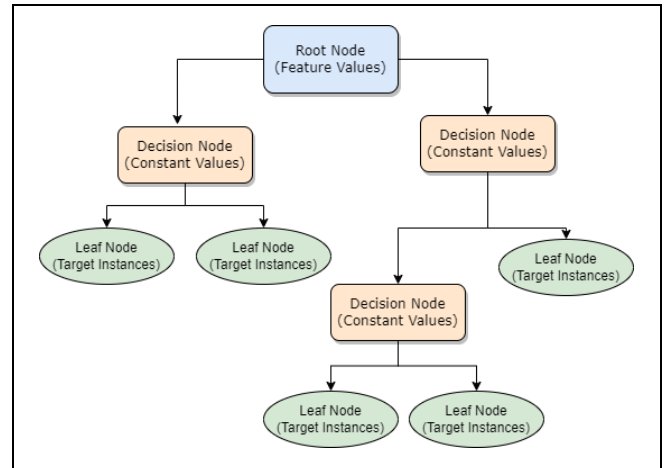


Fig. 4. A Simple Representation of Decision Tree used for Prediction of End Effector Position of the 4-Degree-of-Freedom Robot in Three Dimension Workspace.

3) Artificial Neural Network (ANN)

The ANN (Artificial Neural Network) is defined as computational model which come to existence after being inspired by the structure and the functioning of human brain biological neural networks. This technique comprises of the interconnected neurons, or nodes that are organized in layers ; output layer, input layer and hidden layer (if any). Each of the neuron receives inputs and preforms a desired computation and then pass out the result to the neurons in the next layer.

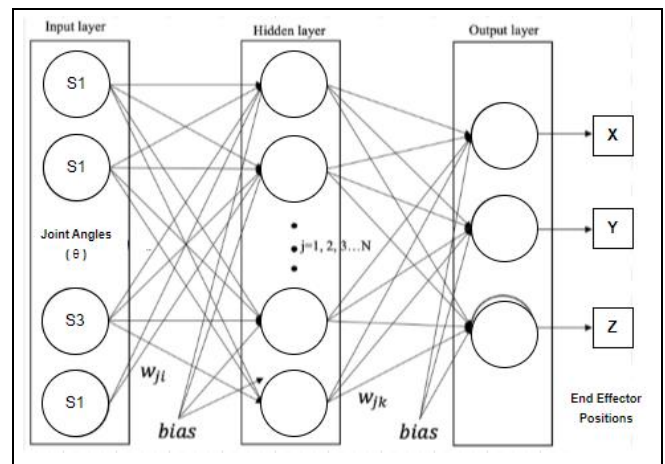


Fig. 5. A Simple Representation of Artificial Neural Network Topology used for Prediction of End Effector Position of the 4-Degree-of-Freedom Robot in Three Dimension Workspace.

The ANN can be used to predict end effector position of a 4-degree-of-freedom (DOF) robot on the basis of 4 joint angles. We train the neural network using the training data. During the training of an ANN the network adjusts its biases and weight iteratively for minimizing the predicted end effector position and the actual positions in the training data. This is typically be done by using Gradient Descent algorithm on the backend of all the ML algorithms. One the model got trained we will evaluate its performance on the ground basis of RMSE (Root Mean Squared Error).

The capability of ANN for capturing the complex and a non-linear relationship between the inputs and the outputs make it suitable for the prediction of end effector position of a 4 DOF robot based on joint angles. Just by adjusting the architecture and training parameters of neural network, the performance for specific prediction can be optimized. The ANN techniques is widely used in robots. The figure 5 shows the Artificial neural network topology used in this work.

During the ANN training, numerous parameters are adjusted, including hidden layer counts, neurons quality within each hidden layer and the activation function choice applied at both the hidden and outer layer. The activation functions like sigmoid, tanh, Linear, and the (ReLU) Rectified Linear Unit are employed during the training duration. More over optimization methods including the Stochastic Gradient Descent (SGD) and (Adam) adaptive moment estimation are utilized for refining the weights during the duration of training. The keen and careful selection of all these parameters leads to notable enhancement in prediction accuracy. The of epochs is set at 1,000 as a maximum number, as empirical evidence shows that the accuracy, as measured by metrics like (R2) R-squared and (RMSE) root mean square error, does not notably improve beyond this threshold. The result of training ANN on MATLAB environment for our work is depicted in Table 4.

TABLE IV. ANN MODEL TRAINING ON MATLAB PLATFORM

Data Division	Random
Performance	Mean Squared Error
Epoch	1000
Computational Time	0:00:13
Performance	0.000100
Gradient	0.266

IV. RESULTS AND DISCUSSIONS

The dataset is compiled comprising of 10,000 entries for both inputs (joint angle values) and outputs (end predictor values) which was subsequently utilized for training of the Machine Learning Algorithms. The end effector position actual value is calculated through analytical approach using the forward kinematics with DH parameters techniques. The predicted value is obtained from the machine learning algorithms implemented on MATLAB environment. The

performance of these algorithms is accessed by measuring the variance between the predicted and the actual value. However, the model's evaluation may vary depending on the random selection of samples within the training set, potentially resulting in either underestimation or overestimation.

The performance of every algorithm is evaluated on the basis of RMSE. Table 4 shows the values of end effector determined through (MLR) Multiple Linear Regression, (ANN) Artificial Neural Network (ANN) and Decision-Tree (DT) Regression. The actual value is also written. The values are found for the joint angle values. The visual representation for the actual and predicted end- effector value is shown in figure (7). The values given for actual and predicted end effector position is calculated on the basis of joint angles given in equation 5.

$$[\theta_1 \ \theta_2 \ \theta_3 \ \theta_4] = \left[0, \ -\frac{pi}{2}, \ -\frac{pi}{2}, \ 0 \right] \quad (5)$$

The dataset is compiled comprising of 10,000 entries for both inputs (joint angle values) and outputs (end predictor values) which was subsequently utilized for the training of the Machine Learning Algorithms. The actual value of the end effector position is calculated through analytical approach using the forward kinematics with DH parameters techniques.

TABLE V. ACTUAL AND PREDICTED VALUES OF END EFFECTOR POSITIONS OBTAINED FROM MATLAB PLATFORM

End Effector Position	X	Y	Z
Actual Value	26.841406	-10.000000	18.918461
Multiple Linear Regression (MLR)	17.735263	-3.969662	17.148879
Decision Tree Regression	28.214849	-12.180011	18.782643
Artificial Neural Network	26.871022	-11.481421	18.872894

The predicted value is obtained from the machine learning algorithms implemented on Here's a table summarizing the (R2) R-squared and (RMSE) Root Mean Squared Error values for each machine learning algorithm.

TABLE VI. ERROR ESTIMATION BETWEEN ACTUAL AND PREDICTED RESULTS OBTAINED FROM MATLAB PLATFORM

	Multivariable Linear Regression	Decision Tree Regression	Artificial Neural Network
RMSE (X)	9.105143	5.374741	0.030384
RMSE (Y)	6.030947	5.399014	1.481579
RMSE (Z)	0.000000	0.000000	0.045567
R2 (X)	0.628651	0.868257	0.999999
R2 (Y)	0.805418	0.840723	0.987424
R2 (Z)	1.000000	1.000000	0.999938

Table V shows the (RMSE) Root mean squared error and (R2) R-squared error for the (MLR) Multiple Linear Regression, (ANN) Artificial Neural Network and Decision-Tree (DT) regression which is found on MATLAB as a result of each algorithm. For Multiple Linear Regression (MLR) RMSE values are relatively high that indicates a noticeable deviation between the predicted and actual values. R2 values suggest moderate to good fit for the X and Y coordinates, but an excellent fit for the Z coordinate. Thus, MLR did not satisfies the demand and lags to capture the complex relationships between variables. Hence it had higher error as compared to other two algorithms used in this study.

For Decision-Tree Regression the RMSE values are lower in comparison with MLR. Thus, it performs well as it has ability to map non-linear relationships and handle interactions between the variable effectively.

For the Artificial Neural Network, the root mean squared error is extremely low as mentioned earlier in table V. So, there is a very small deviation between the actual and predicted value which can be easily being observed in Figure (7) in comparison with MLR and Decision-Tree Regression. R2 values are closed to 1 suggesting an excellent fit for all coordinates. Hence, it proves that for this work ANN excel in capturing the complex patterns and non-linear relationships, resulting in better performance compared to other algorithms to predict end effector position of 4 DOF on the basis of joint angles.

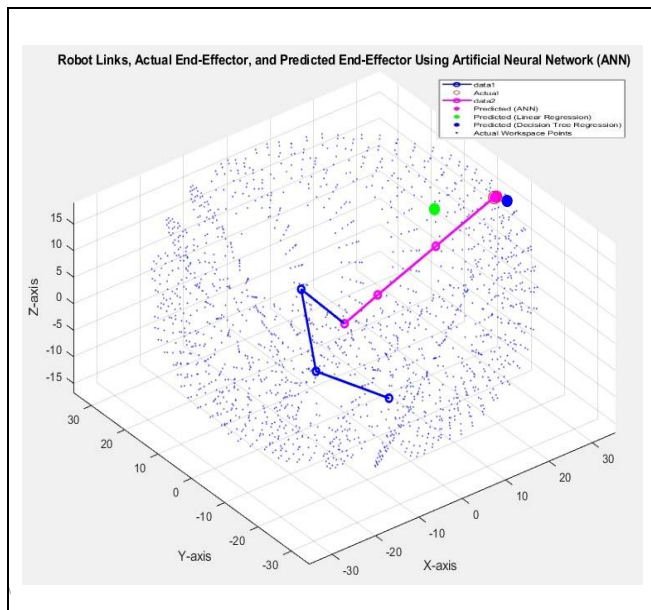


Fig. 6. Graphical Representation of Actual and Predicted End Effector Position of 4-Degree-of-Freedom through trained MLR, Decision-Tree Regression and ANN models.

In summary, the decision tree regression and artificial neural network models outperform multivariable linear regression in accurately predicting the end effector position. The decision tree regression model performs exceptionally well, achieving zero error for the Z coordinate. The artificial neural network model demonstrates outstanding performance with negligible errors across all coordinates, showcasing its effectiveness in handling complex data patterns for the work.

V. CONCLUSIONS

This study explores the machine learning algorithms to predict the operational area of 4-Degree-of-Freedom robot end effector position based on joint angles. The Forward Kinematics While Multiple Linear Regression shows moderate performance, Decision Tree Regression excels with lower errors. However, Artificial Neural Network emerges as the top performer, showcasing remarkable accuracy in predicting end effector positions. These findings highlight the potential of machine learning in enhancing robotics autonomy and task planning.

REFERENCES

- [1] Sharkawy, A.N. and Khairullah, S.S., 2023. Forward and Inverse Kinematics Solution of A 3-DOF Articulated Robotic Manipulator Using Artificial Neural Network. *International Journal of Robotics & Control Systems*, 3(2).
- [2] Carbajal-Espinosa, O., Campos-Macias, L. and Díaz-Rodríguez, M., 2024. FIKA: A Conformal Geometric Algebra Approach to a Fast Inverse Kinematics Algorithm for an Anthropomorphic Robotic Arm. *Machines*, 12(1), p.78.
- [3] Bayro-Corrochano, E., 2021. A survey on quaternion algebra and geometric algebra applications in engineering and computer science 1995–2020. *IEEE Access*, 9, pp.104326-104355.
- [4] Denavit, Jacques, and Richard S. Hartenberg. "A kinematic notation for lower-pair mechanisms based on matrices." (1955): 215-221.
- [5] Virgala, I., Kelemen, M., Varga, M. and Kurylo, P., 2014. Analyzing, modeling and simulation of humanoid robot hand motion. *Procedia Engineering*, 96, pp.489-499.
- [6] Gao, R., 2020. Inverse kinematics solution of Robotics based on neural network algorithms. *Journal of Ambient Intelligence and Humanized Computing*, 11(12), pp.6199-6209..
- [7] Soori, M., Arezoo, B. and Dastres, R., 2023. Artificial intelligence, machine learning and deep learning in advanced robotics, A review. *Cognitive Robotics*.
- [8] Kovincic, N., Gattringer, H., Müller, A. and Brandstötter, M., 2020, June. A boosted decision tree approach for a safe human-robot collaboration in quasi-static impact situations. In *International Conference on Robotics in Alpe-Adria Danube Region* (pp. 235-244). Cham: Springer International Publishing.

Hybrid Approach to solve Thermal Power plants Fuel cost Optimization using Ant Lion Optimizer with Newton based local search technique

Ejaz Ahmed^{1*}, Shahbaz Khan²,
Department Of Electrical Engineering Aero
Space and Aviation, Air University Kamra
Campus Attock, Pakistan
225258@aack.au.edu.pk,shahbaz@au.edu.pk

Abdul Wadood³,Husan Ali⁴
Department Of Electrical Engineering Aero
Space and Aviation, Air University Kamra
Campus Attock, Pakistan
Wadood@au.edu.pk,husan@au.edu.pk

Babar Sattar Khan⁵
Department Of Electrical and Computer
Engineering COMSATS University
Islamabad Attock CampusAttock,Pakistan
babar.sattar@ciit-attock.edu.pk

Abstract — *The optimization of the power system is a complicated problem that is extremely non-convex and nonlinear because of the many real-world constraints involved. Deterministic power system optimization strategies are unable to yield global optimal outcomes because the entrapment in local optimum zones. Stochastic approaches have been frequently used to handle these difficulties. Ant-Lion Optimizer is among them inspired by hunting behavior of Ant-loin such Biological, swarming, and ecology-based hybridization algorithms with local search methods SQP, IPA, and active set. These three are among common hybridized strategies that show promise for solving such challenges. Despite the fact that several metaheuristic algorithms are proposed for solving power system optimization problems the strength of hybridized global search-based techniques has not commonly been applied to power system optimization with economic load dispatch in particular. Results from findings proves for 3 and 13 bus system that hybridized-ALO outperforms modern optimizations methods.*

Keywords: Economic load dispatch; Energy; Ant-loin; Optimization; Metaheuristics.

I. INTRODUCTION

Due to the rising cost of producing electricity and the depletion of fossil fuels utilized in thermal power generating units, optimal Economic Load Dispatch (ELD) has gained significant attention in today's modern power system. The primary goal of the ELD problem is to distribute thermal power generating units' active power generation output as efficiently as possible while accounting for power system operational restrictions. The overall energy capabilities of electrical energy generation grow through the reduction of the generation cost and the enhancement of system reliability through optimum active power allocation. Because there are so many real-world power system scenarios, the research community is more interested in taking a realistic approach to solving the conventional ELD problem[1]. Due to the multilevel steam generating valves connected to contemporary steam powered thermal power generating units, also known as the valve point loading effect (VPLE), the input-output fuel cost generation curve is essentially non-differentiable, non-convex, and non-linear. These steam valves open systematically, which causes ripples in the fuel-cost characteristics curve. Conventional fossil fuel-based thermal power plants emit a variety of harmful gases (SO_x, NO_x, and CO_x) into the atmosphere, contributing to

environmental pollution and global warming[2]. Some of the generating unit shaft's bearings are subject to physical constraints known as prohibited operating zones (POZs) because of enhanced vibrations in specific areas along the rotating axis of the shaft. In recent years, there has been a lot of focus on renewable energy generation sources including while optimal power flow is taken into account for transporting electrical power over long distances to avoid power losses, optimal generation allocation and sizing are important factors in power system optimization for electrical power generation to lower fuel generation cost. Due to power losses from long-distance transmission lines and poor voltage regulation from a heavily loaded network, the efficiency of the power system effectively decreases. On the other hand, by lowering generation costs and power losses, taking into account the integration of renewable energy sources and their ideal placement inside conventional power systems can improve system reliability. Appropriate planning is required for the integration of renewable energy into conventional systems in order to prevent operational issues that could compromise system performance and dependability. In order for the power system to run efficiently and affordably, a number of generating units made up of thermal units renewable energy sources should be managed optimally considering practical constraints of real-world power system[3].Genetic algorithm was applied in [4] for economic load dispatch to reduce the fuel cost. With the incorporation of emissions the cost of production is increased [5] uses hybrid PSO with SQP to solve economic emission problem.

The main aim of our work is to reduce cost of production for three and thirteen units system incorporating hybrid methods.

II. MATHEMATICAL MODEL

The fuel cost and mathematical model of optimizer are described below.

A. FUEL COST EQUATION

Economic load dispatch problem is presented by quadratic equation. The values of cost coefficients can be taken from [11] fuel cost is related to power as:

$$F(P) = \sum (up^2 + vP + w) \quad (1)$$

In equation 1, $F(p)$ presents total generation cost in \$/hr whereas u , v and w are fuel cost equations. Includes loading on generators on value openings.

While second equation 2 models cost function with value point loading cost is multiplied with sin function.

$$f_{wv}(P) = \sum_{j=1}^{ng} (u_j p_j^2 + v_j P_j + w_j + |x_j \times \sin(y_j \times (P_j^{\min} - P_j))|) \quad (2)$$

B. ANT-LION OPTIMIZATION

ALO design got inspired from hunting style of Ant- lions. Ant lions are originated from myrmeleontide family. Ant-lions life span in adulthood lasts only for 3 to 5 weeks out of its 3 years age in which they prey rest of the life it reproduces its off spring. Ant- lions are known for their different style to prey ant's insects. Ant-lions digs out special cone (v) shaped trap in mud to hunt the ants[6]. The edges of cone are sharp Ant-lions tries to catch the prey with in trap range and place its self in middle of cone under sand. Ants tend to slip at sharp edges while ant lions through the sand grains at edge of the cone. When ant tries to leave trap during random movement, Ant-lions trap size is directly related to the hunger.

Operations of ALO

The ants move randomly in the trap and are modeled as

$$x(T) = [0, \text{cummsum}(2S(T^1) - 1), \dots, \text{cummsum}(2S(T^N) - 1)] \quad (3)$$

In the above equation

- $x(T)$ denotes moment of ants
- N shows total iterations number.
- T is for step of walk in random way.
- S is a random weight ranging from 0 to 1.

Initializing position matrix of ants

Random position matrix of ants is generated denoted by ANT^{pos} Each ant will move in different dimensions d and are equal to number of generators for ELD problem [5] also uses matrix, Position matrix of ants shown below

$$ANT^{pos} = \begin{pmatrix} a^{11} & a^{12} & a^{13} & \dots & a^{1d} \\ a^{21} & a^{22} & a^{23} & \dots & a^{2d} \\ \vdots & \vdots & \vdots & & \vdots \\ a^{n1} & a^{n2} & a^{n3} & \dots & a^{nd} \end{pmatrix}; \quad (4)$$

Fitness value calculation for ants

Each ant is passed through the required objective function that will return fitness value of each ant saved in column vector denoted by OA .

$$OA = \begin{pmatrix} oa^{11} \\ oa^{21} \\ \vdots \\ oa^{n1} \end{pmatrix}; \quad (5)$$

Fitness value calculation for ant-lions

Each ant is passed through the required objective function that will return fitness value of each ant saved in column vector denoted by OAL

$$OAL = \begin{pmatrix} oal^{11} \\ oal^{21} \\ \vdots \\ oal^{n1} \end{pmatrix}; \quad (6)$$

Random walk of ants

During optimization process each ant updates its position by adopting random walk in random direction. equation 1 cannot be directly adopted to update its position. Randomness of walk is normalized within range with specific constant.

Trapping in Ant-lions pits

Random walk of ants is affected by ant lion. Ants random walk in hyper sphere defined by vector e and d

$$e_d^t = Ant - lion_j^t + e^t \quad (7)$$

$$d_d^t = Ant - lion_j^t + d^t \quad (8)$$

e^t is minimal of variable d at t -th number iteration and d^t shows vector presenting maxima of variable at t -th number iteration.

Building trap

Each ant is trapped by single ant lion whose Selection is on fitness during optimization.

Sliding ant toward ant-lion and catching pray

Once ant comes in the range of trap ant-lions through sand to detract and slip ants toward center while ant tries to escape.

Elitism

Elitism is the solution which is best at any instant of optimizer running. Best ant-lions are considered as elite so every ant moves randomly around that ant-lion.

III. RESULTS AND DISCUSSION

First case is applied to 3 unit system having power demand of 850 MW Economic load dispatch with Value point loading shows total cost is 8234.07174 (\$/hr.) In case of three Units based ELD test system, Optimizer was set with initial setting of 15000 search agents. Each case is run on 20 independent trials and best results are shown in Figure 1. Complete simulation results are shown in Table I.

TABLE I. OPTIMIZED VALUES OF POWER AND FUEL PRICE FOR THREE UNITS USING A.L.O (Pd = 850 MWATT)

Unit	With V_P_L_EE	
	Best	Best
Unit 1 (MW)	300.26687	300.26687
Unit 2 (MW)	399.99999	399.99999
Unit 3 (MW)	148.73312	148.73312
Fuel Cost (\$/hr.)	8234.07174	8382.728
Total Power TP (MW)	850	850

Each generator have to produce specific power depending on coefficients with ALO optima alone results are shown in convergence curve it can be seen that after 100 iterations results are very close also test function is plotted.

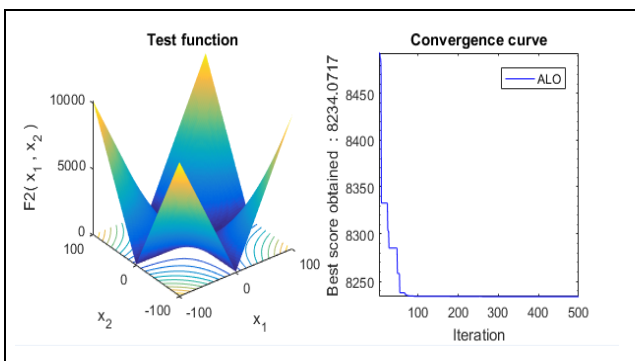


Figure 1: Convergence curve of A.L.O for three units

second case is applied to 13 unit system having power demand of 1800 MW Economic load dispatch with Value point loading shows total cost is 17934.3211 (\$/hr.) In case of 13 Units based ELD test system, Optimizer was set with initial setting of 15000 search agents and three dimension equaling

to number of generators. Each case is run on 20 independent trials and best results are shown in Figure 2. 1000 iterations were set but can be seen that after 100 iteration solution start to converge.

TABLE II. OPTIMIZED VALUES OF POWER AND FUEL PRICE FOR THREE UNITS USING A.L.O (Pd = 1800 MWATT)

Units	With V_P_L_EE
	Best
Unit 1 (MW)	548.3007
Unit 2 (MW)	260.8828
Unit 3 (MW)	235.9395
Unit 4 (MW)	90.0035
Unit 5 (MW)	101.4202
Unit 6 (MW)	98.2217
Unit 7 (MW)	99.5210
Unit 8 (MW)	88.5453
Unit 9 (MW)	87.1620
Unit 10 (MW)	38.0000
Unit 11 (MW)	41.0032
Unit 12 (MW)	56.0000
Unit 13 (MW)	56.0001
Total Power (MW)	1800.0000
fc (\$/hr.)	17934.3211

Complete simulation results are presented in Table II Moreover, the fine-tuning is carried out by Hybridizing procedures

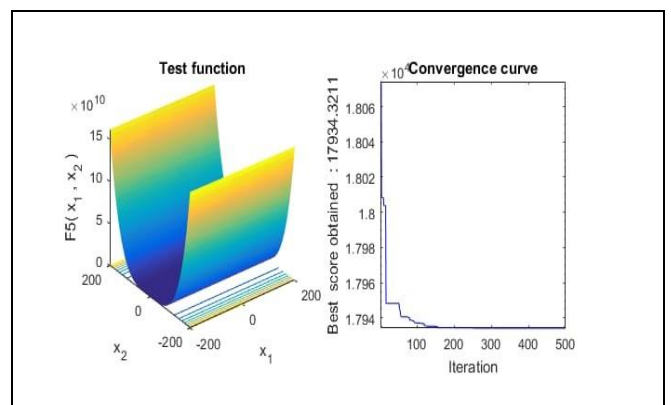


Figure 2: Convergence curve of A.L.O for thirteen units

IV. HYBERDIZATION WITH LOCAL SEARCH RESULTS

Hybridizing procedures ALO-SQP, ALO-ASA and ALO-IPA by taking the best results of ALO as a start point and continue with the refined values. Moreover,

results are summarized in Tables III for each scenario. Each three approaches are compared, one may observe that results of ALO-IPA are better in terms of convergence and accuracy while ALO-SQP gave better outcomes as tabulated in Table III considering ELD problem with VPLe.

TABLE III. HYBERDIZED WITH A.L.O 13 BUS (PD =1800 MWATT)

Unit	With V_P_L_EE		
	ALO-SQP	ALO-Active Set	ALO-IPA
Unit 1 (MW)	539.5587	538.6576	538.5587
Unit 2 (MW)	74.7998	79.2048	88.1070
Unit 3 (MW)	299.1993	300.1728	299.1993
Unit 4 (MW)	59.0000	60.0000	60.0004
Unit 5 (MW)	159.7331	180.0000	159.7332
Unit 6 (MW)	109.8666	109.8846	108.8666
Unit 7 (MW)	109.8666	109.8666	110.8666
Unit 8 (MW)	60.0000	60.0000	60.0004
Unit 9 (MW)	109.8666	60.0000	109.8666
Unit 10 (MW)	40.0000	40.0001	40.0005
Unit 11 (MW)	90.7094	114.8132	77.4000
Unit 12 (MW)	92.3999	92.4003	92.4002
Unit 13 (MW)	55.0000	55.0000	55.0005
Power Total	1800.0000	1800.0000	1800.0000
Fuel Cost	18118.1679	18558.4847	18122.5229

Each generator have to produce specific power depending on coefficients with ALO optima alone results are shown in convergence curve it can be seen that after 100 iterations results are very close also test function is plotted.

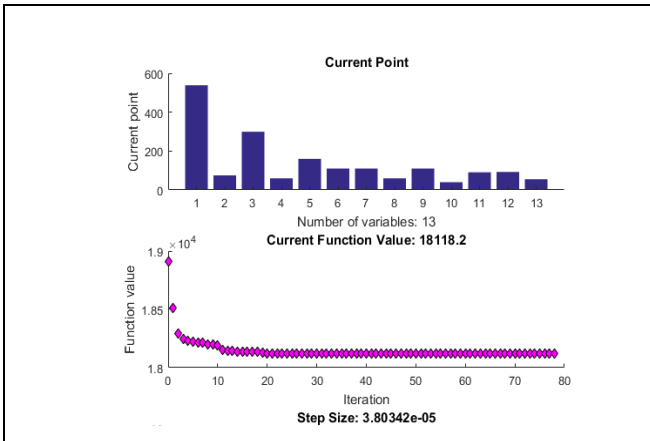


Figure 3: Simulated illustration of integrated ALO-SQP for thirteen units with V-P-L-E

Active set finds equality constraints in inequality constraints it can be seen that after 100 iterations results are very close also test function is plotted.

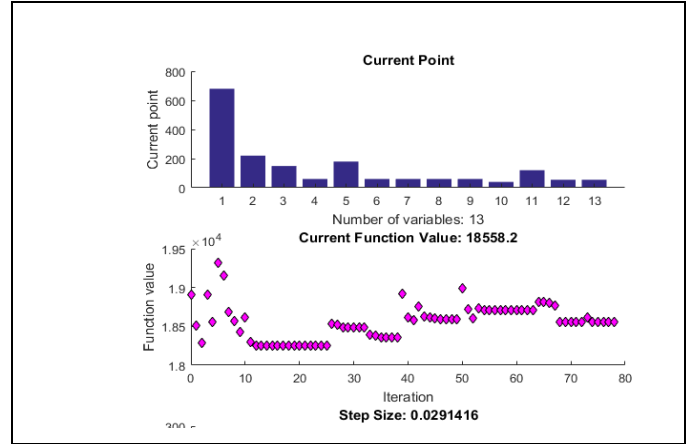


Figure 4: Simulated illustration of integrated ALO-Active Search for thirteen-unit system with V-P-L-E

IPA approach combines the best features of perturbation analysis and sequence quadratic programming in a unified framework it can be seen that after 100 iterations results are very close also test function is plotted.

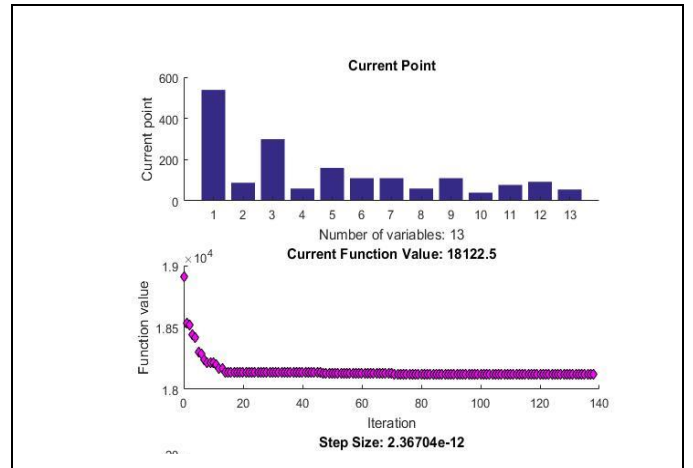


Figure 5: Simulated illustration of integrated ALO-IPA for thirteen-unit system with V-P-L-E

V. COMPARISON

The effectiveness of ALO is compared with latest four techniques for case of 13 units having power demand of 1800MW .These methods include teaching learning optimizer [7], Harmony search optimizer [8], Quazi oppositional inertial weight [9] and Novel Heuristic optimizer [10] while comparison on fuel cost for same operating constraints the results are shown in Table IV

TABLE IV. COMPARISON WITH STSTE OF ART METHODS

	TLBO [7]	H-S [8]	GPSO [9]	MPSO [10]	ALO
Unit 1 (MW)	364.9	628.3	628.3	628.2	548.30
Unit 2 (MW)	277.9	149.5	224.3	149.6	260.8
Unit 3 (MW)	217.4	222.7	148.7	222.7	235.9
Unit 4 (MW)	95.22	109.8	60	109.8	90
Unit 5 (MW)	106.6	60	109.8	60	101.4
Unit 6 (MW)	123.5	109.8	109.6	109.8	98.2
Unit 7 (MW)	112.5	109.8	60	109.8	99.5
Unit 8 (MW)	144.2	109.8	159.7	109.7	88.5
Unit 9 (MW)	126.7	109.6	109.5	109.8	87.1
Unit10 (MW)	60.23	40	40	40	38
Unit 11 (MW)	48.47	40	40	40	41
Unit 12 (MW)	91.36	55	55	55	56
Unit 13 (MW)	81.23	55	55	55	56
P total	1800	1800	1800	1800	1800
Cost (\$/hr.)	18141.2	17963.8	17978.6	17962.7	17934.3

AS seen from chart below teaching learning optimizer [7] have highest fuel cost of 18141.2 (\$/hr), Harmony search optimizer [8] is second, Quazi oppositional inertial weight [9] is third and Novel Heuristic optimizer [10] cost about 17962.7(\$/hr) while ALO performs best among all with least cost.

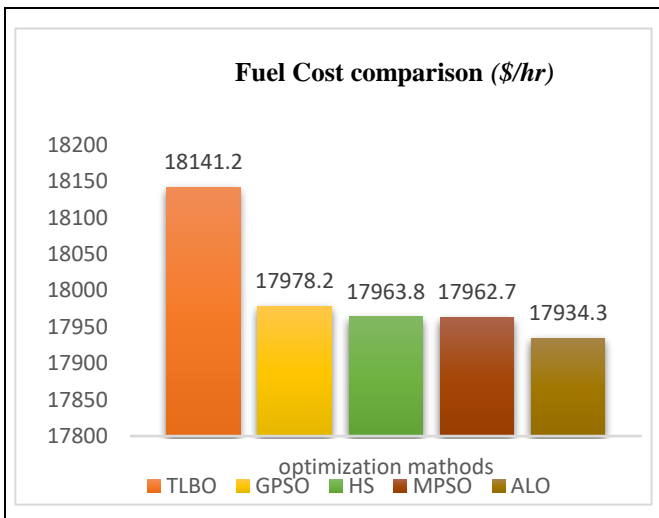


Figure 6: Fuel cost comparison chart

VI. CONCLUSIONS

This study explores the application of ALO with local search methods for economic load and emission dispatch, results of 3 generator systems shows strength of ALO also ALO is compared with four state of art optimizers in terms of fuel cost

for 13 generating units performance is best. With optimum allocation of these generators, not only cost is reduced but emissions are reduced Furthermore, this work can be extended by applying it to more big generating units and also problems related to reactive power compensation and generator scheduling for day ahead forecast. Also other factors like integration of wind and solar can be included in future work.

REFERENCES

- [1] Pradeep, A. and Sreekumar, C., 2022, July. Economic Load Dispatch augmented with Environmental Considerations. In *2022 Second International Conference on Next Generation Intelligent Systems (ICNGIS)* (pp. 1-5). IEEE.
- [2] Goyal, S.K., Singh, J., Saraswat, A., Kanwar, N., Shrivastava, M. and Mahela, O.P., 2020, June. Economic load dispatch with emission and line constraints using biogeography based optimization technique. In *2020 International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 471-476). IEEE..
- [3] Li, N., Uckun, C., Constantinescu, E.M., Birge, J.R., Hedman, K.W. and Botterud, A., 2015. Flexible operation of batteries in power system scheduling with renewable energy. *IEEE Transactions on Sustainable Energy*, 7(2), pp.685-696..
- [4] Chiang, C.L., 2007. Genetic-based algorithm for power economic load dispatch. *IET generation, transmission & distribution*, 1(2), pp.261-269..
- [5] Elaiw, A.M., Xia, X. and Shehata, A.M., 2013. Hybrid DE-SQP and hybrid PSO-SQP methods for solving dynamic economic emission dispatch problem with valve-point effects. *Electric Power Systems Research*, 103, pp.192-200..
- [6] Mirjalili, S., 2015. The ant lion optimizer. *Advances in engineering software*, 83, pp.80-98.
- [7] Banerjee, Sumit, Deblina Maity, and Chandan Kumar Chanda. "Teaching learning based optimization for economic load dispatch problem considering valve point loading effect." *International Journal of Electrical Power & Energy Systems* 73 (2015): 456-464.
- [8] Pandi, V.R., Panigrahi, B.K., Mohapatra, A. and Mallick, M.K., 2011. Economic load dispatch solution by improved harmony search with wavelet mutation. *International Journal of Computational Science and Engineering*, 6(1-2), pp.122-131.
- [9] Salaria, Umair Ahmad, Muhammad Ilyas Menhas, and Sohaib Manzoor. "Quasi oppositional population based global particle swarm optimizer with inertial weights (ppgpso-w) for solving economic load dispatch problem." *IEEE Access* 9 (2021): 134081-134095.
- [10] Singh, N., Chakrabarti, T., Chakrabarti, P., Margala, M., Gupta, A., Praveen, S.P., Krishnan, S.B. and Unhelkar, B., 2023. Novel heuristic optimization technique to solve economic load dispatch and economic emission load dispatch problems. *Electronics*, 12(13), p.2921.
- [11] Adriane, B.S., 2013. Cuckoo search for solving economic dispatch load problem. *Intelligent Control and Automation*, 2013.
- [12] Balbo, A.R., Souza, M.A.D.S., Baptista, E.C. and Nepomuceno, L., 2012. Predictor-corrector primal-dual interior point method for solving economic dispatch problems: a postoptimization analysis. *Mathematical Problems in Engineering*, 2012.

Smart Fire Safety: Real-Time Segmentation and Alerts using Deep Learning

Farhan Khan

*Department of Electrical Engineering
University of Engineering and Technology
Peshawar, Pakistan
farhankhan@uetpeshawar.edu.pk*

Sarmad Rafique

*Department of Computer Systems Engineering
University of Engineering and Technology
Peshawar, Pakistan
sarmadrafiqu.ncai@uetpeshawar.edu.pk*

Salman Khan

*Department of Electrical Engineering
University of Engineering and Technology
Peshawar, Pakistan
engrsalmankhan@uetpeshawar.edu.pk*

Laiq Hasan

*Department of Computer Systems Engineering
University of Engineering and Technology
Peshawar, Pakistan
laiqhasan@gmail.com*

Abstract—Fires are the major cause of property damage, injuries, and death worldwide. The ability to avoid or reduce the effects of fires depends on their early identification. The accuracy and responsiveness of conventional fire detection systems, such as smoke detectors and heat sensors, are constrained. Computer vision-based fire and smoke detection systems have been suggested as a replacement for conventional systems in recent years. To tackle the challenges a robust real-time framework has been proposed, whereby, images are taken from cameras and using a custom train YOLOv8 object segmentation model smoke and fires are localized in the image which are then fed to an expert system for alert generation. The expert system makes decisions on the fire status based on its size and growth across multiple frames. Furthermore, A new dataset was meticulously curated and annotated for the segmentation task, to assess the efficacy of the proposed system, comprehensive benchmarking was conducted on the proposed dataset using a suite of benchmarks. The proposed system achieved an mAP score of 74.9% on the benchmark dataset. Furthermore, it was observed that employing segmentation for localization as opposed to detection, resulted in system accuracy improvement. The system can immediately identify fires and smoke and send accurate alerts to emergency services.

Index Terms—Yolo-v8; Instance Segmentation; Fire and Smoke detection; Fire size; Fire spread; Emergency alert message; Arduino Uno.

I. INTRODUCTION

Fire has played a major role in the advancement of human society, but uncontrolled fires can lead to a significant loss of human life and property, so it is essential to prevent such types of fires because they can be widespread and result in huge losses. The National Fire Protection Association (NFPA) estimates that over 350,000 house-structure fires nationwide require the help of fire departments annually, with direct damages estimated to be around \$7 billion. In addition, there are 12,300 civilian fire injuries and about 2,500 civilian fire fatalities per year [1]. In Pakistan, fire has caused hundreds of casualties and infrastructural damage totaling billions of

rupees. One of Pakistan's most damaging fire disasters is thought to be the Baldia Town fire incident. However, the relevant government officials have not drawn any lessons from the situation [2]. Recently, a terrible fire in Lahore's renowned Hafeez Center destroyed hundreds of shops and caused traders to suffer severe losses [3]. A strong communication network is essential for the efficient operation of firefighting services. Today, many buildings have built-in smoke detectors and fire alarm systems as the most common fire detection method. These systems are activated when smoke from a fire rises and triggers sensors, usually located in the ceilings of the buildings. These sensors then activate the fire alarm and fire suppression systems however in 2018, 38% of fire alarms failed to sound when there was a fire, and 45% of these incidences were due to improper system positioning, according to the Home Office of the United Kingdom [4]. While this method is generally effective, there can be a delay between the smoke rising and hitting the sensor, which can allow the fire to spread quickly [5]. This delay should be minimized to prevent fires from getting out of control so traditional fire detection methods, including smoke detectors and heat sensors, may not provide the level of accuracy and speed necessary to quickly and effectively detect and respond to fires. In light of this, there has been a recent surge in the development of Deep Learning and computer vision-based fire and smoke detection systems as an alternative approach. These systems utilize cameras to gather visual data about the environment and apply machine learning techniques to analyze and detect fires and smoke. These Deep-learning techniques have been a major asset in the extraction of relevant features that best represent fires. Such methods have been applied to a wide range of fields, including image classification, autonomous vehicles, speech recognition, pedestrian detection, facial recognition, and cancer detection among others, showcasing their effectiveness in detecting and segmenting various object classes[6]. In this research, we propose a real-time fire and smoke segmentation detection system that uses the YOLO-v8 [7] object detection algorithm

and an emergency alert and danger level warning system. The proposed system is designed to detect fires and smoke in real-time, providing accurate and timely alerts to emergency services. The YOLO-v8 algorithm is trained on a custom-segmented dataset of fires and smoke to identify these phenomena in real-world environments. Additionally, the system employs color and texture-based features for segmenting and identifying the fire and smoke regions in the image. The system also includes a danger level warning and emergency alert that utilizes the size, location, and intensity of the fire or smoke to determine the level of emergency. The system can communicate the alerts to emergency services through email and push notifications. The results of the experiments demonstrate that the proposed system can detect fires and smoke in real-time with high accuracy and provide accurate and timely emergency alerts. The system also can segment the fire and smoke regions in the image, providing more detailed information about the emergency. We believe that the proposed system has the potential to significantly reduce the impact of fires on people and property. It serves as a valuable tool for emergency response and building safety.

II. LITERATURE REVIEW

Advances in AI, machine learning, and deep learning have fueled the widespread use of these technologies in fire and smoke detection. Souidene Mseddi et al. [8] proposed a fire detection model, combining YOLOv5 and U-net, achieving 99.6% accuracy. Ge Zhang et al.[9] enhanced YOLOv5 with a Swin transformation, improving feature fusion and achieving a 0.7% accuracy boost. Chen et al.[10] introduced a mixed-gaussian algorithm and YOLOv5-based smoke detection with 94.7% accuracy and 66.7 FPS speed. Sun et al.[11] addressed instance segmentation drawbacks with a semi-supervised technique and a lightweight SOLOv2 network, improving accuracy. Solorzano et al. [12] researched gas sensors for fire detection, demonstrating persistent predictive calibration models. Feiniu et al. [13] proposed a smoke segmentation model using CNN with VGG16 architecture. Jiao et al. [14] utilized Yolo v3 for UAV-based forest fire detection, achieving 83% accuracy at 3.2 frames per second. Hao Xu et al. [15] Innovative algorithm which was built using YOLOv5n. A SepViT Block was used to replace the model's final layer to strengthen the connection between the backbone network and the global information. A self-designed Light-BiFPN was also used to strengthen and lighten the network, reduce information loss, and improve accuracy and training convergence speed. Lastly, the Mish activation function was employed. For real-time fire detection on mobile devices, the Light-YOLOv5 significantly decreases the number of parameters and computations while increasing detection accuracy. Bhanumathi. M et al. [16] proposed image-based techniques for fire detection using surveillance cameras. They employ a background subtraction technique to detect fire using an RGB color pattern and motion detection technology. The technique seeks to spot fire quickly to protect people and property from its danger. The proposed system makes use of CCTV cameras to detect environmental changes brought on

by a fire. Fatma M. Talaat et al.[17] proposed a Smart Fire Detection System (SFDS), which is based on the YOLOv8 algorithm, is a transformative approach to fire detection in smart cities achieving a precision of 97.1%. One disadvantage in this is the possibility of including unnecessary items within the bounding box, making it difficult to identify the seriousness of a detected incident. This ambiguity might cause difficulties in discerning between circumstances that require immediate attention due to risk and those that may be less critical. SN Saydirasulovich et al.[18] presents an improved YOLOv8 model for UAV-based wildfire smoke detection, which addresses obstacles such as sluggish recognition and accuracy issues. The model includes Wise-IoU v3, Ghost Shuffle Convolution, and the BiFormer attention mechanism, resulting in a 3.3% improvement in average precision. Despite its success, the model is highly sensitive to atmospheric conditions, which can lead to false positives. Wahyono et al.[19] analyses Faster RCNN, Yolov4, and Yolov5 for fire detection in surveillance footage using datasets including FireNet, VisiFire, and FireSense. Yolov5 scores the best on the FireSense dataset, whereas Yolov4 excels with the highest TPR (84.62%) on VisiFire. However, including undesired background implies that more factors should be investigated in the future for better accuracy, particularly taking into account real-world camera constraints. Leibiao Hu et al.[20] proposed a novel YOLOv8 algorithm FSD-YOLOv8, trained on the FASDD dataset. Specifically designed for accurate flame and smoke detection. It incorporates a de-hazing stage for improved precision and uses dilated convolutions to extract complex features. FSD-YOLOv8 works better than traditional techniques. however, its accuracy is reduced in dynamic lighting and continuously smoke-filled environments. De Venancio et al.[21] presented a low-power automatic fire detection system utilizing YOLOv4. The suggested method reduces computational costs and memory consumption by up to an astounding 83.86%, respectively. The model achieved a f1 score of 72% and a map of 73.9% using the D-Fire dataset during training. X Xie et al.[22] Propose a YOLOv5-based real-time flame detection system for firefighting drones. A coordinate attention method, enhanced small-target recognition, and a novel loss function were introduced in the model to enhance the performance of the model. The experimental findings reveal an average precision of 96.6%, which is 5.4% higher than the original. The lightweight structure of the algorithm makes it suitable for firefighting drones and allows for fast identification. how every its accuracy reduces in detection at night, indicating potential directions for further study. Chayma Bahhar et al.[23] suggests a novel method for detecting forest fires in real-time by combining a classifier to increase precision and using an ensemble of two YOLO architectures (Yolov5s and Yolov5l). The FLAME dataset was used to train the model. The model performs better than the others, as evidenced by the trial results, which show 0.87% recall and 0.8% precision. The model is more resilient in a variety of forest fire conditions thanks to the ensemble architecture which reduces false positives.

In summary, this work uses an organized approach. Our

technique is described in Section III; the experimental setting is described in Section IV; the dataset used and evaluation measures are covered in Section V. Section VI presents the findings and debates, while Section VII wraps up this work.

III. METHODOLOGY

The proposed system attempts to improve building fire safety by installing a fire and smoke detection and alert system. To assess visual data from cameras mounted in the building, the system used the most recent fire detection algorithms. The YOLO-v8 segmentation detection algorithm is the central element of the suggested system which is used to examine visual data from the cameras. The alarm is activated by Arduino setup when fire or smoke is successfully detected in real-time through cameras. The alarm’s activation notifies the building’s residents of the fire emergency and began the water spray, both of which aid in containing the fire’s progress. The system also had a smoke sensor coupled to the Arduino setup. If there is smoke in the building, the sensor will detect it, which will cause the buzzer to sound and the water spray to begin. As a result of the smoke sensor’s integration, the system has a backup method of spotting fires. A danger level warning system that gives real-time updates on the severity of the fire emergency is another component of the planned system. Building occupants and emergency services can react to the fire emergency in an informed way thanks to the warning system, which is based on the size of the fire, the spreading of the fire, and the 1-minute time if the fire is still detected by the YoloV8 model. The proposed system sends emergency notifications to emergency services and building inhabitants via email. The notification give emergency services vital details about the location and severity of the fire, enabling them to react to the emergency swiftly and effectively. It is crucial to remember that the effectiveness and accuracy of the suggested approach depend on the camera’s accurate calibration. The whole methodology is shown through a block diagram in Figure 1.

A. YOLO-v8 Instance Segmentation

YOLOv8-seg (You Only Look Once version 8) is a cutting-edge object detection system that utilizes deep learning to achieve real-time results. It uses a unique network architecture to perform instance segmentation. The structure of yolo-v8 instance segmentation can be seen in Figure 2. YOLOv8-Seg is a significant improvement in the YOLO family, designed primarily for segmentation jobs. This model differs from its predecessors with an improved backbone structure that includes a 3 x 3 convolution, a C2f module, and an SPPF module. The C2f module replaces the basic 6 x 6 convolution, resulting in a lightweight architecture and improved gradient flow with skip connections and split operations. Notably, YOLOv8-Seg deviates from the traditional C3 module by incorporating improved cross-stage partial networks (CSP) for better residual connections. The head module demonstrates complicated feature fusion algorithms that include PANet and FPN, with major performance enhancements applied.

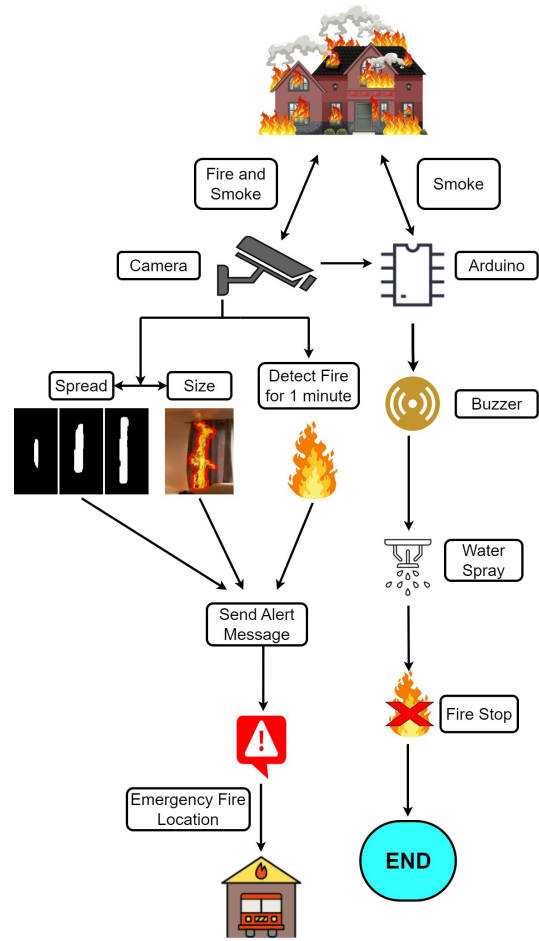


Figure 1: Methodology through block diagram

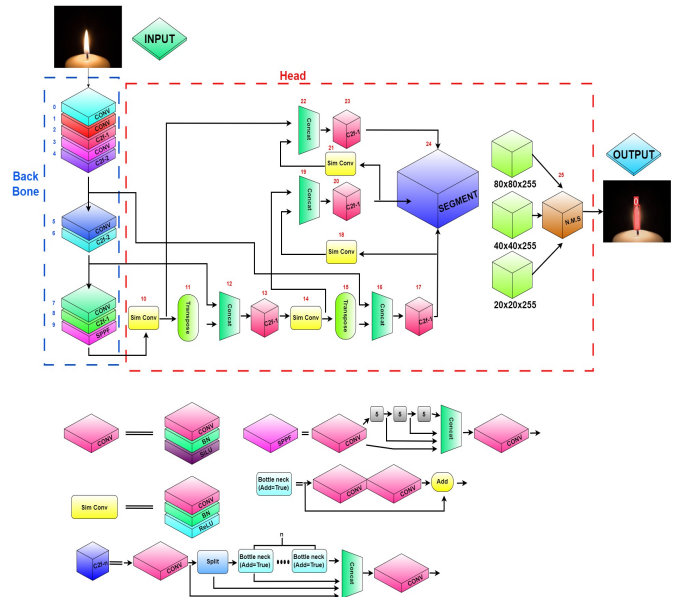


Figure 2: Represents all the layers present in the Yolo-v8 instance segmentation Architecture

IV. EXPERIMENTAL SETUP

A. Experimental Environment

The Experiment was conducted using the Google Colab platform.

B. Training

The training parameters for the fire and smoke detection model are shown in Table I. The fire and smoke data set used in the study was divided into a training set and a validation set, with the proportion being 80:20.

TABLE I: Training Parameter.

Parameter	Details
Picture size	640 x 640
Epochs	500
Batch size	16
Optimizer	SGD
Learning rate	0.01
Early stopping Patience	40
Multi-scale	50%
Momentum	0.937

C. System deployment

The Arduino Setup used in this paper is shown in Figure 3. Arduino UNO will receive information from both the camera and the MQ2 Gas Sensor. When fire or smoke is detected by the camera or smoke is sensed by the MQ2 sensor, Arduino will turn on the Red light and buzzer.

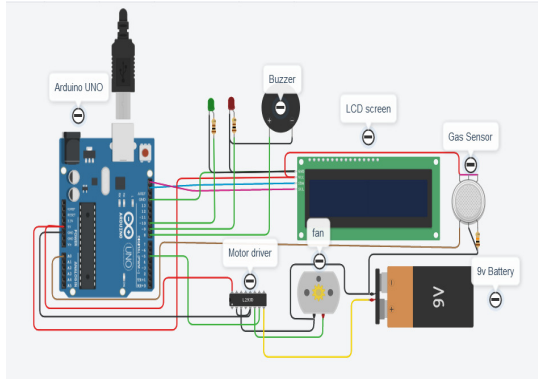


Figure 3: Represent the Experimental Setup

The primary components that have been used in this setup are:

- i. Arduino Uno
- ii. MQ2 Gas Sensor
- iii. I2C LCD Display Driver
- iv. 16x2 LCD display
- v. Passive Piezo Buzzer
- vi. Light-Emitting Diodes

V. EMPLOYED DATASET AND EVALUATION METRICS

A. Dataset:

In this paper, we proposed a novel data set for fire and smoke gathered from images and videos. The data set contains 892 pictures in total, divided into three groups fire, fire-smoke, and smoke. A wide variety of locations, including both indoor and outdoor scenes were represented in the data set. The instance segmentation task was performed on the data set using the Label-Me tool and the annotation files were saved in JSON format. To make the data set compatible with the Yolov8 framework, we wrote a code to convert the JSON format to a txt format. This data set can be utilized for a variety of tasks, such as object detection, semantic segmentation, and fire and smoke instance segmentation, thanks to its accurate annotations and wide set of photos depicting various situations. The data-set high quality and diversity made it easier to create more sophisticated algorithms and models for analyzing fire and smoke. Random samples from the data set are shown in Figure 4.

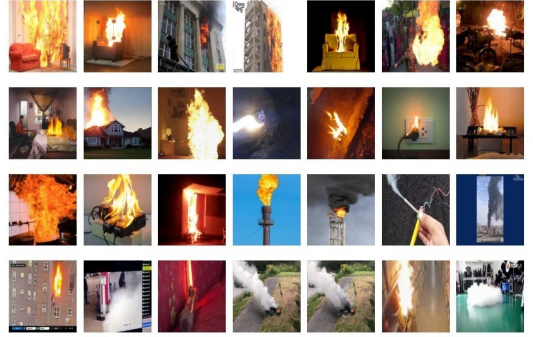


Figure 4: Illustrate fire and smoke images from over data-set

B. Evaluation Metrics

We assess the effectiveness of over model through:

i. F1-Score:

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

ii. Precision:

$$P = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

iii. Recall:

$$R = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (3)$$

iv. mAP50:

$$\text{MAP50} = \frac{1}{|X|} \sum_{i=1}^{|X|} \text{AvgP}(z_i) \quad (4)$$

Where:

- $|X|$ is the number of queries in the dataset
- z_i is the i-th query in the dataset
- $\text{AvgP}(z_i)$ is the average precision for the i-th query for the first 50 items in the ranked list

$$\text{AvgP}(z_i) = \frac{\sum_{Y=1}^{50} P(Y) * \text{rel}(Y)}{\sum_{Y=1}^{50} \text{rel}(Y)} \quad (5)$$

Where:

- $P(Y)$ is the precision at the Y -th item in the ranked list, limited to $Y = 1$ to 50
- $rel(Y)$ is a binary indicator of whether the Y -th item is relevant (1) or not relevant (0)

v. mAP50-90:

$$\text{MAP50-90} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \text{AvgP}(q_i) \quad (6)$$

Where:

- $|Q|$: is the total number of queries in the dataset.
- q_i : represents the i -th query in the dataset.
- $\text{AvgP}(q_i)$: Average precision for the i -th query, using only the top 50 to 90 ranked items.

VI. RESULTS AND DISCUSSION

The training outcomes of different models on the proposed dataset of fires and smoke are shown in Tables II and III. The training's outcomes revealed that the system achieved an f1-Score of 77.4% for Boxes and 75.7% for Masks. Every model performed well but YOLOv8 Large was particularly noteworthy as it produced the greatest results for masks in terms of mAP@50 and map50-90 achieving 74.9% and 51.0% respectively. The results of map50 for Boxes and Masks of all models are shown in Figures 5 and 6. The precision and recall results for boxes and masks were 86.4%, 70.2%, and 85.7%, 67.9% respectively. The fine-tuned YOLOv8 large model was incorporated into the system due to its better results in detecting fire and smoke. The real-time performance of the system can be seen in Figures 7, and 8, showing an excellent result in detecting fire and smoke. The emergency alert system was successfully implemented in the model and can send emergency messages if any of the three danger level conditions are satisfied as shown in Figure 9

TABLE II: Results of the Bounding Boxes

Model	Performance Evaluation metrics				
	F1 score	Precision	Recall	mAP-50	mAP50-90
YOLOv5-n	73.4%	82.3%	66.3%	72.0%	44.8%
YOLOv5-s	76.5%	90.2%	66.5%	74.8%	51.1%
YOLOv5-m	75.4%	85.5%	67.5%	74.0%	51.2%
YOLOv5-l	76.0%	84.5%	69.1%	75.6%	54.1%
YOLOv7	76.9%	86.0%	69.7%	77.9%	55.3%
YOLOv8-n	73.4%	85.0%	64.6%	72.5%	49.3%
YOLOv8-s	73.7%	86.0%	64.6%	72.4%	50.8%
YOLOv8-m	76.2%	86.1%	68.4%	76.5%	56.1%
YOLOv8-l	77.4%	86.4%	70.2%	76.8%	55.8%

TABLE III: Results of the Masks

Model	Performance Evaluation metrics				
	F1 score	Precision	Recall	mAP-50	mAP50-90
YOLOv5-n	70.8%	79.4%	63.9%	68.7%	39.8%
YOLOv5-s	74.5%	87.9%	64.7%	72.5%	43.6%
YOLOv5-m	73.9%	83.9%	66.1%	72.2%	44.3%
YOLOv5-l	74.3%	87.3%	64.8%	72.7%	44.4%
YOLOv7	74.7%	86.9%	65.6%	70.7%	41.7%
YOLOv8-n	73.3%	84.9%	64.5%	72.6%	47.5%
YOLOv8-s	73.5%	85.7%	64.4%	72.3%	48.1%
YOLOv8-m	75.0%	84.7%	67.4%	74.9%	48.9%
YOLOv8-l	75.7%	85.7%	67.9%	74.9%	51.0%

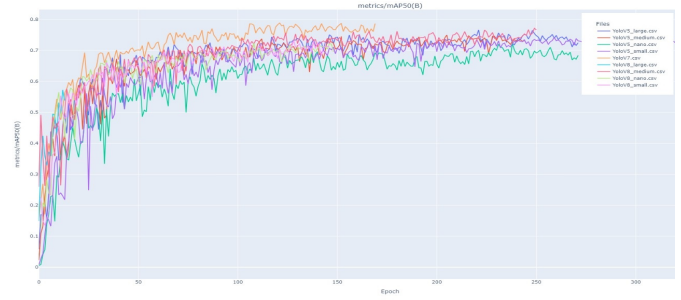


Figure 5: Model training curves of mAP0.5 metric for bounding boxes



Figure 6: Model training curves of mAP0.5 metric for segmentation masks

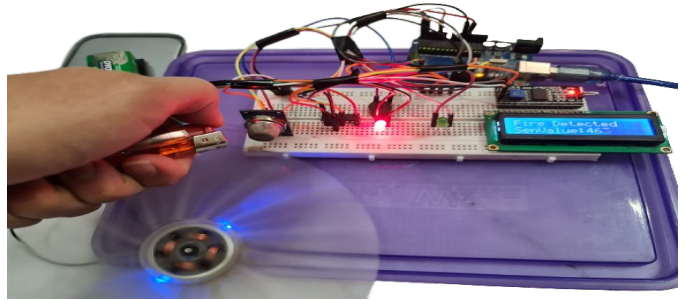


Figure 7: Show that when the smoke sensor detects smoke it starts the alarm and water spray

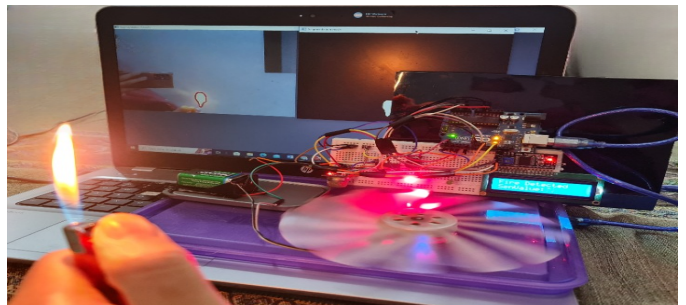


Figure 8: Show that when the Camera detects fire and smoke it starts the alarm and water spray

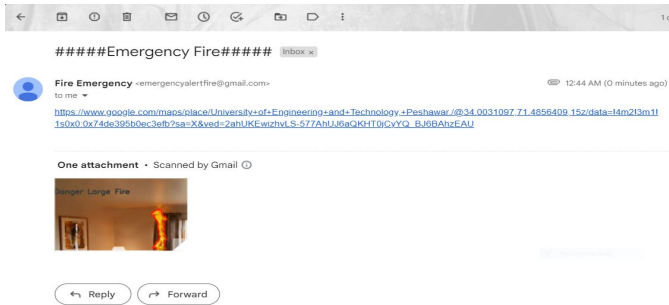


Figure 9: Show that when any danger level condition is satisfied an emergency alert message is sent

VII. CONCLUSION

To ensure public and building safety we proposed and implemented a system that uses the YOLO-v8 object segmentation detection technique for real-time fire and smoke detection. The system was created to instantly detect fire and smoke and send precise, timely alerts to emergency services and building occupants. The system was fitted with a hazard level warning system that was based on the size of the fire, spreading of fire, and 1 minute perennially detection of fire. The YOLO-v8 algorithm was trained on a novel instance-segmented data set. The trained algorithm achieves f1-Score of 75.7% and mAP50 of 74.9% for the Masks. The studies findings demonstrated that the suggested system could accurately and promptly provide emergency notifications while detecting fires and smoke in real time with high precision. The suggested system showed excellent potential in lessening the damage caused by fires to people and property, and it might be an important aid in emergency response and building safety. The novel instance segmented data set, the hazard level warning system, and the emergency alert system are the main contributions to this research. In the future, the system performance can be further enhanced by training the YOLO-v8 algorithm on a larger and more diverse data set. The system for alerting of danger levels and emergency alerts can be further improved by including more complex algorithms to forecast how the fire or smoke will develop as well as interacting with emergency response systems.

REFERENCES

- [1] Home Structure Fires report NFPA. (2021). <https://www.nfpa.org/News-and-Research/Data-research-and-tools/Building-and-Life-Safety/Home-Structure-Fires>
- [2] Iqbal Institute of Policy Studies. (2020). Fire Hazards and Firefighting in Pakistan. Islamabad: IIPS. Available at <https://www.facebook.com/watch/?v=984274088708480>
- [3] Dogar, A. (2020). 500 shops gutted in Lahore's Hafeez Centre fire. The News International. <https://www.thenews.com.pk/print/731453-500-shops-gutted-in-lahore-s-hafeez-centre-fire>
- [4] HOME OFFICE. (2020). FIRE RESCUE INCIDENT STATISTICS, ENGLAND, YEAR ENDING MARCH 2020. DANDY BOOKSELLERS Limited.
- [5] Woodford, Chris. (2007/2020) Smoke detectors. Retrieved from <https://www.explainthatstuff.com/smokedetector.html>. Accessed 20 Jan 2023
- [6] Zhao, Z. Q., Zheng, P., Xu, S. T., Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), 3212-3232.
- [7] Jocher, G., Chaurasia, A., Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. <https://github.com/ultralytics/ultralytics>
- [8] Mseidi, W. S., Ghali, R., Jmal, M., Attia, R. (2021, August). Fire detection and segmentation using YOLOv5 and U-net. In *2021 29th European Signal Processing Conference (EUSIPCO)* (pp. 741-745). IEEE.
- [9] Zhang, S. G., Zhang, F., Ding, Y., Li, Y. (2022). Swin-YOLOv5: Research and Application of Fire and Smoke Detection Algorithm Based on YOLOv5. *Computational Intelligence and Neuroscience*, 2022.
- [10] Chen, X., Xue, Y., Zhu, Y., Ma, R. (2023). A novel smoke detection algorithm based on improved mixed Gaussian and YOLOv5 for textile workshop environments. *IET Image Processing*.
- [11] Sun, G., Wen, Y., Li, Y. (2022). Instance segmentation using semi-supervised learning for fire recognition. *Heliyon*, 8(12), e12375.
- [12] Solórzano, A., Eichmann, J., Fernandez, L., Ziems, B., Jiménez-Soto, J. M., Marco, S., Fonollosa, J. (2022). Early fire detection based on gas sensor arrays: Multivariate calibration and validation. *Sensors and Actuators B: Chemical*, 352, 130961.
- [13] Yuan, F., Zhang, L., Xia, X., Wan, B., Huang, Q., Li, X. (2019). Deep smoke segmentation. *Neurocomputing*, 357, 248-260.
- [14] Jiao, Z., Zhang, Y., Xin, J., Mu, L., Yi, Y., Liu, H., Liu, D. (2019, July). A deep learning based forest fire detection approach using UAV and YOLOv3. In *2019 1st International conference on industrial artificial intelligence (IAI)* (pp. 1-5). IEEE.
- [15] Xu, H., Li, B., Zhong, F. (2022). Light-YOLOv5: A Lightweight Algorithm for Improved YOLOv5 in Complex Fire Scenarios. *Applied Sciences*, 12(23), 12312.
- [16] Bhanumathi, M., Ganesan, A., Kumar, K. R., Wellwin, K. J. (2021). Fire Detection and Alarm Using Gaussian Blur Background Subtraction Technique. *Turkish Journal of Computer and Mathematics Education*, 12(2), 929-934.
- [17] Talaat, F. M., ZainEldin, H. (2023). An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Computing and Applications*, 35(28), 20939-20954.
- [18] Saydirasulovich, S. N., Mukhiddinov, M., Djuraev, O., Abdusalomov, A., Cho, Y. I. (2023). An improved wildfire smoke detection based on YOLOv8 and UAV images. *Sensors*, 23(20), 8374.
- [19] Wahyono, W., Harjoko, A., Dharmawan, A., Kosala, G., Pranata, P. Y. (2021, December). A Comparison of Deep Learning Methods for Vision-based Fire Detection in Surveillance System. In *The 5th International Conference on Future Networks Distributed Systems* (pp. 1-7).
- [20] Hu, L., Lu, C., Li, X., Zhu, Y., Lu, Y., Krishnamoorthy, S. (2023, December). An enhanced YOLOv8 for flame and smoke detection with dilated convolution and image dehazing. In *Fourth International Conference on Signal Processing and Computer Science (SPCS 2023)* (Vol. 12970, pp. 604-608). SPIE.
- [21] de Venancio, P. V. A., Lisboa, A. C., Barbosa, A. V. (2022). An automatic fire detection system based on deep convolutional neural networks for low-power, resource-constrained devices. *Neural Computing and Applications*, 34(18), 15349-15368.
- [22] Xie, X., Chen, K., Guo, Y., Tan, B., Chen, L., Huang, M. (2023). A flame-detection algorithm using the improved YOLOv5. *Fire*, 6(8), 313.
- [23] Bahhar, C., Ksibi, A., Ayadi, M., Jamjoom, M. M., Ullah, Z., Souffene, B. O., Sakli, H. (2023). Wildfire and Smoke Detection Using Staged YOLO Model and Ensemble CNN. *Electronics*, 12(1), 228.

Modeling of Post-Myocardial Infarction and Its Solution Through Artificial Neural Network

Naheed Ali

Dept. of Basic Sciences and Islamiat
University of Engineering and Technology
Peshawar, Pakistan
naheedali581@gmail.com

Dr. Noor Badshah

Dept. of Basic Sciences and Islamiat
University of Engineering and Technology
Peshawar, Pakistan
noorbadshah@uetpeshawar.edu.pk

Abstract: Cardiovascular diseases, particularly myocardial infarction (MI) constitutes a significant health concern globally. A myocardial infarction, which is commonly known as a heart attack, happens when a part of the heart muscle doesn't get enough blood because of a blockage. Studying MI is complex and it requires looking at it from different angles. In recent years the fusion of mathematical modeling and artificial intelligence (AI) techniques has emerged as a promising avenue for understanding the complexities associated with MI. The primary goal of this study, is to provide an AI-based solutions of a new nonlinear mathematical model related to myocardial infarction phenomena. For obtaining the solution we will use a well-known deep learning technique, known as artificial neural networks (ANNs) with the combination of the optimization technique Levenberg-Marquardt back propagation (LMB). This combined method is referred to as ANNs-LMB. The results obtained from the model using ANNs-LMB are compared with a reference dataset constructed through the adaptive MATLAB solver ode45. The numerical performance is validated through a reduction in mean square error (MSE). The MSE is around 10^{-6} and the obtained results by ANNs-LMB are almost overlapped with the reference dataset, which show the accuracy and efficiency of the proposed methodology.

Keywords: Artificial Neural Network; Myocardial Infarction; Mathematical Modeling.

I. INTRODUCTION

Myocardial Infarction, often referred to as a heart attack, stands as a significant contributor to morbidity and mortality globally. Myocardial infarction causes 17.1 million deaths per year throughout the world [1]. Based on the latest statistics from the World Health Organization (WHO) the incidence of heart attacks in Pakistan, it was reported that 240,720 individuals lost their lives due to heart attacks in the year 2020. Smoking, inadequate physical activity, excessive body weight, elevated cholesterol levels, high blood pressure, and an unhealthy diet leading to elevated blood sugar are all factors that contribute to the risk of experiencing a myocardial infarction [2]. After a heart attack, the blockage in blood vessels stops oxygen and nutrients from reaching the heart muscle downstream. This causes damage to the heart, leading to a series of events like cell death, inflammation, and changes in the heart's structure, resulting in scars, stiffness, and altered function. People who've had a heart attack often face serious heart complications later on [3].

To response MI, the left ventricular (LV) of the heart will change its structural and functional behavior i.e. LV size, shape and function called the remodeling of LV [4]. Because of limited availability of the experimental data and biological complexity of the LV remodeling, the understanding of post

MI mechanism is a very complex process. By representing interaction of the factors such as blood flow, tissue oxygenation and cellular response through mathematical models can provide effective and valuable predictive capabilities. Previous studies [4, 5, 6], has explored the remodeling of left ventricular by using mathematical modeling. The approach employed in these articles involve using the numerical Runge-Kutta method with computer assistance to obtain the numerical solutions and collect various pieces of information. These investigations include exploring the important roles played by cytokines in the development of macrophages and other cells. Additionally, Zeigler et al. [7], has been investigated mathematical model for fibrosis, using an ordinary differential equations (ODEs) framework to predict behavior of collagen formation, breakdown, and aggregation. By using different assumptions, there exist some other papers [8, 9], include mathematical models to investigate the behavior of heart after myocardial infarction, but all have some specific limitations. Agent-based models [10, 11] have been employed to study tissue fibrosis, while biomechanical models [12, 13] are also present in the literature. However, there is a scarcity of studies focusing on ODEs models. Dealing with changes in heart after myocardial infarction is always a challenge.

Various disciplines, including health, biology, physics, chemistry, civil and mechanical engineering, and economics, extensively use mathematical models [14, 15, 16, 17, 18, 19]. In particular, there is a notable emphasis on combining these models with deep learning techniques, especially focusing on multilayer neural networks. In 2018, Side et al. [20] emphasized the crucial role of mathematical science in preventing the spread of illnesses. In addressing the spread of viruses, a mathematical model can be implemented, as in 2021 Umar et al. [21] highlighted the significant role of mathematics in exploring disease outbreaks, spread, and predictive patterns, particularly in the field of epidemiology. To obtain numerical outcomes for these models, stochastic solvers based on artificial neural networks along with optimization techniques are employed. Sabir et al. [22] presents applications of artificial neural network with the combination of Levenberg-Marquardt back propagation for the COVID-19 in 2022. In 2023 Haider et al. [23] proposed a system of ODEs for the study of hepatitis B virus (HBV) through deep learning techniques i.e. artificial neural network with the combination of Levenberg-Marquardt back propagation. In this paper, we apply a deep learning methodology, specifically leveraging a widely recognized approach known as artificial neural networks, to

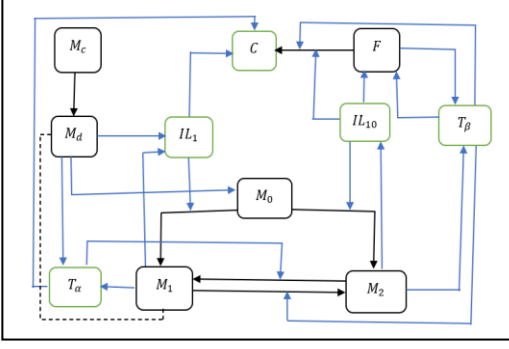


Figure 1: Diagram illustrating the cellular and molecular dynamics following a myocardial infarction. In the diagram, cells are depicted by boxes with black color, while green boxes contain cytokines and specific proteins. Two types of arrows are used: black arrows signify the physical transfer of cells between groups, for instance, the transition of M_1 to M_2 macrophages and vice versa. On the other hand, blue arrows denote interactions between distinct cell populations, like the release of cytokines by macrophages. The dotted black line represents the consumption rate of M_d by M_1 .

investigate the phenomenon of myocardial infarction. The goal of this study is to introduce numerical simulations of the remodeling of MI through a nonlinear system of ODEs, including different compartments of the post-MI phenomenon. The solutions of the system are obtained using artificial neural networks methodology supported by an optimization technique called Levenberg-Marquardt backpropagation. Furthermore, an analysis of various components of ANNs-LMB is conducted to assess the proposed methodology's efficacy in achieving high accuracy and optimal performance. This method is suggested as an artificial intelligence-based approach for solving complex types of ODE systems with known initial conditions [24, 25, 26, 27]. Some salient geographies of the designed study are given as follows:

- The MI mathematical model presented in this study, is the modification of mathematical model proposed by Lafci et al. [3]. We add two more cytokines: Transforming Growth Factor beta (T_β), and Tumor Necrosis Factor alpha (T_α), in the nonlinear system of ODEs proposed by Lafci et al. [3]. TGF- β is involved in the regulation of cell growth, differentiation, apoptosis, immune responses, and other cellular functions. It acts as a signaling molecule in various tissues and cell types, influencing both physiological and pathological processes [28] and is produced by alternatively activated macrophages (M_2) and fibroblasts (F). TNF- α is a cytokine involved in inflammation and immune system regulation. TNF-alpha is produced mainly by activated macrophages (M_1) and damaged cardiomyocytes (M_d) and can induce fever, inflammation, and cell death in certain tissues [29].
- The detailed descriptions of all compartments of the MI model are provided.
- The AI based solutions of the model are performed by using a deep learning technique, ANNs-LMB in MATLAB.

II. MATHEMATICAL MODEL

By incorporating T_β and T_α , and introducing some modifications to the mathematical model proposed in [3], our enhanced model is formulated as:

$$\frac{dM_c}{dt} = -k_1 M_c \quad (1)$$

$$\frac{dM_d}{dt} = k_1 M_c - k_2 M_1 M_d - \mu_1 M_d \quad (2)$$

$$\frac{dIL_1}{dt} = k_3 M_d + k_4 M_1 \frac{c_1}{c_1 + IL_1} - d_{IL_1} IL_1 \quad (3)$$

$$\frac{dIL_{10}}{dt} = k_5 M_2 \frac{c_2}{c_2 + IL_{10}} - d_{IL_{10}} IL_{10} \quad (4)$$

$$\frac{dM_0}{dt} = k_6 M_d - k_7 M_0 \frac{IL_1}{IL_1 + c_{IL_1}} - k_8 M_0 \frac{IL_{10}}{IL_{10} + c_{IL_{10}}} - \mu M_0 \quad (5)$$

$$\frac{dM_1}{dt} = k_7 M_0 \frac{IL_1}{IL_1 + c_{IL_1}} + \tau_1 M_2 \frac{T_\alpha}{T_\alpha + c_{T_\alpha}} - k_9 M_1 \frac{T_\beta}{T_\beta + c_{T_\beta}} - \mu M_1 \quad (6)$$

$$\frac{dM_2}{dt} = k_8 M_0 \frac{IL_{10}}{IL_{10} + c_{IL_{10}}} + k_9 M_1 \frac{T_\beta}{T_\beta + c_{T_\beta}} - \tau_1 M_2 \frac{T_\alpha}{T_\alpha + c_{T_\alpha}} - \mu M_2 \quad (7)$$

$$\frac{dC}{dt} = k_{10} F \frac{IL_{10}}{IL_{10} + c_3} + \alpha_1 F \frac{T_\beta}{T_\beta + \beta_1} - k_{11} C \frac{IL_1}{IL_1 + c_4} - \tau_2 C \frac{T_\alpha}{T_\alpha + c_4} - d_C C \quad (8)$$

$$\frac{dF}{dt} = k_{12} F \frac{IL_{10}}{IL_{10} + c_5} + \beta_2 F \frac{T_\beta}{T_\beta + \beta_1} - d_F F \quad (9)$$

$$\frac{dT_\beta}{dt} = \alpha_2 F + \alpha_3 M_2 - d_{T_\beta} T_\beta \quad (10)$$

$$\frac{dT_\alpha}{dt} = (\tau_3 M_1 + \tau_4 M_d) \frac{c_6}{c_6 + T_\beta} - d_{T_\alpha} T_\alpha \quad (11)$$

The model's parameters, along with their descriptions, values and units are listed in Table 1. This mathematical model captures the cellular and molecular dynamics associated with MI. It is derived from the depicted interactions in Figure 1, which serves as a flow diagram illustrating the system dynamics after MI, specifically focusing on scenarios of post MI without any medical interventions.

Equation 1 shows how the number of heart muscle cells (M_c) changes over time. Equation 2 explains how the number of damaged heart muscle cells (M_d) changes over time. It goes up as healthy cells M_c damage and decreases at rate of k_2 and μ_1 . Equation 3 illustrates how the concentration of interleukin 1 cytokines (IL_1) changes over time. These cytokines are released by both damaged heart muscle cells and a specific type of immune cells, classically activated macrophages (M_1). The impact of the inhibition by interleukin 10 cytokines (IL_{10}) is modeled as a decreasing function, where c_1 signifies the strength of inhibition. Equation 4 outlines the temporal evolution of IL_{10} . These

TABLE 1: MODEL'S PARAMETERS

Parameters	Description	Values
k_1	Death rate of M_c	0.3
k_2	Rate at which M_d are consumed by M_1	0.003
k_3	Rate of Secretion of IL_1 by M_d	0.0004
k_4	Rate of Secretion of IL_1 by M_1	0.0005
k_5	Rate of Secretion of IL_{10} by M_2	0.0005
k_6	Rate of recruitment of M_0 based on M_d	0.4
k_7	Activation rate of IL_1 to activate M_1	0.7
k_8	Rate of activation of IL_{10} to activate M_2	0.3
k_9	Rate of transition from state M_1 to M_2	0.075
k_{10}	C production rate by F	26×10^5
k_{11}	Degradation rate of C by IL_1	0.0003
k_{12}	Fibroblasts growth rate	0.25
c_1	Effectiveness of IL_{10} inhibition on IL_1	2.5
c_2	Effectiveness of IL_1 inhibition on IL_{10}	10
c_3	Effectiveness of IL_{10} inhibition on F	5
c_4	Effectiveness of IL_1 and T_α on C	10
c_5	Impact of promoting of IL_{10} on F	2.5
c_6	Effectiveness of T_β inhibition on T_α	0.0007
τ_1	Transition rate of M_2 to M_1 because of T_α	0.7
τ_2	Degradation rate of C by T_α	0.0003
τ_3	Rate at which M_1 produces T_α	0.0007
τ_4	Rate at which M_d produces T_α	0.000005
α_1	Stimulation rate of transition of F to C by T_β	10
α_2	Secretion rate of T_β by F	0.0167
α_3	Rate at which M_2 produces T_β	0.0144
β_1	Effectiveness of T_β promotion on F	0.00316
β_2	Stimulation rate of T_β on F	0.03
c_{IL_1}	Impact of promoting of IL_1 on M_1	10
$c_{IL_{10}}$	Impact of promoting of IL_{10} on M_2	5
c_{T_β}	Effectiveness of T_β promotion on M_2	0.00316
c_{T_α}	Impact of promoting of T_α on M_1	10
d_{IL_1}	Decay rate of IL_1 considering its half-life time	0.2
$d_{IL_{10}}$	Decay rate of IL_{10} considering its half-life time	0.2
d_c	Decay rate of C by some enzymes	0.002
d_f	Emigration rate of F	0.02
d_{T_β}	Decay rate of T_β considering its half-life time	4.06
d_{T_α}	Decay rate of T_α considering its half-life time	0.5
μ	M_0 , M_1 and M_2 emigration rate	0.2
μ_1	Removal rate of M_d	0.002

cytokines are released by specific type of immune cells, alternatively activated macrophages (M_2). A decreasing function is used to depict the inhibition of IL_{10} by IL_1 . Equation 5 explains how the quantity of monocytes (M_0) changes over time. It rises as because of M_d and declines due to two factors: the differentiation of M_0 into M_1 and M_2 , and a constant emigration rate. The transition of M_0 into M_1 is

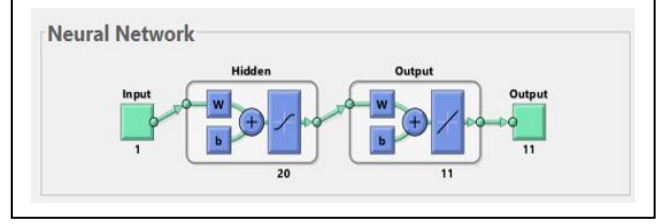


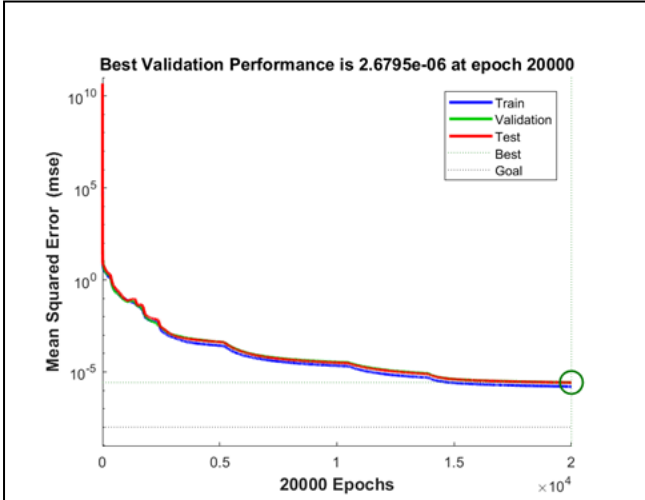
Figure 2: Designed ANNs-LMB

stimulated by interleukin 1, while the transition into M_2 is promoted by interleukin 10 cytokines. Equation 6 delineates how the density of M_1 changes over time. It increases when M_0 differentiates into M_1 and M_2 transferred into M_1 because of T_α , and decreases when M_1 transfer to M_2 by stimulation of T_β and emigration. Equation 7 portrays the temporal evolution of the density of M_2 . It increases when M_0 activates M_2 and when M_1 shifts to M_2 , and it decreases due to emigration. Equation 8 shows the variation in the density of collagen (C) over time. It increases as fibroblasts (F) produce collagen in response to stimulation by IL_{10} and T_β . On the other hand, it decreases due to degradation caused by the presence of IL_1 , T_α and a constant decay rate represented by d_c . Equation 9 characterizes how the density of fibroblasts changes over time. It increases through stimulation by IL_{10} and T_β but decreases due to death or emigration, represented by the rate d_f . Equation 10 details the rate of change of transforming growth factor- β over time. It is secreted by both F and M_2 . Equation 11 represents the change of T_α over time. It produces by M_1 and M_d with constant rates τ_3 and τ_4 respectively. A decreasing function is used for representing the inhibition of T_α by T_β . In this equation d_{T_α} shows the decay rate of T_α by considering its half-life time.

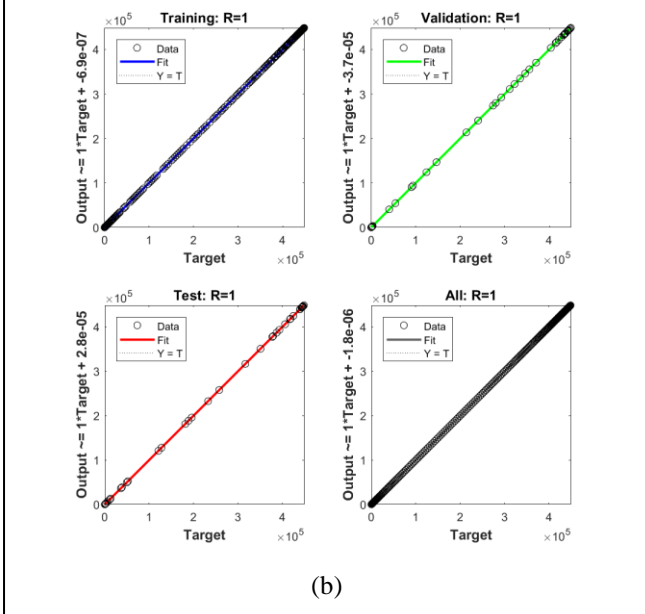
III. METHODOLOGY

The two-step approach to the stochastic numerical procedure for the MI mathematical model is used. The first step involves offering comprehensive and detailed explanations of the computational stochastic numerical procedure, which is centered on ANNs-LMB. The second step encompasses the implementation procedures that bolster the stochastic numerical computation for the MI nonlinear mathematical model. The implementation of ANNs-LMB is employed to analyze and utilize the computational stochastic numerical outcomes of the mathematical model for myocardial infarction. We use MATLAB for implementation of ANNs-LMB to obtain the results.

The ‘‘MATLAB’’ implementation follows a specific structure depicted in Figure 2, comprising a single input layer, hidden layers, and output layers. The configuration involves 20 hidden neurons, n-fold cross-validation, a log-sigmoid activation function, 20000 epochs, and the Levenberg-Marquardt optimization algorithm. It is important to highlight that the label data for input and targets are obtained from the standard numerical solution i.e. MATLAB solver command ode45.



(a)



(b)

Figure 3: The performance of the used methodology ANNs-LMB to solve the MI mathematical model is presented in (a). Error histogram, and regression measure through ANNs-LMB are shown in (b) and (c) respectively.

IV. NUMERICAL SIMULATIONS

The parameters used in the numerical simulations are presented in Table 1, providing descriptions and values for each parameter. The initial values of all compartments used in the model are illustrated in Table 2.

Numerical outcomes for the nonlinear dynamical model of myocardial infarction within the input range $[0, 60]$ are obtained through the ANNs-LMB methodology. The results are generated by using MATLAB and depicted in Figures 3-6.

The graphs illustrating calculated results for the MI mathematical model is presented in Figures 3-6. In particular,

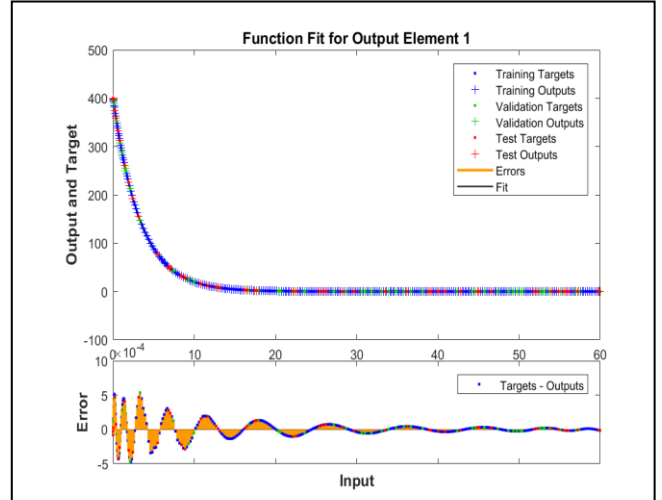


Figure 4: Function fit for output

In particular, Figure 3 offers insights into the performance, error histogram with 20 bins, and regressions of the applied methodology. Specifically, Figure 3(a) displays the calculated mean square error, measures for the best curves during training, validation, and testing with optimal performance achieved at epoch 20000, which is 2.6795×10^{-6} . These visual representations underscore the successful convergence and precision achieved by used methodology. Moreover, in Figure 3(c), correlation measures are presented, highlighting the regression performances. The correlation performances, expressed as the coefficient of determination (R^2 values), predominantly approach 1, underscoring the precision in solving the model. These plots encompass training, validation, testing and collectively indicating the accuracy of the scheme. Finally, fitting curves are depicted in Figure 4 to show the comparison between training, validation, and testing of the used methodology.

Figure 5 and 6 displays comparison plots of the solutions obtained by using the ANNs-LMB methodology and the true solutions (reference dataset constructed through MATLAB solver ode45) for the nonlinear dynamical system associated with myocardial infarction. Figure 5(a) compares the ANNs-LMB solution with the exact solution of the cardiomyocytes. We can observe that the solution obtained by ANNs-LMB and the exact solution are almost overlapped. Similarly, comparisons of dead cardiomyocytes, monocytes, macrophages and fibroblasts are presented in Figure 5(b)-5(f). The comparison of cytokines and proteins after myocardial infarction are presented in Figure 6. These plots reveal a nearly perfect overlap between the exact solutions and those obtained by ANNs-LMB, underscoring the precision and effectiveness of the designed ANNs-LMB in solving the nonlinear system of differential equations related to the phenomena of myocardial infarction.

V. DISCUSSION

A mathematical model is constructed to encompass critical interactions among cardiac cells, immune responses, and matrix proteins following myocardial infarction. Our

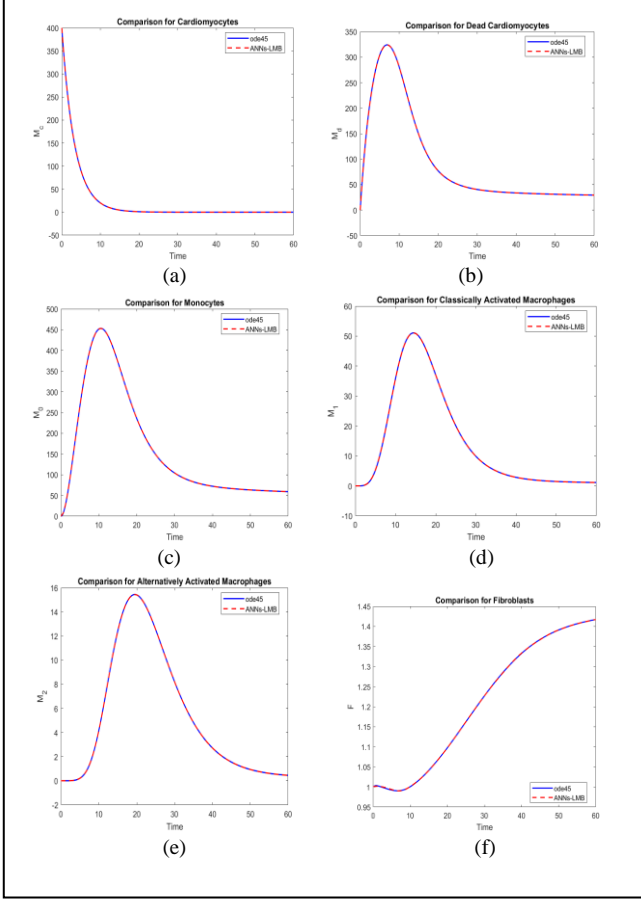


Figure 5: Dynamical behavior of cells after MI

model represents a notable improvement over earlier mathematical models, as highlighted in the work by Lafci et al. [3]. It stands out for considering significant biological factors, explicitly addressing the change of cardiomyocytes, the behavior of fibroblasts, and the deposition of fibrotic collagen in the context of post-myocardial infarction. Through numerical simulations, we tested the model's ability to describe events following a heart attack. The model's accurate predictability enhances our understanding of left ventricular remodeling after a heart attack. To derive solutions for the model, we used the deep learning strategy known as ANNs-LMB.

Many papers on mathematical models lack visual representations, such as plots, that illustrate the evolution of various populations and species over time. While notable exceptions include the works of Jin et al. [4] and Wang et al. [5], Lafci et al.'s paper [3] stands out by providing insightful plots of evolution. To enhance understanding, we introduced changes to Lafci et al.'s model, particularly by adding two new compartments, TGF- β and TNF- α , and then solving the modified model by using deep learning strategy, specifically ANNs-LMB.

Existing models related to MI, capture various aspects but often neglect key components. For example, Wang's model considers the monocytes and macrophages

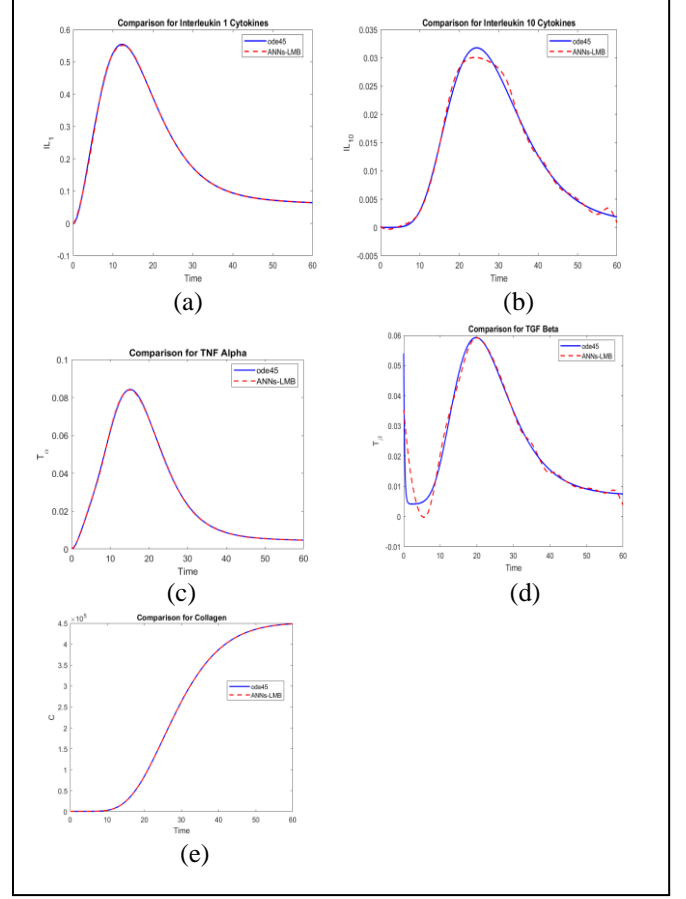


Figure 6: Dynamical behavior of cytokines and proteins after MI

relationship, inhibitory and synthesizing bio factors, yet overlooks collagen, fibroblasts, and cardiomyocytes. Jin's model includes multiple elements but omits the behavior of cardiomyocytes. Zeigler et al. [7] primarily focus on fibroblast and collagen concentrations post-MI. Our study distinguishes itself by incorporating two additional compartments into the existing model proposed by Lafci et al., which plays a role in left ventricular remodeling after a myocardial infarction. Furthermore, we obtained solutions for the myocardial infarction mathematical model by applying the deep learning strategy ANNs-LMB. Despite its thoroughness, a limitation stems from the lack of clinical data to derive certain unknown parameters. This model's limitation can be addressed in the future with more detailed data, allowing for a more precise representation of post-MI biological processes. Additionally, in this work we used eleven compartments of the MI phenomena, a simplification compared to the real-world scenario. Future improvements may involve including more compartments to reduce this limitation.

VI. CONCLUSIONS

The objective of the current study is to apply deep learning strategies for the investigation of myocardial infarction through mathematical modeling. Several parameters in the mathematical model proposed by Lafci et al., for myocardial infarction are modified. The primary

alteration involves the addition of two more compartments, namely Transforming Growth Factor beta and Tumor Necrosis Factor alpha. Results of the nonlinear dynamical MI mathematical model are obtained by using a deep learning technique ANNs-LMB. The mathematical model is dependent on eleven dimensions.

Validation, testing, and training processes are conducted utilizing ANNs-LMB for the MI mathematical model. The numerical solutions derived from the model are compared with a reference dataset constructed through MATLAB. The outcomes demonstrate a notable overlapping with the reference dataset, underscoring the accuracy of the used methodology. Additionally, the results are further validated through the reduction of MSE. To evaluate the precision, reliability, and efficiency of the approach, various analyses, including MSE, error histogram and regressions are used in this study.

REFERENCES

- [1] T. Siddiqui, D. Dikshit, *Platelets and atherothrombosis: causes, targets and treatments for thrombosis*, Current medicinal chemistry 20 (22) (2013) 2779–2797.
- [2] R. Hajar, *Risk factors for coronary artery disease, historical perspectives*, Heart views: the official journal of the Gulf Heart Association 18 (3) (2017) 109.
- [3] M. Lafci, G. K. Sabine, H. V. Kojouharov, B. M. Chen-Charpentier, S. R. McMahan, J. Liao, *Using models to advance medicine: mathematical modeling of post-myocardial infarction left ventricular remodeling*, Computer Methods in Biomechanics and Biomedical Engineering 25 (3) (2022) 298–307.
- [4] Y. F. Jin, H. C. Han, J. Berger, Q. Dai, M. L. Lindsey, *Combining experimental and mathematical modeling to reveal mechanisms of macrophage-dependent left ventricular remodeling*, BMC systems biology 5 (1) (2011) 1–14.
- [5] Y. Wang, T. Yang, Y. Ma, G. V. Halade, J. Zhang, M. L. Lindsey, Y. F. Jin, *Mathematical modeling and stability analysis of macrophage activation in left ventricular remodeling post-myocardial infarction*, BMC genomics 13 (6) (2012) 1–8.
- [6] U. Pagalay, L. Handayani, A. Azzam, *Dynamics of macrophages and cytokines after myocardial infarction*, Conference: 1st International Conference on Science and Technology, ICOST 2019, 2-3 May, Makassar, Indonesia.
- [7] A. C. Zeigler, A. R. Nelson, A. S. Chandrabhatla, O. Brazhkina, J. W. Holmes, J. J. Saucerman, *Computational model predicts paracrine and intracellular drivers of fibroblast phenotype after myocardial infarction*, Matrix Biology 91 (2020) 136–151.
- [8] L. Lee, G. Kassab, J. Guccione, *Mathematical modeling of cardiac growth and remodeling*, Wiley Interdisciplinary Reviews: Systems Biology and Medicine 8 (3) (2016) 211–226.
- [9] N. Moise, A. Friedman, *A mathematical model of immunomodulatory treatment in myocardial infarction*, Journal of Theoretical Biology 544 (2022) 111–122.
- [10] A. D. Rouillard, J. W. Holmes, *Mechanical regulation of fibroblast migration and collagen remodelling in healing myocardial infarcts*, The Journal of physiology 590 (18) (2012) 4585–4602.
- [11] S. M. Rikard, T. L. Athey, A. R. Nelson, S. L. Christiansen, J.-J. Lee, J. W. Holmes, S. M. Peirce, J. J. Saucerman, *Multiscale coupling of an agent-based model of tissue fibrosis and a logic-based model of intracellular signaling*, Frontiers in physiology 10 (2019) 1481.
- [12] P. Saez, E. Kuhl, *Computational modeling of acute myocardial infarction*, Computer methods in biomechanics and biomedical engineering 19 (10) (2016) 1107–1115.
- [13] E. Cutri, A. Meoli, G. Dubini, F. Migliavacca, T. Y. Hsia, G. Pennati, *Patientspecific biomechanical model of hypoplastic left heart to predict post-operative cardiocirculatory behaviour*, Medical Engineering & Physics 47 (2017) 85–92.
- [14] M. L. Schiavo, B. Prinari, J. A. Gronski, A. V. Serio, *An artificial neural network approach for modeling the ward atmosphere in a medical unit*, Mathematics and Computers in Simulation 116 (2015) 44–58.
- [15] J. L. G. Guirao, *On the stochastic observation for the nonlinear system of the emigration and migration effects via artificial neural networks*, International Journal of Mathematics and Computer in Engineering 1 (2) (2023) 177–186.
- [16] M. Shoab, R. Kainat, M. Ijaz Khan, B. Prasanna Kumara, R. Naveen Kumar, M. A. Zahoor Raja, *Darcy-forchheimer entropy based hybrid nanofluid flow over a stretchable surface*, intelligent computing approach, Waves in Random and Complex Media (2022) 1–24.
- [17] Z. Sabir, R. Sadat, M. R. Ali, S. B. Said, M. Azhar, *A numerical performance of the novel fractional water pollution model through the levenberg-marquardt backpropagation method*, Arabian Journal of Chemistry 16 (2) (2023).
- [18] T. Botmart, Z. Sabir, M. A. Z. Raja, R. Sadat, M. R. Ali, *Stochastic procedures to solve the nonlinear mass and heat transfer model of williamson nanofluid past over a stretching sheet*, Annals of Nuclear Energy 181 (2023).
- [19] W. Weera, C. Zamart, Z. Sabir, M. A. Zahoor Raja, A. S. Alwabli, S. Mahmoud, S. Wongaree, T. Botmart, *Fractional order environmental and economic model investigations using artificial neural network*, Computers, Materials & Continua 74 (1) (2023).
- [20] S. Side, Y. M. Rangkuti, D. G. Pane, M. S. Sinaga, *Stability analysis susceptible, exposed, infected, recovered (seir) model for spread model for spread of dengue fever in medan*, in: Journal of Physics: Conference Series, Vol. 954, IOP Publishing, 2018, pp. 012–018.
- [21] M. Umar, Z. Sabir, M. A. Z. Raja, S. Javeed, H. Ahmad, S. K. Elagen, A. Khames, *Numerical investigations through anns for solving covid-19 model*, International Journal of Environmental Research and Public Health 18 (22) (2021).
- [22] Z. Sabir, M. A. Z. Raja, S. E. Alhazmi, M. Gupta, A. Arbi, I. A. Baba, *Applications of artificial neural network to solve the nonlinear covid-19 mathematical model based on the dynamics of sir*, Journal of Taibah University for Science 16 (1) (2022) 874–884.
- [23] Q. Haider, A. Hassan, S. M. Eldin, *Artificial neural network scheme to solve the hepatitis b virus model*, Frontiers in Applied Mathematics and Statistics 9 (2023).
- [24] Z. Sabir, T. Botmart, M. A. Z. Raja, R. Sadat, M. R. Ali, A. A. Alsulami, A. Alghamdi, *Artificial neural network scheme to solve the nonlinear influenza disease model*, Biomedical Signal Processing and Control 75 (2022).
- [25] S. Mall, S. Chakraverty, *Comparison of artificial neural network architecture in solving ordinary differential equations*, Advances in Artificial Neural Systems 2013 (2013) 12–12.
- [26] Y. Wen, T. Chaolu, X. Wang, *Solving the initial value problem of ordinary differential equations by lie group based neural network method*, Plos one 17 (4) (2022).
- [27] N. Yadav, A. Yadav, M. Kumar, *An introduction to neural network methods for differential equations*, Vol. 1, Springer, 2015.
- [28] M. Bujak, N. G. Frangogiannis, *The role of tgf- β signaling in myocardial infarction and cardiac remodeling*, Cardiovascular research 74 (2) (2007) 184–195.
- [29] D. Jang, A. H. Lee, H. Y. Shin, H. R. Song, J. H. Park, T. B. Kang, S. R. Lee, S. H. Yang, *The Role of Tumor Necrosis Factor Alpha (TNF- α) in Autoimmune Disease and Current TNF- α Inhibitors in Therapeutics*, International Journal of Molecular Sciences, Korea, 2021.
- [30] C. Ravaut, N. Ved, D. G. Jackson, J. M. Vieira, P. R. Riley, *Lymphatic clearance of immune cells in cardiovascular disease*, Cells 10 (10) (2021) 25–94.

Honey Adulteration Detection through Hyperspectral Imaging And Machine Learning

Hazrat Usman

Department of Computer Science,
CECOS University of IT and Emerging
Sciences,

Peshawar, KPK, Pakistan

hazrat.usman.bsse-2020a@cecosian.edu.pk

Anna Amjad

Department of Computer Science,
CECOS University of IT and Emerging
Sciences,

Peshawar, KPK, Pakistan

anna.amjad.bsse-2020a@cecosian.edu.pk

Maryam Mahsal Khan

Department of Computer Science,
CECOS University of IT and Emerging
Sciences,

Peshawar, KPK, Pakistan

maryam.khan@cecos.edu.pk

Abstract— The purity and authenticity of honey are paramount for ensuring consumer trust and maintaining the integrity of the honey industry. There is a pressing need for advanced and efficient detection method for increasing prevalence of honey adulteration. A publicly available dataset with spectral features, extracted through hyperspectral imaging, across different classes of honey and adulteration level has been examined and various machine learning models were developed to identify honey adulteration concentration and type of honey. The dataset was balanced and a five-fold cross validation technique was used to train the machine learning models. Random forest was found to perform better with an accuracy of 99.69% than the one reported in literature (95%). The proposed approach aims to provide an accurate and cost-effective solution to address the challenges associated with honey adulteration, contributing to the enhancement of honey quality assessment and consumer confidence.

Keywords: Honey Fraud Detection; Hyperspectral imaging; Machine Learning.

I. INTRODUCTION

Honey, a natural product cherished for its nutritional benefits and distinct flavor. The rise in demand for honey has created a lucrative market, tempting some to engage in fraudulent practices to meet the increasing supply demands. Common adulterants, such as sugar syrups and other sweeteners, are employed to dilute pure honey. The deliberate dilution or contamination of honey compromises its quality, posing serious concerns for both consumers and the honey industry. Traditional methods of detecting adulteration often prove inadequate in addressing the evolving techniques employed by unscrupulous entities. This necessitates the development of advanced methodologies that not only discern subtle variations but also ensure rapid and reliable results.

Hyperspectral imaging emerges as a powerful tool in this context, offering a non-destructive means of acquiring detailed spectral information from honey samples across a broad range of wavelengths. This technology enables the extraction of intricate chemical fingerprints, allowing for the differentiation between authentic and adulterated honey. However, to harness the full potential of hyperspectral data, sophisticated analytical tools such as machine learning are imperative. The current study is research in this direction.

The current study aims to develop an intelligent system that can detect adulteration in honey using machine learning algorithms across three scenarios.

- Identify the class or type of honey.

- Identify the level of adulteration of sugar syrup, in the honey class.
- Identify both the type of honey and the level of adulteration in one-go.

II. LITERATURE REVIEW

A. Existing Quality Assurance techniques

Most honey's plant sources are categorized chemically, however more conventional methods still include honey specialists tasting and smelling the honey. Pollen analysis and assays for specific components that make up different types of honey are among the chemical measures [9]. Numerous techniques have been put up to identify the adulteration of honey with sugar. High-pressure liquid chromatography (HPLC) [17], deuterium nuclear magnetic resonance (NMR) spectroscopy [15, 16], mass spectroscopy of the carbon isotope ratio [13–14] and FTIR spectroscopy [18, 19] can be used to detect adulterated honey.

The detection of honey adulteration with cane sugar using Fourier transform infrared (FTIR) spectroscopy has been the subject of several studies [12]. These studies assessed adulteration in cane sugar concentrations ranging from 0.5 to 25%. In [3], a single variety of honey was utilized to estimate the sugar concentration with an accuracy of 93.75% utilizing statistical techniques and artificial neural networks. When three different varieties of honey were used to classify adulteration, the classification accuracy was less than 80% [12]. These studies demonstrate that it is feasible to anticipate adulteration in honey by combining spectroscopic and machine learning approaches; yet, the capacity to forecast sugar content across a variety of honey varieties has to be enhanced.

By extending spectroscopy and enabling the use of spatial information in addition to spectral information, hyperspectral imaging is a potential method for ensuring the quality of food [1]. Instead of only capturing the spectrum at one spot on the item, spatial information enables the image to highlight certain flaws like bruising on fruit at a specific area [2]. Numerous food quality applications, such as those involving meat, fish, fruit, vegetables, and cereals, have made use of hyperspectral imaging [13].

A hyperspectral imaging technique has been developed to determine the botanical source of honey [4, 7, 8, 11]. With

TABLE I: OVERALL MAKEUP OF THE ADULTERATED HONEY DATA SET FROM EACH BRAND AND BOTANICAL ORIGINS LABEL OF HONEY. TAKEN FROM [21]

Class	Adulteration Concentration					Sum
	0%	5%	10%	25%	50%	
Clover	150	150	300	300	300	1200
MultiFloral	150	150			150	450
ManukaUMF5		150	150	150	150	600
ManukaUMF15		150	150	150	150	600
ManukaUMF20		150	150	150	150	600
ManukaUMF10		150	150	150	125	575
ManukaBlend		150		150	150	450
BorageField	150	150	150	150	150	750
Kamahi	150	150	150	150	150	750
Rewarewa	150	150	150			450
ManukaBlend	150	150	150	150	150	750
Manuka	150	300	300	300	300	1350

90% accuracy, the botanical ancestry of 21 different types of honey was predicted in [11]. These techniques used a class embodiment autoencoder (CEAE) and support vector machines (SVM) to classify the data, which was obtained via a hyperspectral imaging system as detailed in [6]. In [4], 52 samples from five distinct types of honey were identified botanically with a 90% classification accuracy in a small data set.

B. Adulteration Dataset

The dataset [21] comprises 12 unique honey products sourced from seven different brands and characterized by 11 botanical origin labels. Six independent samples were collected for each type of honey, with an equal distribution between Manuka honey, a high-quality honey variant from New Zealand, and other types of New Zealand honey. Throughout the dataset creation process, images of all honey varieties were captured at varying sugar concentrations (5, 10, 25, 50). The detailed composition of the dataset is presented in Table 1. Figure 1 shows the hyperspectral response of six different types of

honey classes with honey adulteration level of 50%. The drift in the spectrum indicates that the effect of adulteration on the spectral response of honey is different and non-linear, where AI would aid in interpreting the trends accordingly.

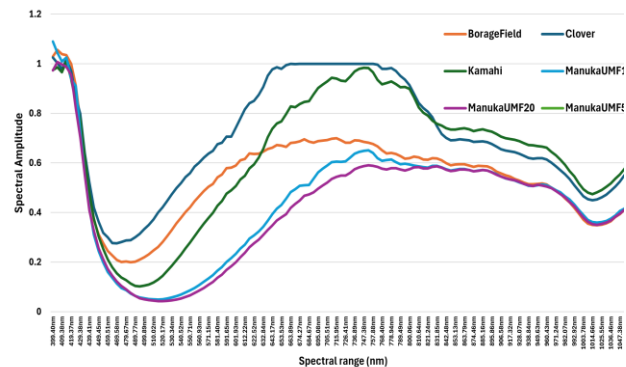


Figure 1: Hyperspectral response of 6 different types of honey classes with an adulteration concentration level of 50%.

III. METHODOLOGY

Figure 2 shows the methodology of the current study in which we have focused on the detection of the following three scenarios. Given a honey sample,

- Task A: To identify the class of honey.
- Task B: To identify the adulteration concentration level.
- Task C: To identify both the class and adulteration concentration level.

The representation of samples is not balanced in all these tasks. In order to reduce bias in the generated machine learning (ML) models, we have then employed balancing the dataset via a popular and common technique called Synthetic Minority Oversampling Technique (SMOTE) [20]. In this manner accurate measurement of ML performance metrics can be recorded accordingly.

- TaskA represents 12 Honey classes (first column in Table 1). After balancing, each class represented 1350 samples.
- For TaskB, honey samples consist of adulteration concentration levels of 0%, 5%, 10%, 25% and 50% in

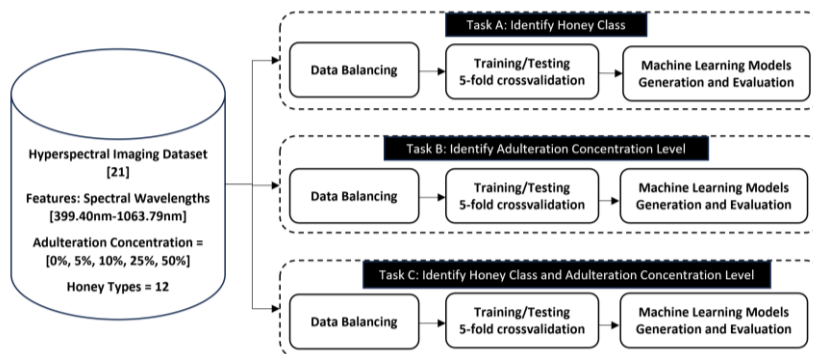


Figure 2: Methodology and execution of various tasks in the current research study.

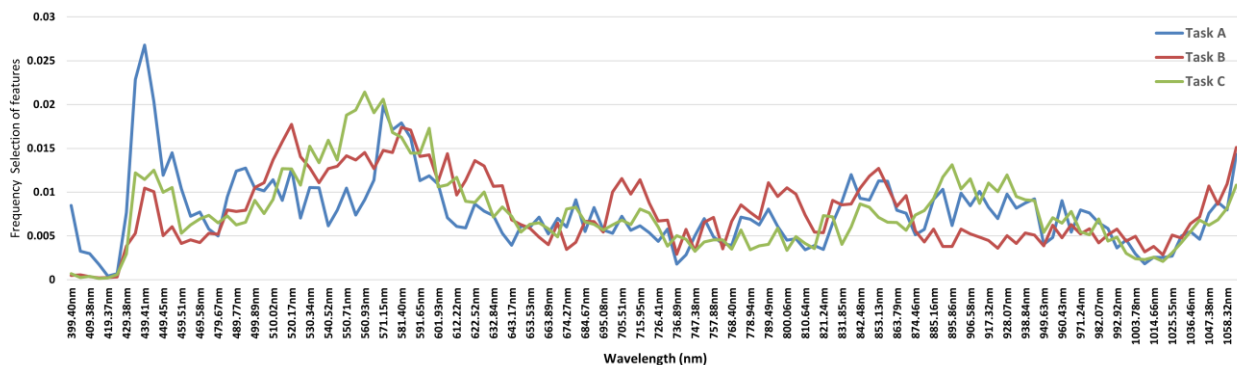


Figure 3: Feature significance graph of the RF models generated for Task A, Task B and Task C respectively.

the dataset. After balancing the set a total of 1950 samples per class were generated.

- Finally, for TaskC, is to identify honey type and adulteration concentration level while using one ML model. After balancing 300 samples per class was generated.

The dataset created in this manner for all the three tasks is then forwarded to the ML model for creation and evaluation. A five-fold cross validation was used to train and test the ML models.

From Figure 3, given a dataset, three different representations of the dataset are generated that are then balanced by using SMOTE. Machine learning algorithms in particular Random Forest (RF), Support Vector Machine (SVM) and Extra Gradient Boosting Trees (XGBoost) were used in generating of the models. The performance of the algorithms across the different tasks are reported in Table II-IV.

IV. RESULTS AND DISCUSSION

Table II-IV shows the performance of the ML models across three different tasks (Task A, Task B and Task C). ML performance metrics namely accuracy, precision, recall and f1-score is reported herewith.

TABLE II. PERFORMANCE OF ML MODELS ON TAKS A, GIVEN A HONEY SAMPLE, IDENTIFY THE TYPE OF HONEY.

ML Model	Accuracy	Precision	Recall	F1 Score
RF	0.99899	0.999043	0.998991	0.999013
SVM	0.853535	0.864275	0.854573	0.851682
XGBoost	0.997306	0.997337	0.997351	0.997334

In Table II, Task A, out of the three ML models, RF model performed best with an accuracy of 99.89% while SVM scored an accuracy of 85.35%. This indicates that the spectral parameters derived from hyperspectral imaging can identify the type of honey investigated.

TABLE III. PERFORMANCE OF ML MODELS ON TAKS B, GIVEN A HONEY SAMPLE , IDENTIFY THE CONCENTRATION ADULTERATION LEVEL

ML Model	Accuracy	Precision	Recall	F1 Score
RF	0.996923	0.99691	0.996953	0.996926
SVM	0.522564	0.51252	0.526156	0.4714
XGBoost	0.995897	0.995878	0.995958	0.995908

While in Table III, Task B, RF performed best with an accuracy of 99.69%. SVM was found to perform with an accuracy of 52.25%. This task was particularly challenging as to identify the amount of adulteration concentration level across different categories of honey. The adulterant used in the dataset was sugar syrup. The main reason towards such performance is due to the presence of natural sugar in honey.

TABLE IV. PERFORMANCE OF ML MODELS ON TAKS C, GIVEN A HONEY SAMPLE, IDENTIFY BOTH CLASS AND YPE OF HONEY.

ML Model	Accuracy	Precision	Recall	F1 Score
RF	0.998936	0.998981	0.998897	0.99893
SVM	0.818794	0.84086	0.821589	0.800515
XGBoost	0.992553	0.992895	0.992604	0.992638

In Table IV, Task C, RF performed with an accuracy of 99.92%. The identification of honey and adulteration level via a single ML model was found to be better than two distinct models. This might be due to the reason of honey types representing unique spectral signature. Hence the change in signature on the spectral properties is reflected differently among different honey types.

Figure 3 shows the feature significance graph obtained from the RF models of the three tasks A, B and C. The feature importance shows that for Task A two identifiable peaks are obtained at a spectral range of 439.41nm and 566.04nm, for Task B around 520.17nm and 586nm while for Task C spectral range of 560nm and 596.79nm were found to be the important spectral features in these tasks. It's interesting to

note that from Task A to Task C a shift in peak features is observed.

Table V, shows the comparison of the ML models with the ones reported in literature [4] on the dataset. As the dataset was not balanced in the reported literature hence f1-score is low. Due to balancing of dataset, our study was able to improve on these metrics.

TABLE V. COMPARISON OF ML MODELS WITH OTHERS REPORTED IN LITERATURE

Task	Accuracy	Precision	Recall	F1 Score
Adulteration Concentration	0.996923	0.99691	0.996953	0.996926
Adulteration Concentration [21]	0.951	x	x	0.940

V. CONCLUSION

The adulteration of honey has become a widespread practice aimed at increasing economic benefits, however it has been shown to have detrimental effects on an individual's health. The current study explores the potential of machine learning algorithms in accurate identification of honey adulteration on a recently public available dataset. SMOTE algorithm was used to balance the dataset. Random Forest, Support Vector Machine and XGBoost algorithms were used to generate models whereby RF was found to perform better than the other two algorithms in identification of quality of honey. In comparison to the reported study (95%), our study produced an accuracy of 99.69% on the same dataset. This indicates the potential of ML algorithms in the accurate identification and quantification of honey adulteration.

ACKNOWLEDGMENTS

We acknowledge the Department of ORIC at CECOS University of IT & Emerging Sciences for funding of resources.

REFERENCES

- [1] El Masry G, Sun DW (2010) Principles of hyperspectral imaging technology. In: *Hyperspectral imaging for food quality analysis and control*. Elsevier, p 3–43. <https://doi.org/10.1016/b978-0-12-374753-2.10001-2>
- [2] Gowen A, O'Donnell C, Cullen P et al (2007) Hyperspectral imaging—an emerging process analytical tool for food quality and safety control. *Trend Food Sci Technol* 18(12):590–598. <https://doi.org/10.1016/j.tifs.2007.06.001>
- [3] Irudayaraj J, Xu R, Tewari J (2003) Rapid determination of invert cane sugar adulteration in honey using ftir spectroscopy and multivariate analysis. *J Food Sci* 68(6):2040–2045. <https://doi.org/10.1111/j.1365-2621.2003.tb07015.x>
- [4] Minai S, Shafee S, Polder G et al (2017) Vis/nir imaging application for honey foral origin determination. *Infrar Phys Technol* 86:218–225. <https://doi.org/10.1016/j.infrared.2017.09.001>
- [5] Noviyanto A (2018) Honey botanical origin classification using hyperspectral imaging and machine learning. PhD thesis. The University of Auckland. <https://doi.org/10.1016/j.jfoodeng.2019.109684>
- [6] Noviyanto A, Abdulla WH (2017) Honey dataset standard using hyperspectral imaging for machine learning problems. In: *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE, pp 473–477. <https://doi.org/10.23919/eusipco.2017.8081252>
- [7] Noviyanto A, Abdulla WH (2020) Honey botanical origin classification using hyperspectral imaging and machine learning. *J Food Eng* 265(109):684. <https://doi.org/10.1016/j.jfoodeng.2019.109684>
- [8] Noviyanto A, Abdulla WH (2021) Signifying the information carrying bands of hyperspectral imaging for honey botanical origin classification. *J Food Eng* 292(110):281. <https://doi.org/10.1016/j.jfoodeng.2020.110281>
- [9] Noviyanto A, Abdullah W, Yu W, et al (2015) Research trends in optical spectrum for honey analysis. In: *Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2015 Asia-Pacific, IEEE, pp 416–425. <https://doi.org/10.1109/apsipa.2015.7415305>
- [10] Phillips T, Abdulla W (2020) Generalisation techniques using a variational ceae for classifying manuka honey quality. In: *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, pp 1631–1640
- [11] Phillips T, Abdulla W (2019) Class embodiment autoencoder (CEAE) for classifying the botanical origins of honey. In: *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE. <https://doi.org/10.1109/ivcnz48456.2019.8961004>
- [12] Sivakesava S, Irudayaraj J (2001) Prediction of inverted cane sugar adulteration of honey by fourier transform infrared spectroscopy. *J Food Sci* 66(7):972–978. <https://doi.org/10.1111/j.1365-2621.2001.tb08221.x>
- [13] Brookes, S.T. Barrie, A. and Davies, J.E. (1991) A rapid 13C/12C test for determination of corn syrups in honey. *J.Ass. Off. Anal. Chem.* 74:627-629.
- [14] White J.W., Winters, K, Martin, P. and Rossmann A. (1998). Stable carbon isotope ratio analysis of honey: Validation of internal standard procedure for worldwide application. *J. Ass. Off. Anal. Chem.* 81: 610-619.
- [15] Linder, P., Bermann, E. and Gamarnik, B (1996). Characterization of Citrus honey by deuterium NMR. *J. Agr. Food Chem.* 44: 139-140
- [16] Giraudon, S., Danzart, M. and Merle, M.H. (2000) Deuterium nuclear magnetic resonance spectroscopy and stable carbon isotope ratio analysis/mass spectrometry of certain monofloral honey. *J. Ass. Off. anal. Chem.* 83: 1402- 1409.
- [17] Cordella C , Moussa I , Martel A.C , Sbirrazzuoli N , and Lizzani-Cuvelier. L, (2002) Recent Development in Food characterization and adulteration detection:Techniqueoriented perspectives. *J. Agric. Food Chem.* 50 (7): 1751– 1764.
- [18] Gok S., Severcan M, Goormaghtigh E, Kandemir I, Severcan F. (2015). Differentiation of Anatolian honey samples from different botanical origins by ATR-FTIR spectroscopy using multivariate analysis. *Food Chemistry.* 170: 234-240.
- [19] Sivakesava, S. and Irudayaraj, J. (2001). Detection of inverted beet sugar adulteration of honey by FTIR spectroscopy. *J. Sci. Food and Agric.* 81 : 683-690.
- [20] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.
- [21] Phillips, T., Abdulla, W. A new honey adulteration detection approach using hyperspectral imaging and machine learning. *Eur Food Res Technol* 249, 259–272 (2023). <https://doi.org/10.1007/s00217-022-04113-9>

Optimized Coverage and Capacity Planning of WiFi Network based on Radio Frequency Modeling & Propagation Simulation

Junaid Ali Shah Syed
Telecommunication Engineering
University of Engineering & Technology
Mardan
Mardan, Pakistan
s.junaid52462@gmail.com

Naveed Mufti
Director Technical
SNM Solutions Pvt. Ltd.
Rawalpindi, Pakistan
mufti.naveed@gmail.com

Toufeeq Ahmad
Telecommunication Engineering
University of Engineering & Technology
Mardan
Mardan, Pakistan
drtoufeeq@uetmardan.edu.pk

Abstract—Investigation for optimized coverage and capacity planning of WiFi network is carried out in the testbed for the purpose of optimization in terms of Received Signal Strength Indicator (RSSI), Signal to Noise & Interference Ratio (SNIR), Interference + Noise (I+N), downlink/uplink data rate and user capacity. The plan is carried out by conducting a site prediction survey through Altair's WinProp Software which is a Radio Frequency (RF) modeling and signal propagation simulation software, using the configuration of actual Wireless Local Area Network-Access Points (WLAN-APs). First the map of the testbed with all respective material properties is drawn in WinProp's Wall Manager (WallMan) Tool as a 3-Dimensional (3D) model. Then that 3D model is implemented in WinProp's Propagation Manager (ProMan) Tool where APs are deployed and wave propagation analysis as well as capacity planning is done. Results are analyzed for optimal signal strength, data rate and user handling capacity. Then results are validated by a smartphone embedded software known as Cellular-Z. The average optimization increase in coverage, downlink & uplink data rates is 3.95 dB, 2.53 Mbps & 3.42 Mbps respectively.

Keywords—capacity; coverage; heatmap; optimization; wifi

I. INTRODUCTION

WiFi means wireless fidelity. It is the technology of providing wireless signals for internet connectivity to the users within an area of deployment. WiFi works on IEEE standards of 802.11b/g/n/a/ac, mainly within 2.4GHz and 5.6GHz Industrial Scientific Medical (ISM) bands. The available bandwidth within each band is further divided into sub-bands/channels which are responsible for user handling capacity. The 2.4GHz band provides low data rate but large coverage as compared to 5.6GHz band and vice versa.

The WiFi is provided to the internet users through Wireless Local Area Network Access Point (WLAN-AP). The antennas of this AP are nearly omni-directional with some gain, radiating radio signals in maximum directions towards the users. The Received Signal Strength (RSS) of the WiFi signals decrease as the user moves away from the WLAN-AP. This variable signal strength as the users move towards and away from the WLAN-AP results in variable data rate [1].

To provide connectivity of internet to maximum number of users with an acceptable data rate, it is important to know the optimum location of WLAN-AP for providing coverage at specific RSS level to the internet users, number

of users that can be handled by a WLAN-AP and sufficient data rate that is provided to each user at specific time.

Conventional ways of performing this task are practical deployment and modification of the location of WLAN-APs and measuring the RSS at different portions of the area of interest without prior simulation surveys. This approach is tedious as well as costly, having low accuracy and take more time for completion. To engage the above issues, software based simulation surveys are performed before actual deployment of WiFi network.

In the background study, the same software is used for a similar type of survey but the study did not use the actual WLAN-AP configuration, assumed omnidirectional antennas and same antenna pattern (single frequency) for all WLAN-APs throughout the simulation process which lacks the opportunity of handling the co-channel interference and did not determine the number of users that could be supported with a specific minimum guaranteed data rate i.e. that survey is limited to propagation with no capacity planning [2].

II. LITERATURE REVIEW

S. ZVANOVEC, P. PECHAC and M. KLEPAL experimented to analyze merits and demerits of two distinct methods for WiFi Network Deployment Survey. One was the practical survey in which WiFi transmitters were deployed in a testbed and Received Signal Strength Indicator (RSSI) measurements were taken on multiple points and the second method was simulation of signal propagation model in software tool. The experimental data was simulated in MATLAB. Results from both methods were analyzed and software simulation method was preferred for WiFi Network Deployment [3].

T. Honda, M. Ikeda and L. Barolli conducted experiments to optimize the coverage by correct placement of WiFi APs through site survey and network simulations for the solution of connectivity problem. Results indicated that the received power from APs was not uniform [4].

T. Witono and Y. Dicky did practical site measurements of RSS for the optimization of 12 WiFi transmitters in an overlapping WiFi environment. The key controllers of a single WiFi transmitter were the direction of transmitting antennas, the combination of channels and the transmit power, all of which were adjusted for the optimization of WiFi deployment [5].

U. Mir, O. U. Sabir, H. Ullah and A. U. Khan experimented to achieve coverage optimization through accurate placement of APs based on RSSI measurements

in the testbed. Site simulation survey was conducted through Tamograph and real time measurements were taken through InSSIDer software and Air Magnet hardware (validation). The results from simulation and measurements were analyzed for the optimization along with the voltage variation effects on the RSSI measurements [6].

J. Tan, X. Fan, S. Wang and Y. Ren collected RSS measurements from the inertial sensors of a smart phone by walking in the testbed of WiFi environment for the purpose of accurate Radio Map (WiFi Fingerprint) construction. The RSS data was processed by the pedestrian dead-reckoning algorithm for the production of raw trajectories. Those trajectories were refined by the assembling of constrains collected at the landmarks, by the use of Factor Graph Optimization (FGO). Then k-Nearest Neighbour (kNN) algorithm was applied for the validation and localization performance testing of the Radio Map. The Radio Map was practically implemented in a shopping mall and mean error of 1.10m and maximum error of 2.25m was recorded for WiFi transmitter locations which is an acceptable error for Radio Map [7].

A. Srivastava, R. Vatti, V. Deshpande, J. Patil and O. Nikte did practical measurement of RSSI for finding dead zones with less or no coverage of WiFi signals in the area through Netspot Tool. Further, optimization techniques of Particle Swarm Optimization {PSO (responsible for optimal WiFi transmitter placement)} and Repeater deployment were implemented to solve the problem of coverage in the dead zones [8].

Y. Tian, B. Huang, B. Jia and L. Zhao, developed an algorithm for the accurate placement of WiFi access points and Bluetooth beacons in WiFi / Bluetooth hybrid environment. The "heuristic differential evolution algorithm" (HDEA) which is based on "Cramer-Rao lower bound (CRLB)". The CRLB is considered as a standard for localization and coverage of WiFi/Bluetooth signals. Further, Motley-Keenan model is assembled in the algorithm instead of ideal Log Distance Path Loss (LDPL) model for the analysis of the effects caused by obstacles in the indoor environment. Based on these contents, the algorithm is deployed in a software application which is used with Geo-Tools for the localization of WiFi access points and Bluetooth beacons. Extensive simulations and experiments in the field were conducted to validate the efficiency of the algorithm [9].

N. A. M. Maung and W. Zaw, conducted experiments to compare and analyze the performance of two techniques of WiFi indoor positioning in 2.4GHz and 5GHz frequency bands. The implemented techniques were path loss model and RSS Fingerprint. Results show that the RSS based indoor WiFi positioning performed better accuracy than the other technique because the path loss model takes direct reading of RSS value (highly variable due to multipath and interference) and estimates the location which leads to positioning error [10].

M. R. Akram, A. H. Al-Nakkash, O. N. M. Salim and A. A. S. AlAbdullah, developed a multi-objective algorithm for the optimization of WiFi coverage, location and number of access points by the use of MATLAB software. The algorithm works on Binary Particle Swarm Optimization (BPSO) technique which takes predefined

RSS values to estimate the optimization of the aforesaid objectives. The deployment of algorithm resulted in 64.6% coverage and 7dBm in average received power optimization [11].

O. S. Naif and I. J. Mohammed experimented Binary Particle Swarm Optimization (BPSO) algorithm used with Wireless Insite (WI) simulation software, to optimize the coverage and interference parameters of a multi-floor WiFi AP deployment. The WI takes RSS values, signal thresholds and current AP deployment locations to process the optimization in conjunction with BPSO. Results depict that the proposed work outperforms the present WiFi deployment in RSS (-11.5dBm), path loss (11.5dBm), interference (7.87%) and coverage/ optimal AP placement of 39.23% [12].

A. S. Haron, Z. Mansor, I. Ahmad and S. M. M. Maharum did simulation based survey to optimize the location of presently deployed WiFi transmitters in 2.4GHz and 5GHz bands in terms of signal strength to overcome the problem of connectivity in the Communication Technology Laboratories Area at University of Kuala Lumpur British Malaysian Institute. HyperWorks' WinProp simulation software was used. First the layout of the testbed was modeled in the WinProp's WallMan Tool, having similar properties/sizes of materials from the map. Then omni-directional antenna patterns of 2.4GHz and 5GHz were modeled in WinProp's AMan Tool. After that both models from WallMan and AMan were implemented in WinProp's ProMan Tool where the signal propagation analysis was done for each antenna with the exact deployment location as was in real. The simulation was carried out in 2.4GHz and 5GHz bands separately for all antennas with modification in the location of WLAN-APs and results were compared. After that physical validation of the acquired results was done using InSSIDer by measuring the RSS and optimum location for the WiFi transmitters was determined [2].

S. Baua and S. Karuppuswami, presented Machine Learning (ML) technique of Modified Extensible Lattice Sequence (MELS) which is a regression based supervised learning algorithm, is used with Global Response Search Method (GRSM) optimization routine to optimize the coverage by correct placement of transmitters and number of WiFi APs for an office area. A dual slot antenna is designed operating in 5.2GHz (S46 / S54 bands) and having 6dBi of gain, to represent a single WiFi AP for the processing of radio optimization. The target of optimization is to reduce the number of APs, place them in accurate locations for wider coverage and to provide at least 13Mbps of data rate at 15000 square foot area at the office location [13].

I. Bridova and M. Moravcik did predictive and passive surveys to overcome the problem of WiFi connectivity at the Department of Information Networks, University of Zilina. The methodology includes model construction from a Map and creation of predictive heat-maps in Ekahau simulation software in terms of relocation of APs with respect to hotspots and unnecessary areas. During prediction, no objects (furniture, electronic devices) were considered. Then physical readings were collected in form of passive survey and results were analyzed for optimal coverage, Signal to Noise Ratio and Throughput [14].

All studies have focused on coverage optimization with no capacity planning. This research work performed propagation (coverage optimization) as well as network simulations (capacity planning) which provided radiation patterns for all antennas at once in the testbed. As a result, the RSSI, Signal to Noise & Interference Ratio (SNIR) and Interference + Noise (I+N) in the testbed are analyzed as a combined result of all antennas at once (received power of the network) as well as data rate (downlink/uplink) and user handling capacity calculations are done.

III. METHODOLOGY

It consists of three parts i.e. Physical data collection, implementation of collected data as 3D model generation & WiFi network simulations and optimization analysis & validation of results.

The First part is carried out by acquiring physical data of the testbed including 2D map with construction objects' sizes and materials; technical datasheet of the access point with supporting WiFi technologies, antenna patterns with respective gain, transmit powers, receiver sensitivities, polarizations & operational bands; RSSI values and downlink/uplink data rates as real time measurements of the present case acquired through Cellular-Z.

In second part, the collected data is practically implemented in Altair's WinProp software in which WallMan is a 3D modeling construction Tool and ProMan is a signal propagation simulation Tool. Map with all details of floor plan, construction sizes and material properties is designed for the area where the WiFi network is optimized as a 3D model in the WallMan Tool.

Then, [1] Access Point's WiFi technologies, antenna patterns with respective gain, transmit powers, receiver sensitivities, polarizations & operational bands; RSSI values and downlink/uplink data rates are created and deployed within the 3D model created in previous step through WinProp's ProMan Tool.

Then, radio coverage and network capacity planning in the modeled environment is simulated for all WiFi APs using WinProp's ProMan for present case. The simulation process is repeated for 73 times in pursuit of coverage and capacity optimization, comparing each simulated case with present case. Results are acquired for 2.4GHz band.

In the third part, results are analyzed for optimal signal strength, data rate and user handling capacity. The best case is selected and applied in the testbed. Then results are validated by a smartphone embedded software(Cellular-Z).

IV. RESULTS

The results in Table I & Table II show the difference in RSSI, Data Rate, SNIR, I+N and Modulation & Coding Schemes (MCSs) in the testbed.

TABLE I. BEFORE OPTIMIZATION OF APS

Received Power			
Mean (dBm)	Median (dBm)	Standard Deviation (dBm)	Maximum (dBm)
-73.13	-73.58	12.07	-45.02

Downlink / Uplink Data Rate (Mbps)	Downlink: Signal to Noise & Interference Ratio (dB)	Downlink: Interference + Noise (dBm)	Number of Modulation & Coding Schemes Operated
19.48 / 4.10	10.17	-53	3

TABLE II. AFTER OPTIMIZATION OF APS

Received Power			
Mean (dBm)	Median (dBm)	Standard Deviation (dBm)	Maximum (dBm)
-71.71	-72.62	11.12	-43.25

Downlink / Uplink Data Rate (Mbps)	Downlink: Signal to Noise & Interference Ratio (dB)	Downlink: Interference + Noise (dBm)	Number of Modulation & Coding Schemes Operated
64.94 / 13.67	63.59	-65	8

In Table III, 20 location points are taken for comparison between real time measurements and simulation predictions before & after optimization. The simulation data can be visualized in Figure 1 & Figure 2 respectively. The average optimization difference between non-optimized real measurements & optimized real measurements is 3.95 dB, average optimization difference between non-optimized simulation predictions & optimized simulation predictions is 0.75 dB and average optimization difference between optimized simulation predictions & optimized real measurements is 1.47 dB.

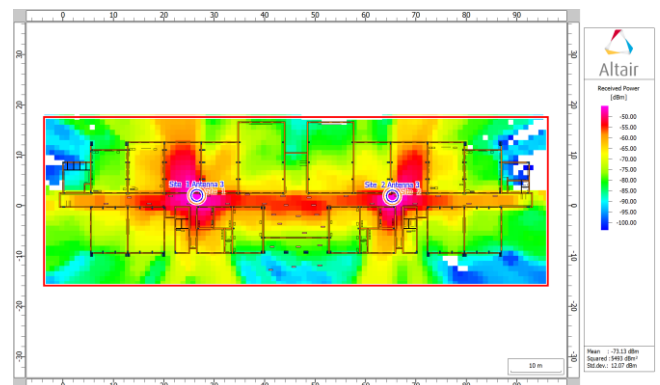


Figure 1: Simulation Values of RSSI Before Optimization

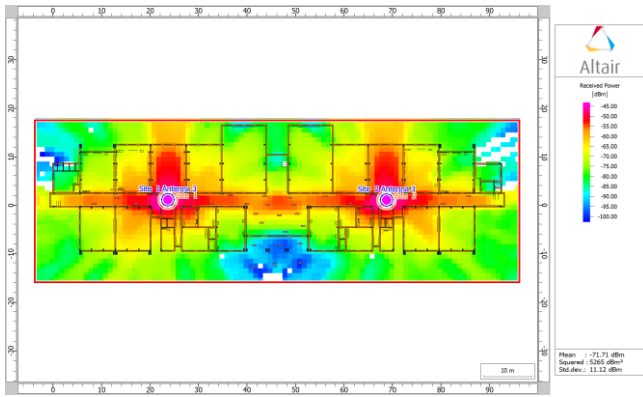


Figure 2: Simulation Values of RSSI After Optimization

TABLE III. COVERAGE OPTIMIZATION COMPARISON

Locations (x, y)	Real Time RSSI (dBm)		Simulation Predictions RSSI (dBm)	
	Non-Optimized	Optimized	Non-Optimized	Optimized
19.5, -8.5	-71	-66	-74.48	-67.26
27.5, -8.5	-71	-68	-65.58	-72.22
31.5, -8.5	-74	-69	-69.93	-74.26
27.5, -5.5	-71	-68	-58.33	-67.01
9.5, -4.5	-76	-72	-71.23	-71.33
23.5, -1.5	-54	-51	-53.89	-50.93
27.5, -1.5	-53	-51	-49.91	-53.88
18.5, 0.5	-58	-53	-53.98	-52.09
23.5, 0.5	-42	-41	-47.77	-44.13
18.5, 1.5	-58	-53	-54.66	-52.09
23.5, 1.5	-42	-41	-47.23	-44.13
0.5, 3.5	-85	-80	-85.3	-78.98
4.5, 3.5	-84	-77	-80.15	-70.94
19.5, 3.5	-63	-56	-59.04	-55.91
26.5, 3.5	-51	-47	-49.84	-53.11
4.5, 6.5	-80	-79	-84.94	-80.64
39.5, 9.5	-78	-72	-77.37	-71.47
47.5, 9.5	-82	-77	-80.14	-82.73
35.5, 15.5	-77	-75	-73.79	-80.62
42.5, 15.5	-82	-77	-79.87	-78.65

In Table IV, 19 location points are taken for comparison between real time measurements and simulation predictions before & after optimization. The simulation data can be visualized in Figure 3 & Figure 4 respectively. The average optimization difference between non-optimized real measurements & optimized real measurements is 2.53 Mbps, average optimization

difference between non-optimized simulation predictions & optimized simulation predictions is 0.75 Mbps and average optimization difference between optimized simulation prediction & optimized real measurements is 4.17 Mbps.



Figure 3: Downlink Data Rate Before Optimization

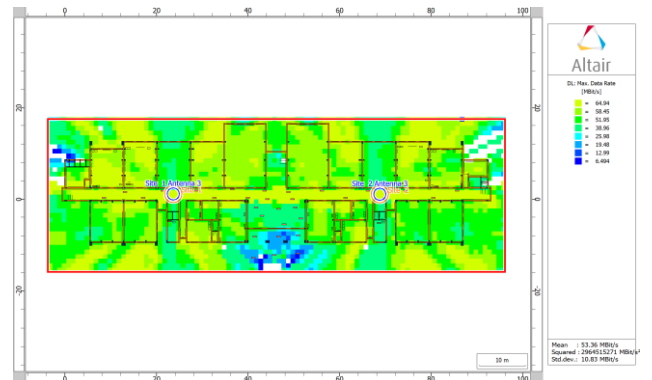


Figure 4: Downlink Data Rate After Optimization

TABLE IV. CAPACITY OPTIMIZATION PER USER DATA RATE (DOWNLINK)

Locations (x, y)	Real Time Data Rate (Mbps)		Simulation Predictions Data Rate (Mbps)	
	Non-Optimized	Optimized	Non-Optimized	Optimized
2.7, 4.5	0.5	1	0.08	0.61
16.2, 7.5	4.3	5.6	0.08	0.93
24, 7.4	4.5	11.5	0.08	0.61
31, 7.6	2.6	2.6	0.08	0.93
39, 9.3	1.2	1.2	0.08	0.93
46, 6	1.4	4.9	0.08	0.81
9, 1	0.8	10.3	0.08	1.03
16.5, 1	1	5.8	0.08	1.03
23.6, 1	2.6	10.3	0.08	1.03
29.7, 1	1.3	6.3	0.08	1.03

36.7, 1	1.3	5.6	0.08	1.03
42, -3	1.4	1.2	0.08	0.61
45.8, -3	1.8	1	0.08	0.61
50.6, -3	2.4	1.6	0.08	0.61
9, -4.9	4.3	1.2	0.08	0.81
16.5, -4.9	11.8	4.5	0.08	0.81
28, -2.5	1.5	11.7	0.08	0.81
29, -7	1	5.3	0.08	0.93
36.5, -5	1.2	3.4	0.08	0.61

In Table V, 19 location points are taken for comparison between real time measurements and simulation predictions before & after optimization. The simulation data can be visualized in Figure 5 & Figure 6 respectively. The average optimization difference between non-optimized real & optimized real measurements is 3.42 Mbps, average optimization difference between non-optimized simulation & optimized simulation predictions is 0.16 Mbps and average optimization difference between optimized simulation predictions & optimized real measurements is 7.55 Mbps.

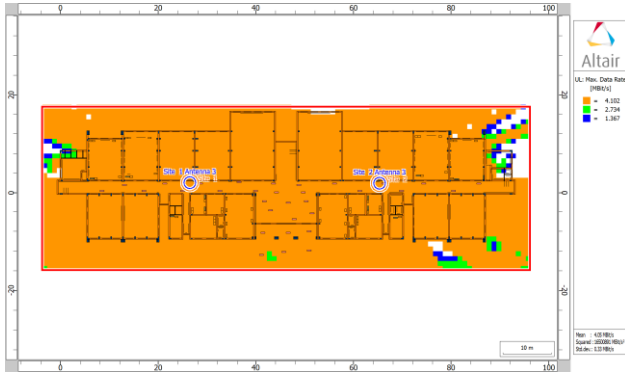


Figure 5: Uplink Data Rate Before Optimization

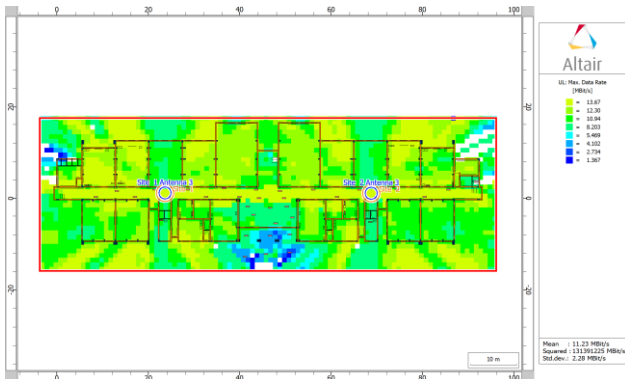


Figure 6: Uplink Data Rate After Optimization

TABLE V. CAPACITY OPTIMIZATION PER USER DATA RATE (UPLINK)

Locations (x, y)	Real Time Data Rate (Mbps)		Simulation Predictions Data Rate (Mbps)	
	Non-Optimized	Optimized	Non-Optimized	Optimized
2.7, 4.5	0.5	0.2	0.02	0.13
16.2, 7.5	5	4.7	0.02	0.2
24, 7.4	7	12.9	0.02	0.13
31, 7.6	5.8	10.9	0.02	0.2
39, 9.3	1.1	9	0.02	0.2
46, 6	3	3	0.02	0.17
9, 1	6.4	10.3	0.02	0.22
16.5, 1	8	2.2	0.02	0.22
23.6, 1	9.4	11.8	0.02	0.22
29.7, 1	5.8	11.2	0.02	0.22
36.7, 1	4.5	8.4	0.02	0.22
42, -3	2.2	9.9	0.02	0.13
45.8, -3	2.2	6.1	0.02	0.13
50.6, -3	4.8	3.3	0.02	0.13
9, -4.9	1	6.8	0.02	0.17
16.5, -4.9	4.2	5.8	0.02	0.17
28, -2.5	4.4	11	0.02	0.17
29, -7	3.6	10.6	0.02	0.2
36.5, -5	2.9	8.7	0.02	0.13

The maximum logical user handling capacity of each AP is 254 users because they use Class C IPv4 addressing whose range is $2^8 = 256$ addresses in which the first address is allocated to the AP itself and the last one is subnet mask, so $256 - 2 = 254$. This setup of IP addressing is assigned by the network administration and the research for optimization is carried out within these bounds. If Class B address is assigned, then capacity planning should be done to that setup accordingly. Practically, the user handling capacity of an AP is further limited by the physical resources but as the number of users under specific MCS decrease, per user data rate increase and vice versa.

V. CONCLUSION

This research has achieved coverage and capacity optimization in terms of RSSI, data rate, SNIR, I+N & user capacity in the already deployed WiFi network. The number of APs in the testbed is decreased from 7 to 2. The optimization is performed in WinProp's ProMan through simulations based on location and antenna polarization/

physical orientation variation of the APs in the testbed. Then the most optimum case with respect to coverage and capacity is selected within the dataset of all possible cases (73) and applied in real time environment. Then real time measurements are taken with respect to coverage and capacity (validation). In calculation of per user data rate (simulation), the total number of users is divided by the total number of MCS operated in the testbed. In this case, every MCS serves equal number of users. So users within same MCS get the same data rate in downlink and uplink. But in reality, more users come under the same MCS, less data rate each user will get and vice versa.

The future work consists of Monte Carlo Simulation (location dependent traffic analysis), Prediction analysis (Delay Spread, Angular Spread & Angular Means), Electromagnetic Compatibility (EMC) analysis, Consideration of Mobile Station properties (propagation & channel properties), MIMO analysis in uplink & downlink, designing accurate pattern of AP antennas in AMan, implementation and analysis of IEEE 802.11 a/ac technologies for higher capacity and RSSI calculations for worst case scenario.

REFERENCES

- [1] "Cisco WAP4410N Wireless-N Access Point," 2011 Cisco and/or its affiliates.
- [2] A. S. Haron, Z. Mansor, I. Ahmad and S. M. M. Maharum, "The Performance of 2.4GHz and 5GHz Wi-Fi Router Placement for Signal Strength Optimization Using Altair WinProp," in *2021 IEEE 7th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, 2021.
- [3] S. ZVANOVEC, P. PECHAC and M. KLEPAL, "Wireless LAN Networks Design: Site Survey or Propagation Modeling?," *RADIOENGINEERING*, vol. 12, 2003.
- [4] T. Honda, M. Ikeda and L. Barolli, "Performance Analysis of User Connectivity by Optimizing Placement of Wireless Access Points," in *2013 16th International Conference on Network-Based Information Systems*, 2013.
- [5] T. Witono and Y. Dicky, "Optimization of WLAN deployment on classrooms environment using site survey," in *2017 11th International Conference on Information & Communication Technology and Systems (ICTS)*, 2017.
- [6] U. Mir, O. U. Sabir, H. Ullah and A. U. Khan, "WLAN configuration and location optimization on RSSI basis," University of Engineering & Technology, Mardan, Mardan, 2017.
- [7] J. Tan, X. Fan, S. Wang and Y. Ren, "Optimization-Based Wi-Fi Radio Map Construction for Indoor Positioning Using Only Smart Phones," *Sensors*, 2018.
- [8] A. Srivastava, R. Vatti, V. Deshpande, J. Patil and O. Nikte, "Coverage Improvement of IEEE 802.11n Based Campus Wide Wireless LANs," in *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, 2018.
- [9] Y. Tian, B. Huang, B. Jia and L. Zhao, "Optimizing AP and Beacon Placement in WiFi and BLE hybrid localization," *Journal of Network and Computer Applications*, 2020.
- [10] N. A. M. Maung and W. Zaw, "Comparative Study of RSS-based Indoor Positioning Techniques on Two Different Wi-Fi Frequency Bands," in *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2020.
- [11] M. R. Akram, A. H. Al-Nakkash, O. N. M. Salim and A. A. S. AlAbdullah, "Proposed APs Distribution Optimization Algorithm: Indoor Coverage Solution," *Journal of Physics*, 2021.
- [12] O. S. Naif and I. J. Mohammed, "Wireless Optimization Algorithm for Multi-floor AP deployment using binary particle swarm optimization (BPSO)," *Journal of Physics*, 2021.
- [13] S. Bua and S. Karuppuswami, "WiFi coverage planning and router position optimization using machine learning," in *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, 2022.
- [14] I. Bridova and M. Moravcik, "A System Approach in a WiFi Network Design," in *PROCEEDING OF THE 33RD CONFERENCE OF FRUCT ASSOCIATION*, 2023.

Breast Masses Detection Using YOLOv8

Abdul Moiz

Hikmat Ullah

Hannia Naseem

Umar Sadique

Muhammad Abeer Irfan

Amaad Khalil

Dept. of CSE, UET Peshawar, Pakistan Dept. of CSE, UET Peshawar, Pakistan Dept. of CSE, UET Peshawar, Pakistan Dept. of CSE, UET Peshawar, Pakistan Dept. of CSE, UET Peshawar, Pakistan Dept. of CSE, UET Peshawar, Pakistan

Email: moizmansoor68@gmail.com Email: hikmatcse1919@gmail.com Email: hanninaseem836@gmail.com Email: umar.sadique@uetpeshawar.edu.pk Email: abeer.irfan@uetpeshawar.edu.pk Email: amaadkhalil@uetpeshawar.edu.pk

Abstract—Breast cancer stands as a formidable global health challenge, necessitating swift and precise diagnostic measures to combat its devastating impact. In this study, we delve into the efficacy of YOLOv8, a cutting-edge artificial intelligence model, for the precise detection and localizing of breast masses in digital mammography images. YOLOv8’s inherent capability to simultaneously detect and localize masses showcases accurate pinpointing of the exact locations of abnormalities within mammographic scans. Our comprehensive evaluation reveals compelling performance metrics, including an F1 score of 0.91 and a mean Average Precision (mAP) of 0.942. These results depict the robustness of the YOLOv8 in mass detection but also show better results than the conventional clinical methods, offering higher accuracy and efficiency in the diagnostic process.

This study explains the transformative potential of YOLOv8 in revolutionizing breast cancer detection paradigms, presenting a promising pathway toward enhancing early detection rates and ultimately improving patient outcomes.

Keywords: YOLOv8, Mass Detection and Localization, Digital Mammography

I. INTRODUCTION

Breast cancer poses a significant global health challenge, underscoring the urgent need for advancements in diagnostic techniques to ensure early identification and improved patient outcomes [1]. While traditional methods like mammography have been pivotal in breast cancer screening, their limitations in sensitivity and specificity have prompted increased interest in leveraging technological breakthroughs, particularly the application of YOLO (You Only Look Once) and its latest iteration, YOLO v8, in medical imaging for enhanced breast cancer detection.

The intricate development of malignant breast masses stems from aberrant cell division within human tissues, leading to the emergence of benign and malignant masses. Benign masses, non-cancerous in nature, exhibit localized growth without aggressive tendencies. Conversely, malignant breast masses, driven by cancerous cells, possess an uncontrolled propensity to multiply and potentially spread to different body parts and adjacent tissues [2]. YOLO v8, as a cutting-edge object detection system, has emerged as a transformative force in medical imaging, heralding improved capabilities for disease detection [3].

In the realm of deep learning for breast cancer detection, the primary focus is on utilizing a diverse dataset to train YOLO v8 with representations from various mammography views. This strategic approach aims to augment the model’s practical performance by fostering a comprehensive understanding of breast cancer lesions. Overcoming challenges associated

with traditional diagnostic techniques is crucial for achieving greater accuracy and efficacy [4]. The deliberate inclusion of mediolateral oblique and craniocaudal views is deemed essential, enhancing the model’s ability to detect subtle patterns indicative of malignant growth. This comprehensive strategy elevates sensitivity and equips YOLO v8 to navigate complexities in identifying and classifying breast cancer [5].

This study is different from conventional approaches by taking leverage of a carefully curated breast mass dataset obtained from Roboflow to accurately annotate them which leads to achieving better results; further, we also conducted a validation process to ensure the dataset’s quality and accuracy. Furthermore, collaborating with a radiologist for the result validation strengthens the clinical relevance of our findings. By employing YOLOv8 on this dataset and incorporating expert validation, our research offers valuable insights into the efficacy of YOLOv8 for breast mass detection. This investigation paves the way for further exploration of deep learning in breast cancer screening, potentially leading to more accurate diagnoses and improved patient care.

II. LITERATURE REVIEW

Each year, the American Cancer Society estimates the numbers of new cancer cases and deaths in the United States and compiles the most recent data on population-based cancer occurrence and outcomes using incidence data collected by central cancer registries (through 2020) and mortality data collected by the National Center for Health Statistics (through 2021). In 2024, 2,001,140 new cancer cases and 611,720 cancer deaths are projected to occur in the United States [6]. This highlights the critical need for breakthroughs in automated detection technology (Amrane et al., 2018). [7][8]. The importance of mammography images in the diagnosis of breast cancer has led to a thorough investigation of developments in detection and classification.

Breast cancer impacts more than one in ten women globally, but it is particularly prominent—across all racial and ethnic groups—in the United States. The need for focused diagnostic efforts is highlighted by differences in incidence rates amongst ethnic groups [9][10]. Breast lesions are complex, three-dimensional anomalies that reflect a variety of radiologically defined illnesses. The distinction between benign and malignant lesions must be made early to improve the prognosis of patients with this cancer, which is the most common in women and the second largest cause of cancer-related fatalities [11][12].

The use of deep learning techniques in computer vision, segmentation, detection, and image identification has increased dramatically in recent years, overcoming the drawbacks of conventional computer-aided diagnosis (CAD) methods [13-15]. Despite these developments, problems in manually identifying lesions and controlling memory complexity during training still exist.

Surprisingly, when compared to shallower models, deep learning models like AlexNet, ResNet, VGG16, Inception, GoogleNet, and DenseNet have shown improved classification performance. The classification accuracies attained by VGG16, ResNet50, and Inception-V3 were 95%, 92.5%, and 95.5%, respectively. Although these deep learning techniques perform better than shallow models, problems with memory complexity during training and manual detection still exist.

Building upon the landscape of breast cancer detection, recent advancements by Mahoro and Akhloufi [16] showcase the potential of YOLOv7 and YOLOv8 in breast mass detection. The utilization of the VinDr-Mammo dataset, coupled with innovative image enhancement techniques, positions YOLOv8 as a superior model, outperforming its predecessor YOLOv7 and contributing to enhanced breast cancer diagnostics.

Al-antari et al.'s [17] important study involved estimating a Full Resolution Convolutional Network (FrCN) using a Computer-Aided Diagnosis (CAD) model. Their method, which used X-ray mammography and a four-fold cross-validation, showed a high accuracy of 95.96%.

One approach to identifying breast cancer is called Diverse Features-Based Detection (DFeBCD), proposed by Chouhan and colleagues [18]. They assessed their method on the IRMA mammography dataset, and it attained an accuracy rate of 80.30% by combining an emotion learning-inspired integrated classifier (ELiEC) with the Support Vector Machine (SVM).

Through the application of the Lifting Wavelet Transform (LWT) for feature extraction from breast images, Muduli et al. [19] made a substantial contribution to the field. With the use of the Extreme Learning Machine (ELM) and moth flame optimization methodology, their method, which combined Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), produced remarkable accuracy rates of 98.76% and 95.80% on the DDSM and MIAS databases, respectively.

A technique for detecting breast cancer based on diversity analysis, geostatistics, and alpha form was presented by Junior et al. [20]. They achieved a 96.30% detection accuracy using the Support Vector Machine (SVM) classifier on the DDSM and MIAS datasets.

An approach for segmentation using a multigranulation rough set and intuitionistic fuzzy soft set was presented by Ghosh et al. [21]. Their method distinguished between malignant and unaffected tissue in mammograms, hence addressing ambiguity in pixels.

An effective Adaboost deep-learning technique for identifying breast cancer was created by Zheng et al. [22]. Their strategy, which combined multiple deep learning techniques with feature extraction and selection, produced a noteworthy 97.2%

accuracy.

Mahoro and Akhloufi investigated sophisticated deep-learning methods for breast mass identification, particularly the YOLO (You Only Look Once) framework. The promise of YOLO-based techniques in improving breast cancer diagnostics was demonstrated by the researchers by utilizing the VinDr-Mammo dataset and incorporating the YOLOv7 and YOLOv8 architectures.

III. METHODOLOGY

In this study, a dataset containing breast mass mammograms was obtained from Roboflow publically available [23]. The dataset was carefully observed, ensuring that each image was annotated with bounding boxes around the identified masses. Block diagram of overall system is shown in figure 1.

A. Preprocessing

Mammograms were preprocessed using functionality such as Auto-Orient and resizing the dataset. An auto-orient operation ensures that images are oriented correctly, and resizing is applied to stretch images to a constant size of 640x640 pixels.

CLAHE (Contrast Limited Adaptive Histogram Equalization) was applied to enhance mammogram quality and visualize masses. Preprocessed mammograms can be seen in figure 2

B. Data Splitting

The previously developed dataset consists of a total of 1025 mammographic images. This dataset is divided into three groups: training, validation, and testing, to ensure robust model training, performance validation, and unbiased evaluation.

The dataset is split as follows:

- **TRAIN SET:** 80% of the dataset, totaling 823 images, is allocated for training the model.
- **VALID SET:** 13% of the dataset, totaling 135 images, is reserved for validating the model's performance during training.
- **TEST SET:** 7% of the dataset, totaling 67 images, is kept separate for final evaluation on unseen data.

Furthermore, each training example underwent data augmentation to enhance the model's robustness. The augmentation process included horizontal flipping, resulting in three augmented outputs per training example.

Regarding the categories of the dataset, it comprises two classes: "mass" and "null." The "mass" class represents cases where a mass is present in the mammogram, indicating a potential abnormality, while the "null" class denotes mammograms without any detectable masses, indicating normal cases.

C. Model Selection and Training

The YOLOv8 model was selected for its superior ability in object detection tasks. The YOLOv8 architecture allows for the simultaneous detection of several objects in one image, which is convenient for efficient breast mass detection, so the pre-processed data set for model training is transferred to Google Colab and the YOLOv8 model is trained on the training set.

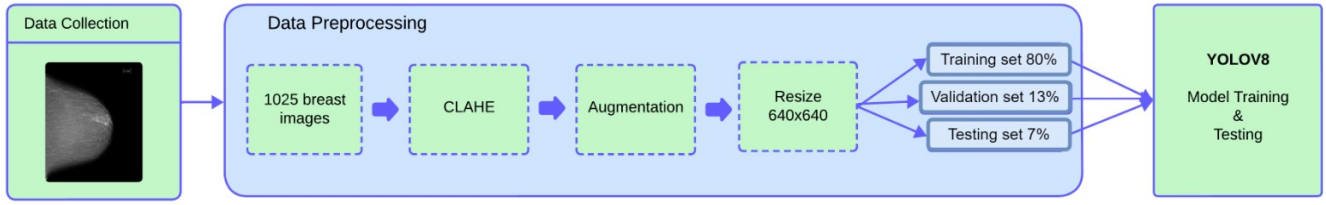


Fig. 1: Block Diagram of YOLOv8-based Breast Mass Detection System

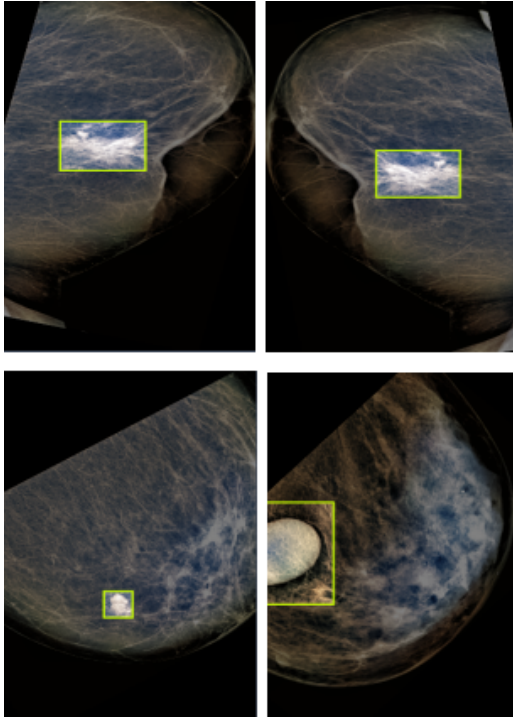


Fig. 2: Preprocessed Mammograms

1) *F1-Score*: F1-score is measured as the harmonic mean of precision and recall and is a valuable metric for assessing the balance between false positives and false negatives. It is calculated by the formula as shown in equation 1

$$F1 = \frac{2 \times P \times R}{P + R} \quad (1)$$

The F1 score for the breast mass detection system is 0.91. This score is an important indicator of the model's ability to achieve high accuracy and recall in breast mass detection, as shown in Figure 3.

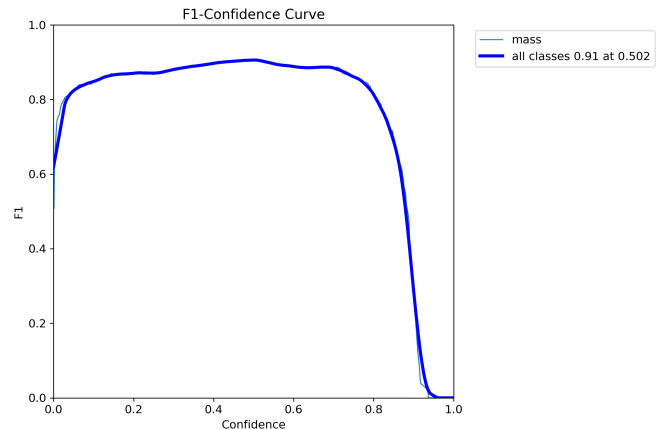


Fig. 3: F1-Score Variation with Confidence Threshold

Figure 3 also shows the variation of F1-score with different confidence limits. It provides insight into model performance at different confidence points, showing robustness in balancing accuracy and recall at different operating points.

2) *mAP (mean Average Precision)*: The map is calculated because it is a comprehensive measure that takes into account accuracy at different levels of confidence. The system achieved a commendable map of 0.942 as shown in Figure 4, confirming its high retention ability.

3) *Recall and Precision Confidence Curves*: Recall and precision confidence curves provide a detailed analysis of the model's behavior at different confidence limits, showing how precision and recall vary. This curve is necessary to understand the trade-off between precision and recall at different confidence levels.

Figure 5 shows the confidence Recall curve for the breast mass detection system. This curve represents the relationship

D. Testing Unseen Mammograms

To test the generalization ability of the model, it was tested on images that were not used during training or validation. This step was taken to measure and visualize the ability of the model to detect breast mass accurately in real-world and unseen scenarios.

IV. RESULTS AND DISCUSSIONS

The breast mass detection system using YOLOv8 was evaluated accurately and comprehensively, and the results showed its effectiveness in detecting and localizing masses in mammographic images.

A. Performance Metrics

To test the robustness and generalization of the system, the model is evaluated using matrices such as F1-scores, maps (average accuracy) and PR curves.

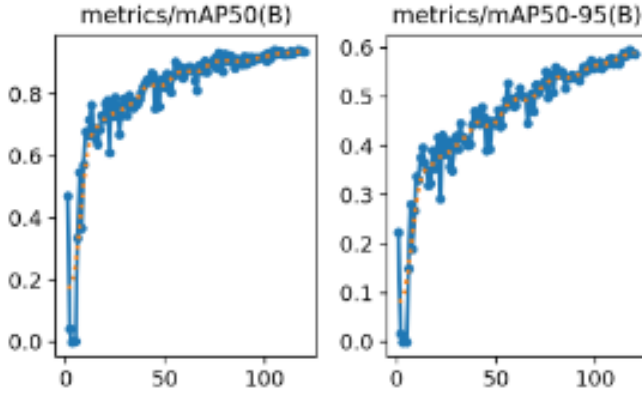


Fig. 4: Mean Average Precision with IoU Threshold on the x-axis and Mean Average Precision on the y-axis

between the return (R) and the confidence limit, helping to determine the optimal return point for the model without compromising accuracy.

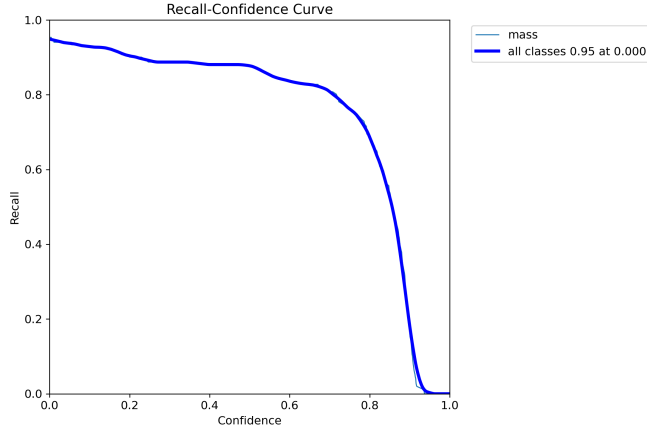


Fig. 5: Recall Confidence Curve for Breast Masses Detection

From equation 2 Recall (R) is defined as:

$$R = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (2)$$

Similarly, Figure 6 shows the precision confidence curve showing how precision (P) varies with different confidence limits. This curve helps to choose an appropriate operating point based on the desired balance between accuracy and recall.

The formula for Precision (P) is given in equation 3 as:

$$P = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \quad (3)$$

4) *PR Curve*: Likewise, the Precision-Recall (PR) curve is an important visual tool that shows the model's performance across various precision-recall tradeoffs. Precision is produced by plotting against recall at different confidence limits. The PR

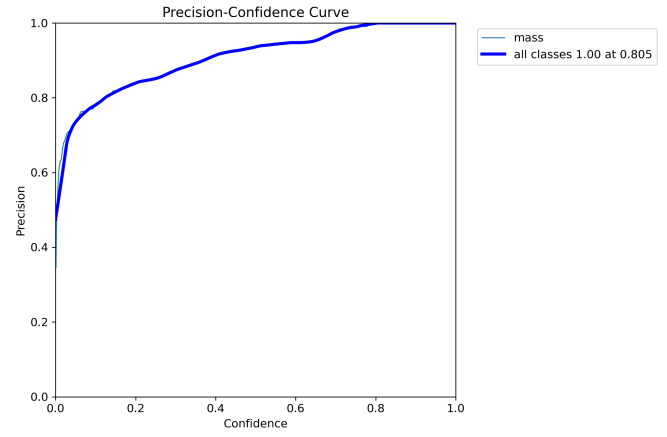


Fig. 6: Precision Confidence Curve for Breast Masses Detection

curve of this system is shown in Figure 7, which is equal to 1.00, indicating its ability to provide high accuracy even at high recall levels.

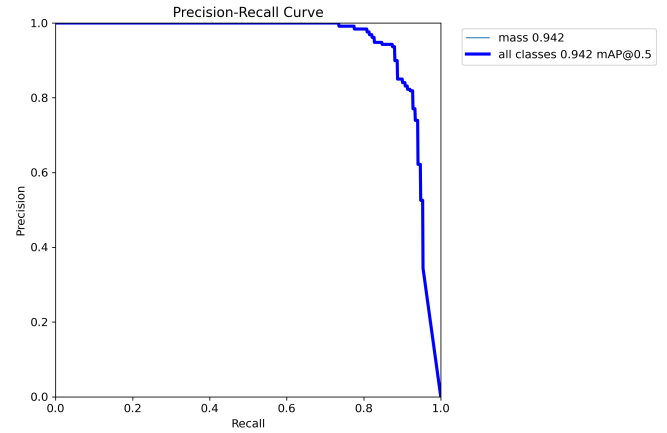


Fig. 7: Precision-Recall Curve for Breast Masses Detection

B. Overall System Performance

The overall performance of the system is carefully evaluated through a combination of quantitative measures, and various learning and evaluation metrics are considered to gain a deeper understanding of system performance. Figure 8 shows the plot of training and validation loss ($L_{\text{box}}, L_{\text{cls}}, L_{\text{diff}}$), recall and map (average accuracy) score during training.

- **Box-Loss (Bounding Box Loss)**: Measures the localization accuracy of predicted bounding boxes and it is defined by the formula mentioned in equation 4.

$$L_{\text{box}} = \frac{1}{N \cdot B} \sum_{i=1}^N \sum_{j=1}^B \left[(x_{i,j} - x_{i,j}^*)^2 + (y_{i,j} - y_{i,j}^*)^2 + (w_{i,j} - w_{i,j}^*)^2 + (h_{i,j} - h_{i,j}^*)^2 \right] \quad (4)$$

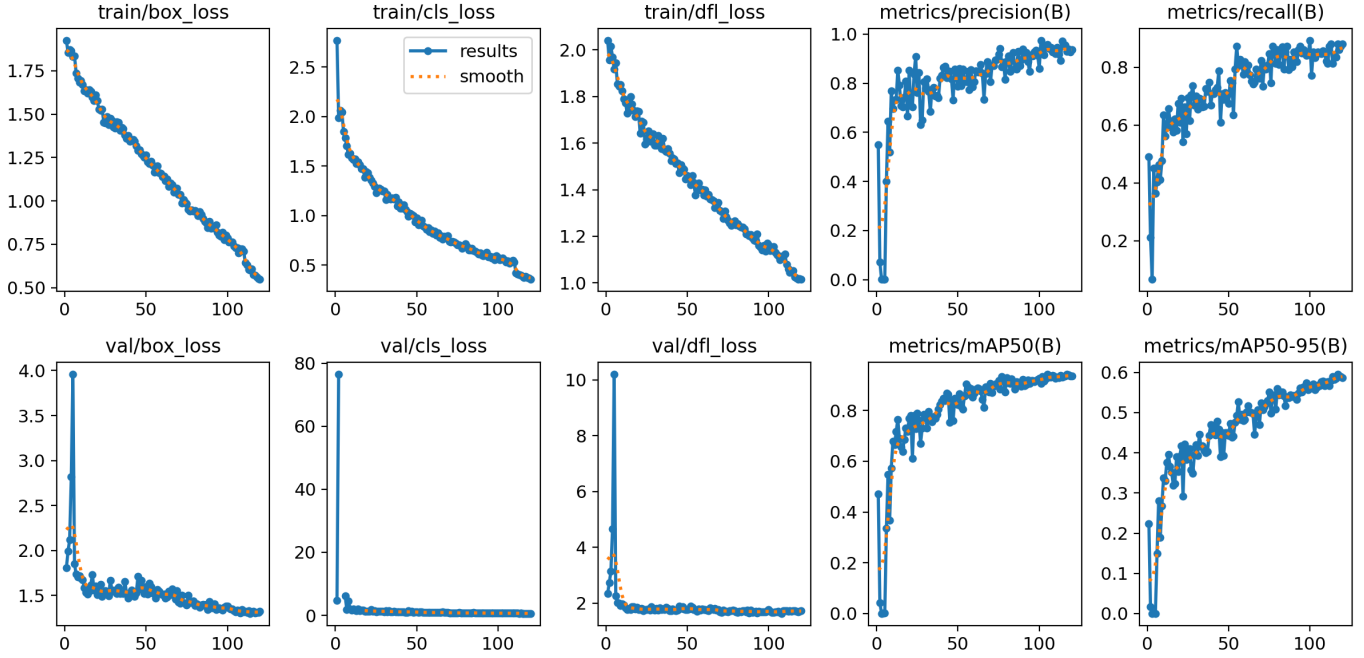


Fig. 8: Overall Results of Breast Masses Detection

N : Number of samples.

B : Number of bounding boxes.

$x_{i,j}, y_{i,j}, w_{i,j}, h_{i,j}$: Predicted coordinates.

$x_{i,j}^*, y_{i,j}^*, w_{i,j}^*, h_{i,j}^*$: Ground truth coordinates.

- **Cls-Loss (Classification Loss):** Evaluates the accuracy of object class predictions and it is calculated as in equation 5.

$$L_{\text{cls}} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C \left[y_{i,c} \cdot \log(p_{i,c}) + (1 - y_{i,c}) \cdot \log(1 - p_{i,c}) \right] \quad (5)$$

N : Number of samples.

C : Number of classes.

$y_{i,c}$: Ground truth class.

$p_{i,c}$: Predicted probability.

- **Dfl-Loss (Detection Focal Loss):** Represents the overall detection performance, combining localization and classification and its formula can be seen in equation 6.

$$L_{\text{dfl}} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C \left[y_{i,c} \cdot (1 - p_{i,c})^\gamma \cdot \log(p_{i,c}) + (1 - y_{i,c}) \cdot p_{i,c}^\gamma \cdot \log(1 - p_{i,c}) \right] \quad (6)$$

N : Number of samples.

C : Number of classes.

$y_{i,c}$: Ground truth class.

$p_{i,c}$: Predicted probability.

γ : Tunable parameter.

- **mAP (mean Average Precision):** Quantifies the precision-recall trade-off across different confidence thresholds. (Computed numerically using algorithms like the trapezoidal rule)

V. CONCLUSION

Our results demonstrate the effectiveness of the advanced YOLOv8-based breast mass detection system. This model shows promising performance in mass localization in breast mammograms showing the potential to help in the diagnosis of breast cancer. Validation of system results with radiologists strengthens its clinical utility and reliability. However, it is important to acknowledge that the quality and quantity of descriptive images play an important role in model performance. In Roboflow, a more comprehensive and annotated dataset can improve the accuracy and reliability of the model. Our collaboration with radiologists in the annotation process ensures that the database faithfully reflects real-world scenarios.

VI. FUTURE DIRECTIONS

For the future direction, it is recommended to expand interpretation efforts to include supplementary mammographic views like axillary tail, and tangential views. Adding these additional views to existing mammograms can significantly improve the model's ability to detect breast masses in a wider range of scenarios.

In addition, the study of advanced imaging modalities such as tomosynthesis and ultrasound may contribute to a more comprehensive and multimodal breast cancer detection system.

REFERENCES

- 1) Hamed, G., Marey, M. A. E. R., Amin, S. E. S., Tolba, M. F. (2020). Deep learning in breast cancer detection and classification. In Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020) (pp. 322-333). Springer International Publishing.
- 2) World Health Organization. (2020). Cancer: Key Facts. Retrieved from <https://www.who.int/news-room/fact-sheets/detail/cancer>.
- 3) Li, Q., Smith, J. (2018). Deep Learning in Medical Imaging: A Comprehensive Review. *Journal of Computer Assisted Tomography*, 42(2), 223-233.
- 4) Johnson, M., et al. (2020). Dataset Diversity in Deep Learning for Medical Image Analysis. *International Journal of Computer Vision*, 128(3), 777-795.
- 5) World Health Organization. (2020). Cancer: Key Facts. Retrieved from <https://www.who.int/news-room/fact-sheets/detail/cancer>.
- 6) Rebecca L. Siegel, Surveillance Research, American Cancer Society, 270 Peachtree Street, Atlanta, GA 30303, USA.
- 7) Gupta, R., Patel, K. (2019). Enhancing Mammogram Analysis with Deep Learning: A Comprehensive Review. *Journal of Medical Imaging and Diagnosis*, 12(3), 189-202.
- 8) Osareh, A., Shadgar, B. (2010). Machine learning techniques to diagnose breast cancer. In 2010 5th International Symposium on Health Informatics and Bioinformatics (pp. 142-146). Ankara, Turkey.
- 9) World Health Organization. (2020). Cancer: Key Facts. Retrieved from <https://www.who.int/news-room/fact-sheets/detail/cancer>.
- 10) Torre, L. A., et al. (2015). Global cancer statistics, 2012. *CA: A Cancer Journal for Clinicians*, 65(2), 87-108.
- 11) Ward, E., et al. (2004). Cancer disparities by race/ethnicity and socioeconomic status. *CA: A Cancer Journal for Clinicians*, 54(2), 78-93.
- 12) Dalaker, et al. (1999). Bureau of the Census, current population report, series P60–210. Poverty in the United States; 1997. U.S. Government Printing Office, Washington, DC.
- 13) Copeland, et al. (2015). Cancer in North America: 2008–2012. Volume one: combined Cancer incidence for the United States, Canada and North America. North American Association of Central Cancer Registries, Springfield.
- 14) Cheng, H. D., et al. (2010). Automated breast cancer detection and classification using ultrasound images: a survey. *Pattern Recognition*, 43(1), 299-317.
- 15) Al-Antari, M. A., et al. (2018). A fully integrated computer-aided diagnosis system for digital X-ray mammograms via deep learning detection, segmentation, and classification. *International Journal of Medical Informatics*, 117, 44–54.
- 16) Byra, M. (2021). Breast mass classification with transfer learning based on scaling of deep representations. *Biomedical Signal Processing and Control*, 69, 102828.
- 17) E. Mahoro and M. A. Akhloufi, "Breast masses detection on mammograms using recent one-shot deep object detectors," 2023 5th International Conference on Bio-engineering for Smart Technologies (BioSMART), Paris, France, 2023, pp. 1-4, doi: 10.1109/BioSMART58455.2023.10162036.
- 18) Al-Antari MA, Al-Masni MA, Kim TS. Deep Learning Computer-Aided Diagnosis for Breast Lesion in Digital Mammogram. *Adv Exp Med Biol*. 2020;1213:59-72. doi: 10.1007/978-3-030-33128-34.PMID : 32030663.ChouhanN, KhanA, ShahJZ, HussnainM,KhanMW.Deepconvolutionalneuralnetworkandemotionallerningbasedbreastcancerdetectionusingdigitalmammography.*ComputBiolMed*.2021May;132:104318
- 19) Muduli, D., Dash, R., Majhi, B. (2022). Automated diagnosis of breast cancer using multi-modal datasets: A deep convolution neural network-based approach. *Biomedical Signal Processing and Control*, Volume 71, Part B, 102825. ISSN 1746-8094. <https://doi.org/10.1016/j.bspc.2021.102825>.
- 20) Braz Junior, Geraldo; Rocha, S. V. ; Gattass, M.; Silva, Aristofanes Coorea; De Paiva, A. C.. A Mass Classification Using Spatial Diversity Approaches in Mammography Images for False Positive Reduction. *Expert Systems with Applications*,v. 40, p. 7534-7543, 2013
- 21) Ghosh, P., Mitchell, M., Tanyi, J. A., Hung, A. Y. (2016). Incorporating priors for medical image segmentation using a genetic algorithm. *Neurocomputing*, 195, 181-194. <https://doi.org/10.1016/j.neucom.2015.09.123>
- 22) Zheng, Jing Lin, Denan Gao, Zhongjun Wang, Shuang He, Mingjie Fan, Jipeng. (2020). Deep Learning Assisted Efficient AdaBoost Algorithm for Breast Cancer Detection and Early Diagnosis. *IEEE Access*. PP. 1-1. 10.1109/ACCESS.2020.2993536.
- 23) Breast Cancer. "Breast Cancer Dataset." Open Source Dataset. Roboflow Universe, Roboflow, May 2023. Available at: <https://universe.roboflow.com/breast-cancer-ce1zx/breast-cancer-jtuaz>. Accessed on 2024-03-07.

AI-driven Prediction of Electricity Production and Consumption in Micro-Hydropower Plant

Osman Safi
Electrical Engineering
UET Peshawar
 Kunar, Afghanistan
 osman.eep@uetpeshawar.edu.pk

Gul Muhammad Khan
National Center of AI
UET Peshawar
 Peshawar, Pakistan
 gk502@uetpeshawar.edu.pk

Gul Rukh Khattak
National Center of AI
UET Peshawar
 Peshawar, Pakistan
 gulrukhhattak@uetpeshawar.edu.pk

Abstract— Micro hydropower plants must effectively manage demand response to preserve operational firmness and prevent system breakdowns. This research focuses on accomplishing a fine balance while predicting consumption and production, which is significant for upholding system integrity. The study delves into predictive modelling methods to forecast patterns in the production and consumption of electricity over an array of time horizons. We set a baseline to resolve this and examined various algorithms, including convolution, dense, and recurrent neural networks. For the sake of Long Short-Term Memory (LSTM) neural networks, we utilize an autoregressive technique and integrate Cartesian Genetic Programming (CGP). The CGPLSTM forecasting output sequences with different time horizons precisely outperform the RNN-LSTM. The dataset utilized is downloaded from the Kaggle website. 50% of the data is used to train the models, and the rest is used to test the models. This work deals with the complex fluctuations in the demand response system and provides electricity production and consumption predictions.

CGP-LSTM model gave a training MAPE of 6.67 (Accuracy of 93.33%) and a testing MAPE of 6.68 (accuracy of 93.32%) for the next three hours; on the other hand, LSTM gave a training MAPE of 6.53 (accuracy of 93.47%) and testing MAPE of 7.46 (accuracy of 92.54%) for the next three hours.

The results offer a base for further developments and improvements in the field, drawing attention to more effective and reliable energy management capabilities in micro hydropower plants.

Keywords: artificial intelligence; micro-hydropower plant; time series forecasting; LSTM; CGP; hourly electricity prediction

I. INTRODUCTION

Among the most effective types of producing sustainable electricity is hydropower. Among the multiple advantages of these systems is the production of sustainable energy. However, different factors, like the volume of water in reservoirs, the cost of electricity, the temperature, and more, impact hydropower plants' efficiency. These factors make it further challenging to precisely forecast and suggest a power station's optimum functional outputs. For that reason, Accurate energy production predictions are vital for efficient scheduling, capacity planning, and collective management of power systems[1].

Many hydropower plants worldwide are considered small-scale run-of-river hydroelectric plants[2] This sort of hydroelectric plants utilize hydro energy to produce electrical energy, which is then used to provide electricity to the communities. 95% of the mean annual discharge of a river's flow can be deflected by the plants using a pipeline or tunnel that feeds the hydropower system's dynamos.

After the water has been utilized to produce electricity, it is sent back to the flow downstream.

Enhanced industry applications can be noticed due to incorporating the latest AI progress toward the energy sector. However, the need for more worldwide investment in AI research based on data models should be emphasized. Artificial intelligence (AI) has the power to facilitate utilities in giving consumers secure, renewable, and inexpensive electricity. In parallel, AI offers users the tools to optimize their energy use, encouraging more effective energy use habits[2].

Flowing water is the chief energy source for MHPs, giving sustainable electrical power solutions in rural areas. However, the capacity of MHPs to produce electricity is heavily affected by cyclic variations in water intake, which leads to an unevenness between production and consumption. The Jungle-Inn Micro-Hydroelectricity Plant in the Swat, Kalam region is one of many examples of how cold season water reduction causes decreased output and electricity production difficulties. This concept of forecasting electrical production and consumption at MHPs proposes establishing a load management system based on artificial intelligence (AI) that prioritizes power allocation to various customer classes, ensuring perfect utilization of resources and maintaining the MHP's integrity.

Given that they depend on unpredictable weather conditions and various temporal factors; renewable energy sources have significant intermittency. The patterns of consumer energy use are also closely related to sociocultural and environmental factors. Electrical load demand forecasting has become more complex and uncertain due to the increasing integration of irregular renewable energy sources and the complex dynamics of utility-customer behavior. As a result, load forecasting accuracy and precision become challenging tasks, particularly in dynamic environments with non-stationary behavior. This intricacy results from multidimensional dependencies on climatological, socio-environmental, and calendar-related elements[3].

II. BACKGROUND AND RELEVANT WORK

Precise electrical consumption forecasting models are requirements because of certain driving motives, the most obvious and serious being climate change. With information being published, carbon dioxide emissions are one of the prime reasons for climate change[4]. The significance of electrical power in everyday life means that forecasting its consumption is increasing in importance. Because of their universal application, a range of articles, study papers, blogs, and videos are accessible. Referring to Weron's[5] prediction

techniques, he examines several methods to handle the electrical energy forecast issue, including reduced form, statistical, and artificial intelligence, ML methodologies. It has been observed that Machine learning models frequently surpass many traditional approaches.

It can still be split into different computational methods (ML models); one uses deep learning models based on neural networks to explore time-variant data, and the other contains time series models focused on regression techniques[6]. The auto-regressive moving average is one of the regression techniques (ARMA)[7], and the moving average model that is integrated auto-regressive (ARIMA)[8] such models needs to have highly reliable data[6], which might not always be attainable.

Electric company's planning operations rely on accurate models for forecasting electric power consumption. An electric company may use consumption forecasting to assist in making important choices about the production and consumption of electricity, load switching, and industry development. Accurately forecasting consumption requirements is an electric power utility's main task. Energy is considered fundamental to the modern world and a core aspect of economic sustainability. A renewable energy resource supply is essential for economic growth. Most renewable energy sources, including wind, solar radiation, geothermal heat, hydropower, etc., are long-term sustainable. For instance, the hydroelectric turbine systems of large-scale traditional hydroelectric stations, or dams, with water reservoirs offer varying electricity production in response to variations in energy consumption. Atmospheric factors like precipitation and temperature influence small and micro hydropower plants' ability to generate energy. Due to the previously mentioned, the energy produced by these systems varies and must be predicted[9].

Lately, the usefulness of artificial intelligence techniques has overtaken that of traditional approaches, in particular in the domain of electricity consumption forecasting. Notably, ANNs have acquired significant prevalence and have been extensively used in this field. [11], [12].

As reported by Weron[5], Several prediction methodologies, including reduced-form, statistical, and computational intelligence methods like Machine Learning (ML), have been explored to address the electrical power forecasting challenges. ML models have performed better than conventional approaches in different circumstances. This finding is by the results drawn by Pallabi Paik et al.'s research [13], Which concentrates on stock price prediction; nevertheless, the research context is separate, and the resemblances between the data trend, data types, and setup in stock price prediction and electricity energy prediction suggest similar methods. Both domains concentrate on time series as the key element. The survey by Pallabi Paik et al. reveals that data-capturing technologies oftentimes outperform traditional techniques in multiple cases. These findings highlight the ability of machine learning approaches, including data mining, to offer more precise and firm predictions for complicated time series data like electricity energy consumption, outperforming the capabilities of conventional methods.

Deep learning models have shown better performance while operating on sequential data that show fickleness and volatility compared to traditional regression approaches. Notably, real-world data is frequently subject to dynamic alterations and instability. By means of experience-based data, research states the performance of artificial neural network models, namely the Long Short-Term Memory model, surpasses regression methods in such schemes.[6], [14], [15]. These results highlight the importance of deploying deep learning approaches, such as LSTM, to achieve more precise and reliable forecasting when dealing with diverse and non-stationary time series data.

III. METHODOLOGY

The study methodology follows a stepwise approach, using an artificial intelligence algorithm to forecast the consumption and production of micro hydropower plants. The initial research phase included finding a dataset that has historical data on the hourly-based consumption and production of electricity historical data. First, we went to Dare Noor, an MHP located in Nangarhar province, Afghanistan. The data we acquired for Dare Noor was insufficient to conduct a successful training. The data for Jungle-Inn was not available at first, so we looked it up on the Internet. Finally, we downloaded a dataset named "Hourly Electricity Consumption and Production." [17] The dataset has an hourly time series of electricity consumption and production data in Romania that spans over four years. All values are in Mega Watts.

After finding the dataset, the next phase involved data pattern inspection and evaluation. By assessing these patterns, we can discover crucial insights about power consumption and production in different conditions. This analytical process enabled us to identify anomalies, trends, and potential fields of improvement.

With the understanding obtained from the data analysis, we moved on to the development phase. We have utilized different algorithms, Dense, Convolution, Recurrent Neural Networks and Cartesian Genetic Programming designed for optimal predictions. The algorithms are destined to forecast the consumption and production of electricity based on historical data.

The established models, such as DNN, CNN, and RNN, are trained, validated, and tested. The dataset is divided into three parts, which are 70% of the data chosen for training, 20% of the data chosen for validation, and 10% of the data selected for the testing of the models. The training phase uses the data to train the model on scenarios and predictable responses. After the model is trained, it is validated against a separate set of samples from the dataset to ensure its generalization.

We additionally explore autoregressive methods utilizing RNN-LSTM and CGP-LSTM. The models are trained and tested using 50-50 data from the dataset. For executing the autoregressive approach, the model input includes observed and predicted values to predict the second and third hours of electricity. However, to predict the first hour, the

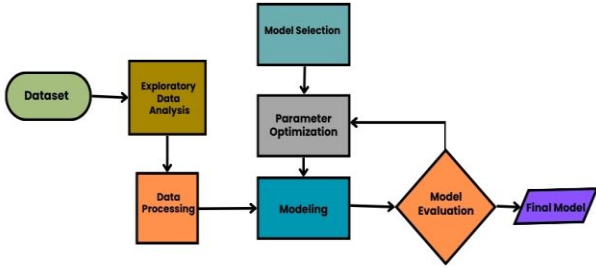


Figure 1: Research methodology adopted for this project[18]

models only used observed values from the dataset achieved by a custom sliding window approach.

Eventually, this approach will be favored to provide a permanent solution to the issues of load management during times of low water intake and excessive energy production, thus enhancing the reliability and efficiency of the micro hydropower station. The approach used in this study is implemented using a typical machine-learning project workflow[19] as demonstrated in Figure 1 above.

A. Data processing:

For ML algorithms, using preliminary data is often not suitable[20]. The initial data is usually unorganized and contains missing and noisy data, which might prevent algorithms from catching the patterns within the data during training. Data processing entails converting the data into a format that is suitable for the machine learning algorithms.

B. RNN in Time series:

The RNN, as we have already stated in the prior chapter, is, at the present time, being executed in natural language processing to resolve issues with language modeling, machine translation, language understanding, and other issues. Extensive research has been done on using RNN for sequential data analysis, even in the field of failure prediction [21] Deciding the right timeframe for predicting short-term sustainable energy sources [22] and more.

C. Long short-term memory (LSTM) neural network model:

The RNN consists of a particular variation named the LSTM model. This model discriminates itself by using memory units that encourage the refreshing of prior hidden states, memory that is long-term. Every neuron in this architecture gets feedback on its previous state. The output of RNN is persuaded not only by the input and weight of the neuron but also by the feedback from prior neurons. This ability allows the model to detect temporal relationships throughout long sequences. The LSTM model makes effective use of problems like the exploding and vanishing gradient concerns that often affect traditional RNN training. The gate mechanism and internal memory unit are crucial in mitigating these problems. The LSTM is organized with four primary components: The Input Gate, Output Gate, Forget Gate, and Cell State.

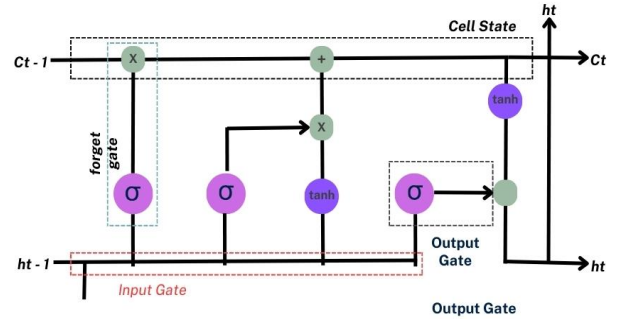


Figure 2: The LSTM model's internal structure[4]

The gates all together manage the preservation and updating of the data stored in the cell state [23].

An LSTM cell's structure is depicted in Figure 3.2. The computational method can be described as [24] :

$$ft = \sigma(wf [ht - 1, Xt] + bf) \quad (1)$$

$$it = \sigma(wi [ht - 1, Xt] + bi) \quad (2)$$

$$ot = \sigma(wo [ht - 1, Xt] + bo) \quad (3)$$

$$at = \tanh(wa [ht - 1, Xt] + ba) \quad (4)$$

$$ct = ft * ct - 1 + it * at \quad (5)$$

$$ht = ot * \tanh(ct), \quad (6)$$

The sigmoid activation function, represented by σ , is defined as follows:

$$\sigma(x) = (1 + e^{-x})^{-1} \quad (7)$$

The symbols ft , it , and ot signify the output values of the forget, input, and output gates in the context of the LSTM architecture. The memory cell is represented by the symbol ct , while its updating and activation process is denoted as at . The vectors ht and Xt correspond to the output and input vectors at time t , respectively.

The matrices Wf , Wi , Wa , Wo along with the bias vectors bf , bi , ba , bo is employed to parameterize the weights and biases within the LSTM model.

D. Cartesian genetic programming evolved recurrent neural network (CGPRNN):

Recurrent networks play a crucial role in numerous nonlinear and dynamic system operations. Khan et al. [25] proposed the CGPRNN. Cartesian genetic programming, or CGP, is used by CGPRNN to evolve RNNs. Rothemich and Miller developed the genetic programming method called Cartesian genetic programming [26]. In CGP, a digital circuit or computer software is designed using a two-dimensional graphical representation. CGP is an extraordinarily adaptable and productive genetic programming method. Arrays and the Cartesian structure are used by CGP to represent its architecture[27].

A neuro-evolutionary algorithm called CGPRNN is used to evolve recurrent neural networks. It makes use of CGP's exceptional ability to generate a recurrent artificial neural network. CGPRNN differs from other types of CGPANN in that it makes use of a feedback mechanism, which involves feeding one or more outputs returned into the system as input.

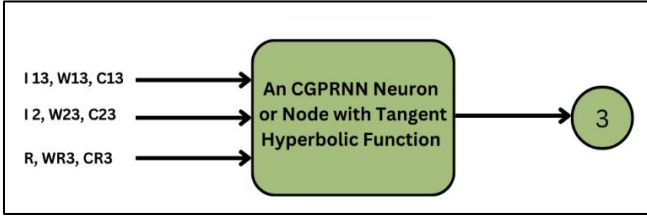


Figure 3: CGPRNN node with 3 inputs [5]

Because the neurons formed from the CGPRNN network are not fully linked, the topology of this network differs from that of a regular RNN. Additionally, Programme inputs are not given to any of the input layer neurons. This feature makes it possible for CGPRNN to generate topologies that are efficient to implement in terms of both hardware and computational costs[28].

The nodes in the CGPRNN genotype stand in for the RNN's neurons. The nodes are associated with certain inputs, connecting functions, and weights, as seen in Figure 3.4.

Three different sorts of inputs are provided: Programme inputs, inputs originating from earlier nodes, and inputs derived via feedback. The first layer of the genotype in CGPRNN has two types of inputs: recurrent inputs and system inputs. Whether or not a feedback input is chosen through mutation to be used as a node input determines whether or not a feedback path will exist for the subsequent layers.

If a node's connection value is one, it is considered to be connected; if it is zero, it is considered to be disconnected. Between -1 and +1, the weights of the connections are generated at random. Nonetheless, the input weight of feedback is consistently +1. The weights of the linked inputs are added after all of the inputs have been multiplied and added. It then moves on to an activation function, which can be either linear or non-linear and represents a step function, linear, tangent hyperbolic, linear, or log-sigmoid function.

Every node's ability to provide output is dependent on the activation function. The output of the node is then used as either the system output or the node that follows it as an input. Either any node output or the Programme input might be considered the genotype's output(s). If the recurrent input is already connected, the genotype output can be fed back into the nodes. After that, the CGPRNN genotype continuously evolved by mutation until the target fitness is reached. The resulting genotype is then converted to the final RNN once The weights and linkages of state units are frozen [27].

E. Setup of a neural network model:

1) Dropout

Deep learning neural networks possess a formidable capacity for memory retention. Nonetheless, the phenomenon of overfitting arises due to the network's inclination to acquire intricate features from data, thereby impeding generalizability. To tackle this challenge, dropout has emerged as a highly esteemed regularization technique. By intermittently nullifying the output of select hidden units, dropout seeks to mitigate the intricacies inherent in neural network structures, consequently curtailing complexity [29].

2) Scaling:

a) Normalization

Normalization is used to transform the numerical column values in the dataset into a common scale without affecting the variations within the value ranges. Datasets with features that fall across a wide range can be normalized [30].

Utilizing Sklearn's min max scaler, which is based on the formula that follows, the data gets scaled so that all values are between 0 and 1.

b) Standardization

Standardization or Z-Score: The mean is calculated and divided by the standard deviation to normalize characteristics. The Z-score is the term used to describe this. When the data has a Gaussian distribution, standardization may be useful. That being said, this need not always be the case. From a geometric perspective, it compresses or enlarges the points if std is 1, and it converts the data to the original data's mean vector at the origin. As we can see, we are merely changing the mean and standard deviation to a standard normal distribution, which is still normal; the distribution's form remains unchanged. Standardization is unaffected by outliers because the modified characteristics do not fall inside a predefined range.

F. Indicators for performance evaluation:

After creating the model, it's critical to put it through an evaluation procedure based on predetermined standards to determine its performance and accuracy. This is frequently accomplished using pre-defined functions that enable you to obtain a numerical number that aids in determining the efficacy of a particular model and permits model comparisons. The following parameters were measured and utilized in order to achieve that:

- MAE
- MAPE

a) MAE

The computation of the Mean Absolute Error (MAE) involves the amalgamation of forecasted values (denoted as Ft) and actual observed values (denoted as At) across a set of observations (n). This metric encapsulates the measure of dissimilarity between predicted and real values, quantifying the magnitude of the discrepancies. The MAE is calculated through the utilization of the formula represented below:

$$MAE = \frac{1}{n} \sum_{t=1}^n |Ft - At| \quad (8)$$

MAE's strength lies in its sensitivity to error magnitudes, yet its weakness is its inability to provide error direction, making comparisons across datasets complex [31].

b) MAPE

One quantitative measure used to evaluate the accuracy of forecasting techniques is the MAPE. It measures the mean of the absolute percentage differences between the values that were predicted and those that were observed in a dataset, evaluating the forecasts' accuracy with respect to the actual quantities. MAPE proves particularly valuable in the analysis of substantial datasets and necessitates the application of non-zero dataset values for meaningful evaluation.

$$MAPE = (1/n) * \Sigma [(Actual - Forecast) /$$

$$|Actual|) * 100] \quad (9)$$

Where:

The sample size (n) is the total number of data points.

Σ denotes the summation symbol, suggesting that for each data point, the expression included in brackets should be added up.

The absolute difference between the predicted and actual numbers is shown as $|Actual - Forecast|$. The absolute value of the actual value is shown by $|Actual|$. The result is expressed as a percentage error, reflecting the average percentage difference between the forecasted and actual values across the dataset **Error! Reference source not found.**

IV. RESULTS

The primary goal is to predict power production and consumption for the next three hours. To achieve this a sliding window approach is used, where we use observed and predicted values as an input for prediction. This approach is practical for real-world applications, especially micro-hydro power stations.

The dataset that was utilized for this project spans four years, from 2019 to 2023, and has three columns: "DateTime," "Consumption," and "Production" in MWh. We use different types of models for this task. Initially, we start with simple models to establish a baseline. Then, we explore more models, including Convolutional, DNN, and Recurrent Neural Networks. These models make all their predictions in a single shot (all 24 hours prediction at a single shot), unlike a sliding window approach where we predict one hour at a time and then we make the predicted hour part of an input, the input then have observed and predicted values to predict the next hour.

In the final phase, we introduce an approach using a custom sliding window technique with LSTM and the novel algorithm called CGPRNN. We use the MAE and MAPE to assess the effectiveness of both forecasting models. baseline model, linear model, dense model, CNN model, and RNN models.

A. Custom sliding window approach

In the Custom Sliding Window methodology, our approach entails leveraging the preceding 12 hours of electricity consumption to predict the subsequent 3 hours. However, rather than providing a singular predictive output, our method involves incorporating both observed and predicted values in the input sequence. This iterative process involves predicting one hour at a time, with each predicted value being appended to the input sequence for the subsequent prediction.

Dummy Train Data	6352	6116	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	5776
	Train Data													
	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	Prediction	Actual
Core Model	6352	6116	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	Y1	X13
Update core model	6116	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	Y1	Y2	X14
Update core model	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	Y2	Y3	X15
Update core model	6116	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	Y4	X14
Update core model	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	Y4	Y5	X15
Update core model	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	Y4	Y5	Y6	X16
Update core model	5873	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	5776	Y7	X15
Update core model	5682	5557	5525	5513	5524	5510	5617	5643	5743	5737	5776	Y7	Y8	X16
Update core model	5557	5525	5513	5524	5510	5617	5643	5743	5737	5776	Y7	Y8	Y9	X17

Figure 4: Sliding window mechanism used in this project

Subsequently, upon completing a prediction cycle, the window of observed values is shifted by unit size, and the process is reiterated until the model is sufficiently trained. The mechanism is depicted in the following picture. Figure 4.10 shows some input data for the model through which we train the model, based on 12 hours, we predict the next hour, so from X1 to X12 is the input data; in the Prediction column we have the model's predicted values shown by y1, y2,...yn, in Actual column we have the actual values from the historical data we then find out the error by (Actual - Predicted) values and in Core Model column the model gets updated with each iteration.

This iterative approach involves utilizing both LSTM and a novel algorithm known as Cartesian Genetic Programming for training and testing purposes. However, we train our model using half of the available data and test it on the other half.

Given the iterative nature of this methodology, which necessitates considerable computational resources for the purpose of identifying optimal hyper parameter combinations, a series of extensive experiments has been diligently conducted. In the following section I have only mentioned two experiments that provided comparatively good results.

1) RNN-LSTM

In the experiments to train LSTM model, we have used standardization as a normalization technique, we used 50% of data to train the model and 50% to test the model on it. We used 12 nodes with ReLU activation function.

Here's a summary of the steps in the provided code:

- Import necessary libraries including NumPy, Pandas, and TensorFlow for building and training a neural network model.
- Define a function `split_sequence` (sequence, n_steps) to split a univariate time series sequence into input-output pairs with a specified number of time steps (n_steps).
- Set the number of time steps (n_steps) for sequence splitting and define a neural network model (core_model) using TensorFlow's Keras API.
- Compile the core_model with the adam optimizer and mean squared error loss.
- Define column names for the result DataFrames for both training and testing.
- Create empty DataFrames `Result_Train` and `Result_Test` to store the training and testing results, respectively.
- Define a function `Model_Train` (sequence, model) to train the neural network model on a given sequence. This function fits the model to the data, makes predictions, and appends the predicted value to the sequence.
- Define a function `seq_train` (raw) to perform training iteratively. It calls `Model_Train` for training the model on subsequences of the training data, appends the results to `Result_Train`, and updates the input sequence for the next iteration.

- Set the number of iterations and the raw data length initial values based on the length.
- Iterate through the training data, extracting subsequences and applying the seq_train function.
- Extract test data from the remaining portion of the raw data.

Following figures 4.11 and 4.12 displays the model's training and testing curves for 7000 data points.

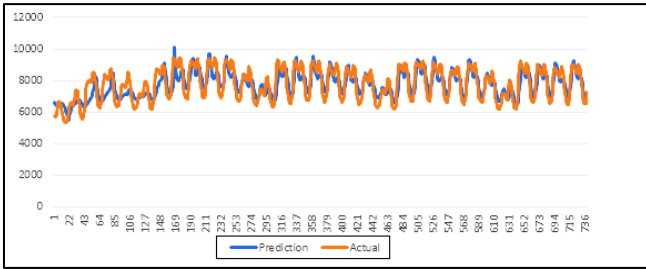


Figure 5: The training result of the model for 7k rows

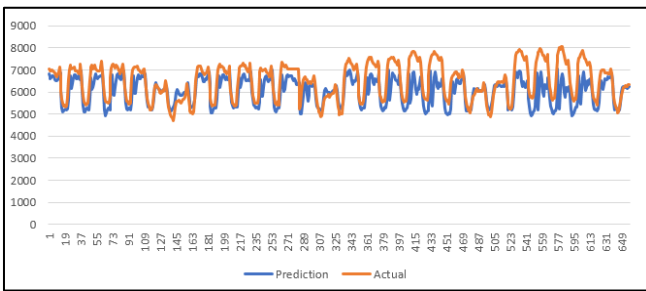


Figure 6: The testing result of the model for 7k rows

The following Figures 4.13 and 4.14 shows the model performance when trained on 4k data samples.

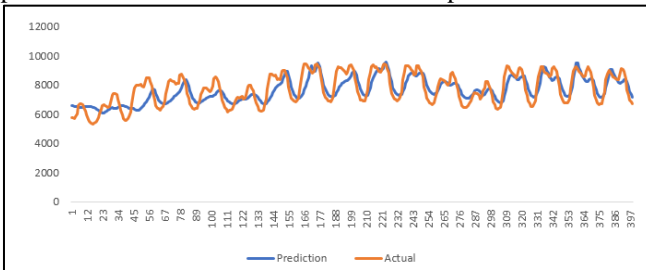


Figure 7: The training curves for 4000 rows

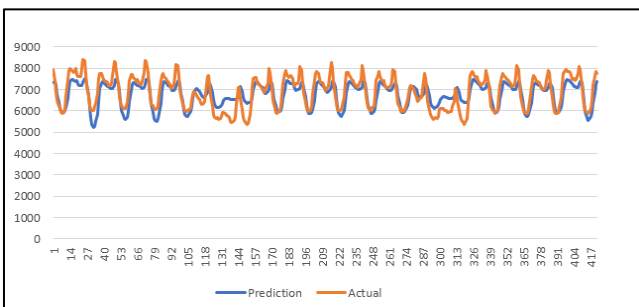


Figure 8: Testing curves for 4k rows

TABLE 1: LSTM MODEL RESULTS

Model	Training/Test ing	Input/output	MAE	MAPE
4k rows	Train C	12 inputs 1 output	363.47	4.86
	Test C		324.41	5.04
	Test P		340.91	5.21
	Train C	12 inputs 3 outputs	489.87	6.53
	Test C		450.11	7.46
	Test P		445.65	6.75
7k rows	Train C	12 inputs 1 output	279.813	3.92
	Test C		465.18	7.01
	Test P		392.77	6.69
	Train C	12 inputs 3 outputs	382.16	5.36
	Test C		678.81	10.2
	Test P		586.93	9.95

Table 4.2 presents a performance comparison of both of the experiments. The models are trained and tested on 4k and 7k data samples, providing one- and three-hours predictions into the future. The errors are presented in Megawatt hours (MWh).

The model is trained and tested on both “Consumption” and “Production,” as you can see in the above table 4.2. The model is run on 4k and 7k rows from the dataset. “Train C” represents that the model is trained on “Consumption” historical data, “Test C” indicates that the model is tested on “Consumption” data, and “Test P” represents that the model is tested on “Production” data. In the experiment with 4k data points, we obtained an accuracy of **93.47 %** for training and **92.54%** for testing to predict the next three hours. The best results in the table are shown in bold text.

2) Cartesian genetic programming recurrent neural network

The CGP-RNN training was carried out on the production data, and validation was performed on both the training and testing data. Figure 4.18 shows the actual and predicted values; the predicted values curve follows the actual values curve very closely.

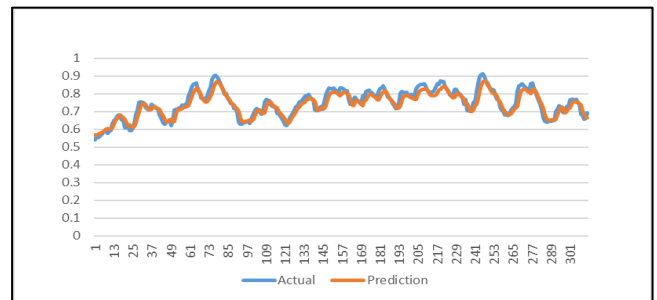


Figure 9: Testing result curve with CGP-RNN algorithm

TABLE 2: RESULTS OF CGPRNN MODEL

Model	Training/ Testing	Input /Output	MAE	MAPE
50 Nodes Full dataset	Train- P	6 input 6 Output	0.064384	9.407604
	Test- P		0.062959	9.409408
	Test- C		0.048834	7.790334
	Train- P	6 input 3 output	0.045892	6.679367
	Test- P		0.044945	6.684573
	Test- C		0.035013	5.558505
	Train- P	6 input 1 one output	0.031806	4.589071
	Test- P		0.031083	4.583583
	Test- C		0.024525	3.869808

CGPRNN used 50 nodes, although the actual nodes used in the model are fewer. The model is trained on “Production” column from the dataset then it is tested on both “Production” and “Consumption”, Train- P (Model trained on “Production”), Test-P (Model test on “Production) and Test –C (Model test on “Consumption”). As you see in Table 4.4, we have six inputs and 6 outputs, which means that based on the previous six hours, we are predicting the next 6 hours. Then we have three inputs and three outputs, which means that the model predicts the next three hours based on previous six hours which have both observed and predicted values as explained in the custom sliding window approach. Finally, we have six inputs and one output, which means that we are predicting the next hour based on the previous six hours. When the model was first trained and tested on the entire dataset, the outcomes are displayed in Table 4.4.

V. CONCLUSION:

In conclusion, the various approaches and algorithms employed in this study present unique advantages and drawbacks, contributing to a detailed understanding of their applicability. Convolutional Neural Networks and Recurrent Neural Networks in the prediction process offer a robust foundation for capturing spatial and temporal dependencies in the data. CNNs stand out in extracting spatial features, whereas RNNs are adept at modelling temporal patterns. However, the dependency on vast training data and the possibility of overfitting are noteworthy drawbacks. Moreover, the autoregressive nature of the models presents challenges in precisely predicting distant subsequent values. Finding a balance between model complexity and forecasting accuracy is a key concern across all approaches. These insights help the continued discussion about the optimization of predictive modelling for electricity consumption and production prediction, leading the way for future improvements and refinements in the field. On the other hand, with its iterative prediction methodology using LSTM and CGP, the Custom Sliding Window approach excels in its adaptability in yielding an output of varying lengths. This adaptability verifies advantageous in scenarios demanding several output predictions. However, the recurrent nature of the approach may present higher computational complexity. Comparing the results of these two established models, CGP-RNN gave good results compared to LSTM. However, it

must be mentioned that both models can be improved by experimenting with different combinations of hyper parameters. CGP-RNN gave a training MAPE of **6.67** and a testing MAPE of **6.68** for the next three hours; on the other hand, LSTM gave a training MAPE of **6.53** and a testing MAPE of **7.46** for the next three hours.

We have validated The LSTM model on the Jungle-Inn dataset, which contains hourly data spanning 65 days. The results are promising, and in the future, when more data is available, the same method can be extended.

VI. REFERENCES:

- [1]. J. Barzola-Monteses, J. Gómez-Romero, M. Espinoza-Andaluz, and W. Fajardo, “Hydropower production prediction using artificial neural networks: an Ecuadorian application case,” *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13253–13266, Aug. 2022, doi: 10.1007/s00521-021-06746-5.
- [2]. C. L. T. Borges and R. J. Pinto, “Small Hydro Power Plants Energy Availability Modeling for Generation Reliability Evaluation,” *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1125–1135, Aug. 2008, doi: 10.1109/TPWRS.2008.926713.
- [3]. G. Notton *et al.*, “Intermittent and stochastic character of renewable energy sources: Consequences, cost of intermittence and benefit of forecasting,” *Renew. Sustain. Energy Rev.*, vol. 87, pp. 96–105, May 2018, doi: 10.1016/j.rser.2018.02.007.
- [4]. A. K. Singh, Ibraheem, S. Khatoun, Md. Muazzam, and D. K. Chaturvedi, “Load forecasting techniques and methodologies: A review,” in *2012 2nd International Conference on Power, Control and Embedded Systems*, Allahabad, Uttar Pradesh, India: IEEE, Dec. 2012, pp. 1–10. doi: 10.1109/ICPCES.2012.6508132.
- [5]. R. Weron, “Electricity price forecasting: A review of the state-of-the-art with a look into the future,” *Int. J. Forecast.*, vol. 30, no. 4, pp. 1030–1081, Oct. 2014, doi: 10.1016/j.ijforecast.2014.08.008.
- [6]. M. Rafik, A. Fentis, T. Khalili, M. Youssfi, and O. Bouattane, “Learning and Predictive Energy Consumption Model based on LSTM recursive neural networks,” in *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*, Fez, Morocco: IEEE, Oct. 2020, pp. 1–7. doi: 10.1109/ICDS50568.2020.9268733.
- [7]. J.-F. Chen, W.-M. Wang, and C.-M. Huang, “Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting,” *Electr. Power Syst. Res.*, vol. 34, no. 3, pp. 187–196, Sep. 1995, doi: 10.1016/0378-7796(95)00977-1.
- [8]. G. Juberias, R. Yunta, J. Garcia Moreno, and C. Mendivil, “A new ARIMA model for hourly load forecasting,” in *1999 IEEE Transmission and Distribution Conference (Cat. No. 99CH36333)*, New Orleans, LA, USA: IEEE, 1999, pp. 314–319 vol.1. doi: 10.1109/TDC.1999.755371.
- [9]. J. A. e Konica and E. Staka, “Forecasting of a hydropower plant energy production with Fuzzy logic Case for Albania,” vol. Vol. 4, no. Issue 5, p. 5, May 2017.
- [10]. J.-F. Chen, S.-K. Lo, and Q. Do, “Forecasting Monthly Electricity Demands: An Application of Neural Networks Trained by Heuristic Algorithms,” *Information*, vol. 8, no. 1, p. 31, Mar. 2017, doi: 10.3390/info8010031.
- [11]. K. Kandanand, “Forecasting Electricity Demand in Thailand with an Artificial Neural Network Approach,” *Energies*, vol. 4, no. 8, pp. 1246–1257, Aug. 2011, doi: 10.3390/en4081246.
- [12]. N. Amjady and F. Keynia, “A New Neural Network Approach to Short Term Load Forecasting of Electrical Power Systems,” *Energies*, vol. 4, no. 3, pp. 488–503, Mar. 2011, doi: 10.3390/en4030488.
- [13]. D. P. Gandhmal and K. Kumar, “Systematic analysis and review of stock market prediction techniques,” *Comput. Sci. Rev.*, vol. 34, p. 100190, Nov. 2019, doi: 10.1016/j.cosrev.2019.08.001.
- [14]. E. Yuniarti, N. Nurmaini, B. Y. Suprpto, and M. Naufal Rachmatullah, “Short Term Electrical Energy Consumption Forecasting using RNN-LSTM,” in *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Batam Island, Indonesia: IEEE, Oct. 2019, pp. 287–292. doi: 10.1109/ICECOS47637.2019.8984496.

- [15]. F. U. M. Ullah, A. Ullah, I. U. Haq, S. Rho, and S. W. Baik, "Short-Term Prediction of Residential Power Energy Consumption via CNN and Multi-Layer Bi-Directional LSTM Networks," *IEEE Access*, vol. 8, pp. 123369–123380, 2020, doi: 10.1109/ACCESS.2019.2963045.
- [16]. A. B. Georges Hebrail, "Individual household electric power consumption." [object Object], 2006. doi: 10.24432/C58K54.
- [17]. S. COMANITA, "Hourly Electricity Consumption and Production." [Online]. Available: <https://www.kaggle.com/datasets/stefancomanita/hourly-electricity-consumption-and-production>
- [18]. D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python: implement advances deep learning and neural network models using TensorFlow and Keras*. Birmingham Mumbai: Packt, 2018.
- [19]. S. Fallah, R. Deo, M. Shojafar, M. Conti, and S. Shamshirband, "Computational Intelligence Approaches for Energy Load Forecasting in Smart Energy Management Grids: State of the Art, Future Challenges, and Research Directions," *Energies*, vol. 11, no. 3, p. 596, Mar. 2018, doi: 10.3390/en11030596.
- [20]. I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research Directions," *SN Comput. Sci.*, vol. 2, no. 3, p. 160, May 2021, doi: 10.1007/s42979-021-00592-x.
- [21]. F. A. Olivencia Polo, J. Ferrero Bermejo, J. F. Gómez Fernández, and A. Crespo Márquez, "Failure mode prediction and energy forecasting of PV plants to assist dynamic maintenance tasks by ANN based models," *Renew. Energy*, vol. 81, pp. 227–238, Sep. 2015, doi: 10.1016/j.renene.2015.03.023.
- [22]. E. İzgi, A. Öztopal, B. Yerli, M. K. Kaymak, and A. D. Şahin, "Determination of the Representative Time Horizons for Short-term Wind Power Prediction by Using Artificial Neural Networks," *Energy Sources Part Recovery Util. Environ. Eff.*, vol. 36, no. 16, pp. 1800–1809, Aug. 2014, doi: 10.1080/15567036.2011.561274.
- [23]. S. Mahjoub, L. Chrifi-Alaoui, B. Marhic, and L. Delahoche, "Predicting Energy Consumption Using LSTM, Multi-Layer GRU and Drop-GRU Neural Networks," *Sensors*, vol. 22, no. 11, p. 4062, May 2022, doi: 10.3390/s22114062.
- [24]. M. Tovar, M. Robles, and F. Rashid, "PV Power Prediction, Using CNN-LSTM Hybrid Neural Network Model. Case of Study: Temixco-Morelos, México," *Energies*, vol. 13, no. 24, p. 6512, Dec. 2020, doi: 10.3390/en13246512.
- [25]. N. M. Khan and G. M. Khan, "Multi-chromosomal CGP-evolved RNN for signal reconstruction," *Neural Comput. Appl.*, vol. 33, no. 20, pp. 13265–13285, Oct. 2021, doi: 10.1007/s00521-021-05953-4.
- [26]. R. Joseph and J. Miller, *Studying the Emergence of Multicellularity with Cartesian Genetic Programming in Artificial Life*. 2002.
- [27]. G. M. Khan, J. F. Miller, and M. Mahsal Khan, "Evolution of Optimal ANNs for Non-Linear Control Problems using Cartesian Genetic Programming," 2010. [Online]. Available: <https://api.semanticscholar.org/CorpusID:9538954>
- [28]. G. M. Khan and F. Zafari, "Dynamic feedback neuro-evolutionary networks for forecasting the highly fluctuating electrical loads," *Genet. Program. Evolvable Mach.*, vol. 17, no. 4, pp. 391–408, Dec. 2016, doi: 10.1007/s10710-016-9268-6.
- [29]. L. Wen, K. Zhou, and S. Yang, "Load demand forecasting of residential buildings using a deep learning model," *Electr. Power Syst. Res.*, vol. 179, p. 106073, Feb. 2020, doi: 10.1016/j.epsr.2019.106073.
- [30]. "How, When, and Why Should You Normalize / Standardize / Rescale Your Data?" [Online]. Available: <https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff>
- [31]. R. G. Pontius, O. Thontteh, and H. Chen, "Components of information for multiple resolution comparison between maps that share a real variable," *Environ. Ecol. Stat.*, vol. 15, no. 2, pp. 111–142, Jun. 2008, doi: 10.1007/s10651-007-0043-y.
- [32]. S. Bermejo and J. Cabestany, "Oriented principal component analysis for large margin classifiers," *Neural Netw.*, vol. 14, no. 10, pp. 1447–1461, Dec. 2001, doi: 10.1016/S0893-6080(01)00106-X.

Meta-Space: Pioneering Education in the Metaverse

Ibrahim Khalid

Dept. of Computer Systems
Engineering

University of Engineering and
Technology, Peshawar Pakistan
20pwce1970@uetpeshawar.edu.pk

Umair Adnan

Dept. of Computer Systems
Engineering

University of Engineering and
Technology, Peshawar Pakistan
20pwce1960@uetpeshawar.edu.pk

Hussnain Sajjad

Dept. of Computer Systems
Engineering

University of Engineering and
Technology, Peshawar Pakistan
20pwce1958@uetpeshawar.edu.pk

Muhammad Abeer Irfan

Dept. of Computer Systems
Engineering

University of Engineering and
Technology, Peshawar Pakistan
abeer.irfan@uetpeshawar.edu.pk

Yaser Ali Shah

Department of Computer Science,
COMSATS University Islamabad,
Attock Campus, Attock 43600,
Pakistan Yaser@cuiatk.edu.pk

Abid Iqbal

Dept. of Electrical Engineering
University of Engineering and
Technology, Peshawar Pakistan
abid.iqbal@uetpeshawar.edu.pk

Abstract—In the evolving landscape of learning methodologies, technology has emerged as a catalyst, transforming the educational experience. This study delves into the realm of Virtual Reality (VR) and Augmented Reality (AR), collectively referred to as the "Metaverse," as a pivotal tool in education. By conducting systematic literature reviews, we investigate the potential, effectiveness, and associated pros and cons of employing the Metaverse for learning. Our findings affirm that the Metaverse proves to be a highly effective learning platform, enhancing engagement through lifelike avatars and bridging the gap between the real and virtual worlds. While this innovative approach facilitates visualizing materials and fosters interactive and interesting learning environments, challenges such as the cost of requisite devices remain. Despite limitations, the advantages of integrating the Metaverse into education are evident, necessitating ongoing development to amplify benefits and address existing constraints. This research contributes valuable insights to the ongoing discourse on leveraging Metaverse technologies for enriching educational practices.

Keywords: Metaverse; Virtual Classroom; Education; Communication.

I. INTRODUCTION

Learning is a lifelong journey, an ever-present thread woven into the fabric of our lives, guiding us to new insights and knowledge, regardless of age. From our earliest days, the pursuit of knowledge has been a fundamental aspect of our existence, often shaped by formal education. However, the methods through which we learn vary in their ability to captivate our interest. Traditional learning materials, dominated by text with sparse illustrations, often present a challenge to making the learning experience truly enjoyable and engaging.

In the contemporary landscape, the rapid evolution of technology has revolutionized various domains, with education standing prominently affected [1]. Technological tools, once as simple as projectors in classrooms [2], have now transformed into virtual classrooms conducted via video conferences, especially accentuated by the global impact of the COVID-19 pandemic. Yet, the shift to virtual learning comes with its own set of limitations, particularly in fostering interactive student-teacher and peer interactions.

Amid the ongoing quest for optimal technological solutions to enhance education, Virtual Reality (VR) and Augmented Reality (AR) have emerged as promising contenders [3], [4]. Virtual Reality enables users to immerse themselves in computer-simulated environments, as seen in applications like Google Earth's VR implementation. On the other hand, Augmented Reality seamlessly integrates digital elements into the real world, exemplified by social media filters and popular games like Pokémon Go. Together, VR and AR converge to form what is colloquially known as the "Metaverse", a virtual realm that melds the tangible and digital universes.

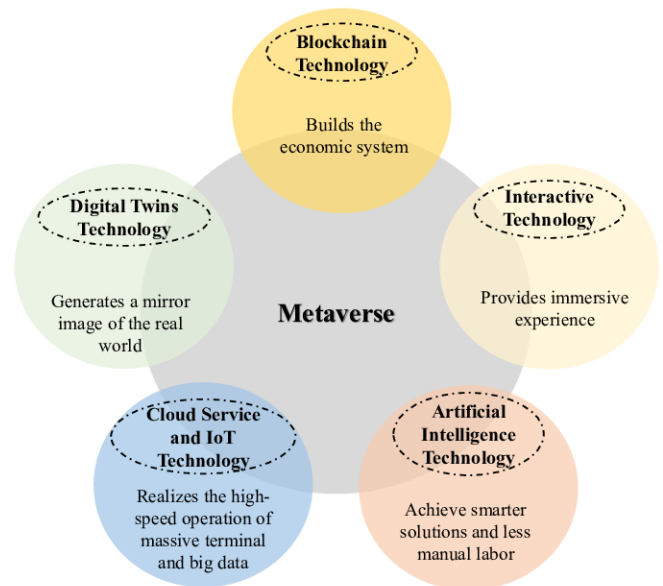


Figure 1: Metaverse-related technologies and their impact on the Metaverse[8]

In addition, the virtual classroom has broader applications in studying human behavior, skill training, and gaming. Neuropsychology benefits greatly from immersive virtual environments. The virtual classroom is revolutionizing education with its impact on accessibility, flexibility, and the learning experience.

We examine the difficulties and opportunities for social good associated with the metaverse in this study as we delve into

its complex domain. We look at how digital technologies might improve educational experiences and meet social requirements, especially in the setting of virtual classrooms. As part of our methodology, a metaverse classroom is meticulously built using state-of-the-art technologies and techniques to guarantee realism and usefulness. Our study strives to unearth insights that pave the path for revolutionary improvements in education and beyond as we negotiate the complexity of the metaverse landscape.

II. CHALLENGES IN THE EDUCATIONAL METAVERSE

A. Challenges from Regulators in the Education Metaverse

The government and other regulatory bodies have not given the education metaverse the attention it deserves, which is one of its main challenges. Through resource sharing and real-time interactive platforms, this virtual learning environment offers enormous potential to educate more people. But because the regulators weren't involved from the beginning, there aren't any clear guidelines or directives. On paper, the government's plans don't offer enough support for this new approach to education, which incorporates socializing, trading, and creative activities in addition to traditional classroom instruction. Encouraging the regulators to join the education metaverse is essential to its growth and development [9].

B. Challenges Faced by Designers in the Education Metaverse

The education metaverse is a dynamic and easily navigable digital learning environment that you can utilize at any time and from any location thanks in large part to the work of metaverse designers. Currently, the technology mostly depends on AR/VR/MR devices, however there is a small issue. The learning experience isn't as immersive as we'd like it to be because of these devices' limitations. It's like having a great gadget that's not quite ready for prime time. The metaverse's required gadgets are also somewhat expensive and difficult to transport. So, even if you have this amazing learning space, it's not always practical to use it. Furthermore, designers still haven't quite figured out how to best organize the many components of the metaverse and determine what each one should be used for in terms of teaching. Therefore, more work needs to be done to ensure that the education metaverse is truly excellent.

C. Challenges Faced by Users in the Education Metaverse

When users step into the education metaverse, they rely on a network of shared resources and strong social connections to enhance their learning. But here's the catch – the ethical rules of this virtual world haven't been set up properly yet. This means users might face similar problems as they would in the real world. Because the metaverse is a big collection of social ties, users might be tempted to explore using less trustworthy resources and tricks.

III. METAVERSE FOR SOCIAL GOOD

Even though the metaverse is essentially a virtual world centered around human interaction, it has a notable positive influence on the real world. This impact is particularly evident in areas such as accessibility, diversity, equality, and humanity. In the following section, we highlight some notable applications that showcase how the metaverse contributes to social good [7].

A. Metaverse Applications for Enhanced Accessibility

In today's globalized world, communication and collaboration between countries have become more frequent. However, the challenge of geographical distance remains a significant obstacle, leading to increased costs in various processes. The COVID-19 pandemic further exacerbated this issue, causing the suspension of many events due to preventive measures.

Enter the metaverse, a solution that enhances accessibility to meet diverse social needs. For instance, numerous events have seamlessly transitioned to virtual formats, thanks to the metaverse. A notable example is UC Berkeley hosting its graduation ceremony on *Minecraft* in 2020. Additionally, platforms like *Fortnite* host a plethora of virtual events, including concerts such as the one featuring Travis Scott. These instances underscore how the metaverse has seamlessly integrated into our daily lives, offering a cost-effective and secure means to fulfill our social needs.

B. Exploring Metaverse for Inclusive Experiences

The limitations of the physical world, including factors like geography and language, make it challenging to integrate diverse elements in one place to cater to the needs of different individuals. Enter the metaverse, offering an expansive virtual realm with seamless scene transformations that can effectively achieve diversity. The metaverse provides a platform for a myriad of intriguing scenarios, breaking free from physical constraints. For instance, *Animal Crossing* organized a presidential campaign for Joe Biden, showcasing the diverse possibilities within the metaverse. Similarly, students at Stanford University exhibited their posters in *Second Life*. These examples, however, only scratch the surface as the metaverse hosts a plethora of activities spanning education, shopping, political campaigns, artwork, pets, haunted houses, and more. Consequently, the metaverse significantly fulfills the diversity requirements of our physical society.

C. Digital Twins

Besides metaverse users, even things in our everyday world can also connect and interact with the virtual realm, appearing as digital twins in this digital space. Imagine them as identical virtual copies that mimic real-world objects. How does this work? Well, devices in the real world have their information collected through widespread sensing technologies. These technologies keep the virtual copies, or digital twins, updated to reflect the current status of their real counterparts. This

connection between the physical and virtual worlds opens up exciting possibilities. It's like having a parallel version of the real world in the metaverse, allowing for a seamless exchange of information and actions between our physical surroundings and the virtual space.

IV. METHODOLOGY

This methodology presents a comprehensive guide to developing an interactive and immersive metaverse classroom using Blender, a powerful 3D modeling software. The process commences with meticulous planning of the classroom layout, encompassing crucial elements like avatars, furniture, walls, doors, windows, whiteboard, roof, and lighting. Designing the core structure in Blender involves creating walls, doors, windows, and roofs, ensuring accurate proportions and a realistic environment.

virtual classroom. Thorough testing and iterative refinement phases ensure optimal functionality and visual quality, promoting an immersive and rewarding learning experience. Upon successful completion, the virtual classroom project is saved in Blender and exported to incompatible formats, making it readily accessible and adaptable for various virtual reality platforms and Metaverse applications.

A. 3D Modeling with Blender

The very first step in our workflow, shown in Figure 2, is creating 3D assets for our metaverse. Before diving into Blender, we carefully planned the layout and design of the virtual classroom. Considering the essential elements, such as avatars, chairs, tables, walls, whiteboards, windows, doors, roofs, and lighting. We envisioned a modern and interactive learning space to ensure an engaging and immersive experience for students.

1) *Designing Classroom:* With the design plan in mind, we opened Blender, the powerful 3D modeling software, created a new project, and set the appropriate dimensions for the virtual classroom scene. Using Blender's versatile tools, creating the core structure of the virtual classroom. Modeling the walls, doors, windows, and roof to give the classroom its basic form. We ensured that the proportions and scale were accurate to provide a realistic environment.

2) *Designing Desks and Chairs:* We focused on designing the furniture and props for the virtual classroom. Using Blender's modeling tools, we crafted chairs, tables, and a whiteboard that fit seamlessly into the classroom setting. Attention to detail, such as textures and colors, to make the objects visually appealing and true to real-life counterparts.

B. Designing Avatars

MetaHuman Creator is a free, cloud-streamed tool you can use to create your digital humans in an intuitive, easy-to-learn environment. Using MetaHuman Creator, you can customize your digital avatar's hairstyle, facial features, height, body proportions, and more [5]. In the Metaverse classroom, integrating avatars using MetaHumans enhances interactivity and engagement. These lifelike digital representations of users allow for real-time communication and collaboration. Avatars foster active participation in discussions, role-playing, and group activities, creating a personalized and immersive learning environment. With that in mind, we created avatars for teachers and students.

C. Gaming Engine Integration

A gaming engine like Unity or Unreal Engine would be the perfect platform to use to bring our Meta-Space concept to life, merging our 3D classroom elements like chairs, desks, and avatars into a unified virtual setting. Using a gaming engine's capabilities enables dynamic interactions, smooth asset integration, and the deployment of several functionalities. We can give avatars the ability to move around the virtual classroom, interact with furniture like desks and

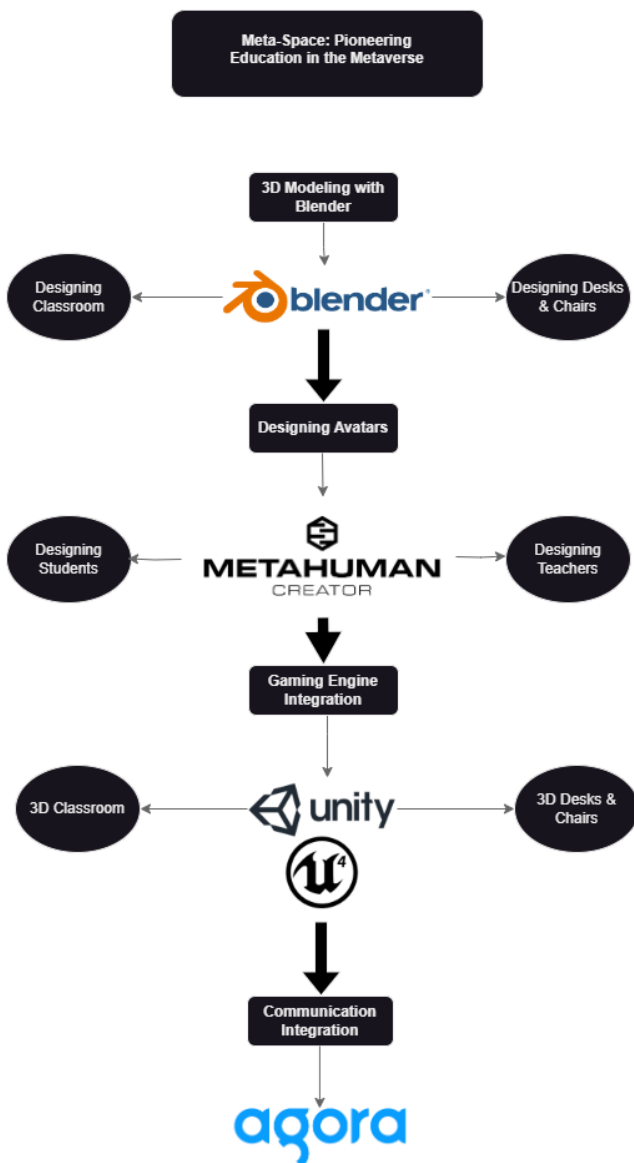


Figure 2: Meta-Space Flowchart

The methodology emphasizes applying appropriate texturing and lighting to achieve an authentic ambiance within the

chairs, and even participate in group activities by using scripting and programming within the gaming engine. These engines also provide capabilities for improving visual fidelity and performance, guaranteeing an engaging and immersive experience in our Meta-Space environment.

D. Communication Integration

Agora SDK is an all-inclusive real-time communication solution that gives developers the infrastructure and resources they need to incorporate message, audio, video, and live streaming features into their projects. The Agora SDK's capability to provide 3D spatial audio is one of its most notable features. By mimicking how sound behaves in the real world, spatial audio improves the immersive experience of virtual worlds by enabling users to detect audio sources from various angles and distances. Agora uses spatial audio techniques and sophisticated audio algorithms in its SDK to do this. Agora SDK generates realistic audio environments that improve users' sense of presence and immersion when interacting with 3D spaces or virtual worlds. These environments are achieved by precisely placing audio sources within a virtual space and adjusting properties like volume, directionality, and distance attenuation. Applications where spatial awareness and realistic audio interactions are crucial for an engaging user experience, like virtual events, online gaming, remote collaboration, and virtual classrooms, will greatly benefit from these capabilities. We currently are working to implement Agora SDK in our project.

IV. RESULTS

The results of the experimental process yielded insightful findings across various dimensions, shedding light on the efficacy and potential of the implemented Metaverse classroom.

A. Classroom in Blender

The classroom is designed with a capacity of 10-15 students. This number ensures that there isn't too much traffic on the server. The classroom has enough room for the teacher to move around during lectures and for students to not get too cramped in a space.

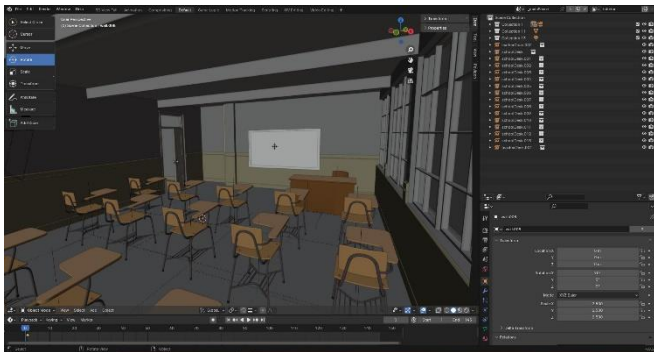


Figure 3: Classroom Model in Blender.

B. Furniture and Props

For the furniture, we designed simple chairs and desks for the classroom to give a real-life-like feeling. The props include

lights in your classroom and a whiteboard. The users can freely interact with the furniture and lights, bringing the virtual classroom to life. Figure 3 and Figure 4 show the whole classroom and the teacher's desk.

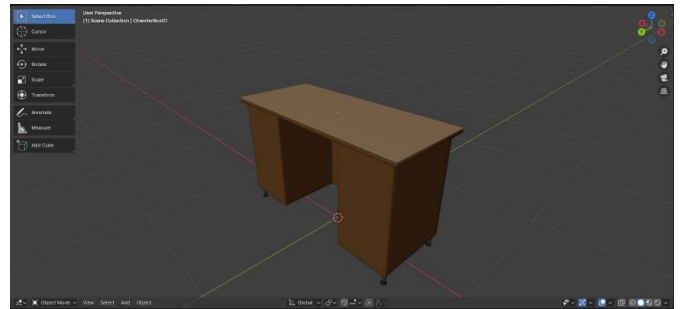


Figure 4: Modeling of Desk in Blender

C. Avatars for the Classroom

In the Metaverse classroom, we are currently working on integrating avatars using MetaHumans, which enhances interactivity and engagement. These lifelike digital representations of users allow for real-time communication and collaboration. Avatars foster active participation in discussions, role-playing, and group activities, creating a personalized and immersive learning environment. In Figure 4, we tried to create our professor digitally.



Figure 4: Sir Yasir Avatar in MetaHuman

V. CONCLUSION

To sum up, the education metaverse has the amazing potential to completely transform the way we learn by removing obstacles based on geography and enabling access to education for everyone. We do, however, have certain obstacles. To control the expansion of the metaverse, regulators including the government, need to be more vigilant and establish explicit guidelines. To use accessible technology to create an immersive and useful learning experience, Designers have their work cut out, too, as they strive to make the learning experience immersive and practical with accessible technology. While social networks and shared resources are beneficial, users must exercise caution to avoid taking unnecessary risks in this rapidly changing digital environment. To guarantee that everyone

involved has a positive and meaningful learning experience, teamwork, careful rules, and ongoing technological advancements are necessary to realize the full potential of the education metaverse.

REFERENCES

- [1] R. Raja and P. C. Nagasubramani, "Impact of modern technology in education," *Journal of Applied and Advanced Research*, vol. 3, no. 1, pp. 33–35, 2018, [Online]. Available: https://www.academia.edu/download/63887921/Impact_of_modern_technology_in_education20200710-27957-jsmaeg.pdf
- [2] N. Amanda, "A Systematic Literature Review: Learning with Visual by The Help of Augmented Reality Helps Students Learn Better," *Procedia Computer Science*, vol. 179, pp. 144–152, 2021, doi: 10.1016/j.procs.2020.12.019.
- [3] C. Akshay, D. Visagaperumal, and V. Chandu, "Metaverse future of internet," *International Journal of Research Publication and Reviews*, vol. 2, no. 8, pp. 386–392, 2021.
- [4] Y. Sun, M. G.-Ep. S. in B. Environment, and undefined 2021, "Potentials of Virtual Social Spaces for Construction Education," yahootechpulse.easychair.org, [Online]. Available: <https://yahootechpulse.easychair.org/publications/download/cK2t>.
- [5] <https://dev.epicgames.com/documentation/en-us/metahuman/metahuman-creator>
- [6] J.-H. Won, Y. Choi, and Y. S. Kim, "A Metaverse Platform for Engineering Education: Case of South Korea".
- [7] Haihan Duan, Jiaye Li, Sizheng Fan, Zhonghao Lin, Xiao Wu, and Wei Cai. 2021. Metaverse for Social Good: A University Campus Prototype. In Proceedings of the 29th ACM International Conference on Multimedia (MM '21). Association for Computing Machinery, New York, NY, USA, 153–161. <https://doi.org/10.1145/3474085.3479238>
- [8] Metaverse Security and Privacy: An Overview - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Metaverse-related-technologies-and-their-impact-on-the-Metaverse_fig1_367455946 [accessed 4 Mar, 2024]
- [9] T. Hao and H. Lailin, "Educational Metaverse Dilemmas and Solutions: a stakeholder-based perspective," 2022 12th International Conference on Information Technology in Medicine and Education (ITME), Xiamen, China, 2022, pp. 714–718, doi: 10.1109/ITME56794.2022.00150.

Use of Artificial Intelligence in Ethereum Forecasting: The Deep Learning Models RNN And CNN With Ensemble Averaging Technique

Fozia Zeeshan

School of Landscape Architecture,
Lincoln university New Zealand

Yoobee College of Creative

Innovation

Christchurch, New Zealand

Fozia.zeeshan@lincolnuni.ac.nz

Fozia.zeeshan@yoobeecolleges.co

[m](#)

Narayan Nepal

Head of Faculty, Yoobee College of
Creative Innovation

Christchurch, New Zealand

Narayan.Nepal@yoobeecolleges.co

[m](#)

Mohammad Norouzfard

Auckland Bioengineering Institute,
University of Auckland.

Yoobee College of Creative

Innovation.

Auckland, New Zealand

m.norouzfard@auckland.ac.nz

Abstract— In the fast-evolving cryptocurrency market, accurately predicting Ethereum prices is crucial for investors, traders, and financial analysts. Traditional machine learning (ML) models often struggle to capture the market's complex dynamics due to their inability to consider all influencing factors. This study introduces an advanced ensemble machine learning approach to enhance Ethereum price prediction accuracy. By combining the strengths of Bi-directional Long Short-Term Memory (Bi-LSTM) and Convolutional Neural Network (CNN) models, our ensemble averaging method compensates for individual model weaknesses, improving forecast reliability and precision. Results show that our ensemble model offers significant advantages, particularly in terms of generalizability and resistance to overfitting with LSTM and CNN models and this technique is offering a more effective tool for navigating cryptocurrency market complexities. This research highlights the importance of ensemble learning in financial forecasting and provides a practical framework for developing superior predictive models."

Moreover, This study explores an advanced ensemble machine learning approach to enhance Ethereum price predictions, combining the strengths of Bi-directional Long Short-Term Memory (Bi-LSTM) and Convolutional Neural Network (CNN) models. While Bi-LSTM individually exhibits slightly higher performance in our tests, the ensemble method demonstrates enhanced stability and reliability, making it a valuable tool for navigating the unpredictable dynamics of the cryptocurrency market.

We found that Bi-LSTM is good on its own, but the balanced approach of the ensemble model is far better, especially when it comes to generalizability and overfitting resistance. Insights into creating flexible and trustworthy prediction models are provided by this study, which highlights the possibilities of ensemble learning in financial forecasting.

Keywords— LSTM(Long short term memory) , CNN(Convolutional Neural Network), RNN(Recurrent Neural network), Ensemble learning, Deep learning

I. INTRODUCTION

This paper investigates the causes for the effectiveness of the most well-known and accurate forecasting techniques. This study examines the impact of incorporating all relevant columns, such as 'Open', 'Close', 'High', 'Low', and Volume, on the accuracy of forecasting models.

This research contributes to the field of cryptocurrency forecasting through the development of two distinct deep learning models, CNN and the variant of RNN the Bi-LSTM, and a state-of-the-art ensemble model to evaluate the accuracy of all models and the performance of individual and ensemble models. This research casting light on the most precise prediction methods by evaluating the accuracy level of individual model.

Researchers used diverse statistical, machine learning, and deep learning techniques to forecast various cryptocurrencies, but it is unclear which of these approaches is preferable. Because most of the studies are done over relatively short time periods and concentrate on forecasting the price of cryptocurrencies across a wide range of time intervals, the research is scattered and lacks generalization. In addition, the accuracy and evaluation of the performing models are grossly neglected.

Also, the models are very complicated, which makes it hard to use them in the real world because of the high costs of application, training, and forecasts. Lastly, due to the diverse datasets, pre-processing strategies, and experimental methodologies, the comparisons between the approaches are inconsistent, the experiments are difficult to reproduce, and their results are therefore unreliable. The primary objective of this paper is to overcome these limitations and raised light on the effectiveness of the most prominent approaches proposed to date in the literature for the crypto price prediction task. As a significant contribution, we develop a framework for comparing the accuracy of each deep learning model and its performance relative to ensemble learning models.

In this paper, we propose various Deep Learning models and evaluate their efficacy in relation to Ethereum price forecasting.

Additionally, we provide a novel ensemble learning approach that utilizes two deep learning architectures, namely Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN), for the purpose of predicting Ethereum values. The findings obtained from the LSTM and CNN models are used to implement an ensemble averaging strategy for result aggregation. Based on existing scholarly understanding, the main emphasis in Ethereum forecasting study is to matters of safety and confidentiality. Previous research in the field of price forecasting has mostly used data such as price, volume, and textual content. These studies have primarily concentrated on datasets characterized by modest swings and short-term time periods.

In our research, however, we use daily data from the past five years to predict the next day's price that means we worked on long term time period.

II. LITERATURE REVIEW

The issue of forecasting future values based on historical data poses a significant challenge. This phenomenon is widespread in several practical contexts investigated by scholars, including but not limited to finance, climate prediction, and energy production. Financial time series forecasting is a very intricate issue that has garnered significant scholarly attention over the course of many decades. Nevertheless, the predictive capacity of conventional statistical models is constrained when it comes to forecasting financial time series data, mostly owing to the intrinsic volatility that characterizes such data. Predicting financial time series, especially stock prices, is a challenging task due to the presence of heavy-tailed distributions. Currently, the unsolved issue of stock market return prediction persists, with disputed conclusions[2]. The researchers highlighted the phenomenon of this type of predictions and their controversial results.[3]. The recent rise in the value of cryptocurrencies, particularly Ethereum, has sparked a corresponding surge in the desire to forecast their future pricing[4]. Similar to the stock market, the precision of Ethereum price predictions will lead to enhanced financial gains for investors. In order to effectively capture complex nonlinear interactions among data variables, the analysis of time series pricing data requires the use of advanced techniques, such as deep learning models [5]. As a result, the anticipation of investment markets has emerged as a prominent subject of study within the domain of machine learning.

In addition, a wide array of scientific and financial approaches are used to forecast the future price fluctuations of cryptocurrencies[6]. ARIMA is a notable illustration of a financial and statistical methodology [7]. This model is often used by researchers for the purpose of predicting Ethereum values [8,9]. Additional models have been used in the field, including generalized autoregressive conditional heteroscedasticity (GARCH) models for predicting volatility in cryptocurrencies [10].

The GARCH model is considered volatility forecasting model in many researchers' point of view[11] as well as diffusion processes in probabilistic forecasting of cryptocurrencies also used for volatility forecasting [12]. Another cohort of researchers use machine learning (ML) methods such as stochastic gradient boosting machines [13], linear regression, random forest, support vector machines, and k-nearest neighbours [14]. To enhance the precision of price forecasts, these methodologies use past data to ascertain the most important aspects that will shape future cryptocurrency values. A third body of literature use deep learning (DL) models to predict the value of cryptocurrencies, building upon their recent achievements in quantitative finance [15]. The values of Bitcoin, Ethereum, and Litecoin have been forecasted via the use of several neural network models, including recurrent neural networks (RNN) such as gated recurrent unit (GRU) and long short-term memory (LSTM), temporal convolutional networks (TCN), and hybrid designs. Deep learning (DL) methods are considered to be useful for time series forecasting due to their ability to handle noise, their inherent capability to handle data sequences, and their capacity to learn nonlinear temporal relationships from these sequences [16].

III. MODELING AND APPROACHES

This section describes the procedures followed during the preprocessing and modeling phases of the investigation. The performance of CNN, Bi-LSTM, and Ensemble models is then assessed. The outcomes of each model's forecasting and prediction graphs are explained. In conclusion, our examination of the study's results is comprehensive and exhaustive.

The objective of this study is to forecast the Ethereum price for the next day using CNN, Bi-LSTM, and ensemble learning techniques. The following research methods employ a state-of-the-art technique that will be a game changer in the field of Cryptocurrency price prediction.

1. The historical data encompasses a diverse range of attributes, including the opening and closing prices. The usual methodology used in trade market forecasting entails the examination and analysis of these aforementioned features. The highest and lowest prices refer to the uppermost and lowermost values at which a Ethereum was exchanged during a certain time frame. The closing price is often defined as the last price at which a transaction using Ethereum takes place within a certain timeframe. The open price refers to the specific price at which a cryptocurrency commences trading at the commencement of a designated time period. We employed historical Ethereum daily pricing data spanning from 2017 to September 2023.
2. To separate the dataset into training and testing sets, the `prepare_data` function is created. It also prepares the data for input into all models.
3. Development of Bi-LSTM, CNN, and ensemble Averaging models for deep learning
4. Training consisting of Bi-LSTM, CNN, and Ensemble Averaging models.

5. Models Forecasting
6. Models' evaluation.
7. Comparing the results obtained from Bi-LSTM, CNN, and ensemble models for Ethereum forecasting using various evaluation matrices, including RMSE, MAE, MAP, and Accuracy.

The URL to our proposed models on github is provided below.

[Ensemble-learning-model/Ensemble_deep_learning_model.docx at main · Fozeeshan/Ensemble-learning-model \(github.com\)](https://github.com/Fozeeshan/Ensemble-learning-model)

A. Historical Data

Ethereum historical price data is taken from from CoinMarketCap from Nov 2017 till sep 2023 <https://coinmarketcap.com/>

Table-I explains our dataset. It contains Date, open, high, low, close and volume columns. We are incorporating all columns to determine the influence of these column values on price forecasting. Close column is our targeted column, and the remaining columns are our input columns; therefore, the predicted next day value is forecasted through the inclusion of all other columns of the data set.

We applied following methodology on data for predicting Cryptocurrency prices.

TABLE-I SHOWS COMPLETE DATA COLUMNS OF DATASET.

Parameter	Description	Data Type
Date	Date through that we can predict our Ethereum	Date
Open	Daily prices from which Ethereum values starts	Number
High	Daily high price of the Ethereum	Number
Low	Daily low price of the Ethereum	Number
Close	These are basically our targeted value. It is daily close price of the Ethereum	Number
Volume	Total volume of Ethereum at that close price.	Number

B. Proposed Structure of our Ensemble Averaging deep learning model.

In deep learning and machine learning in general, simple averaging is a foundational ensemble technique. In straightforward averaging, the predictions of multiple neural networks (or models) are averaged together. This technique is simple and effective for enhancing the accuracy of predictions. Regression and classification problems are both amenable to simple averaging.

By merging the results of many distinct models, a simple averaging ensemble deep learning model is a method for improving forecast accuracy. The average of the predictions made by each model in the ensemble is calculated using this technique.

The fundamental equation for simple averaging using two models (M1 and M2) is as follows:

$$(\text{Prediction from M1} + \text{Prediction from M2}) / 2 = \text{Final Averaged Prediction}$$

This research applied an advanced ensemble Averaging deep learning model that included LSTM, a kind of recurrent neural network (RNN) often used for time series prediction, in conjunction with a convolutional neural network (CNN). The convolutional neural network (CNN) is a commonly employed network design within the field of deep learning. It is designed to learn directly from input and has shown high efficacy in tasks such as time series predictions and ensemble learning, where numerous individual models are combined to obtain improved generalization performance. Consequently, throughout the process of our study, we used an ensemble approach by combining our Bi-Long Short-Term Memory (Bi-LSTM) and Convolutional Neural Network (CNN) models in order to enhance the accuracy of our outcomes.

C. The structure of Bi-LSTM.

Bidirectional -Long Short-Term Memory model is constructed and trained. Bidirectional LSTMs have both forwards and backwards passes and are effective at sequence prediction tasks. To prevent overfitting, the model architecture consists of three LSTM layers with variable numbers of units and dropout layers.

A loss function and the Adam optimizer with a specified learning rate are used to compile the model.

Early stopping is used to monitor the validation loss, allowing the model to end training when the validation loss no longer improves.

D. The Structure of CNN.

Convolutional Neural Network (CNN) model is constructed. The architecture of the model consists of a 1D convolutional layer, max-pooling, flattening, and dense connected layers.

Similar to the LSTM model, the CNN model is compiled using a loss function and the Adam optimizer. Early stopping is also implemented.

E. Predictions with Bi-LSTM and CNN

Using both Bi-LSTM and CNN models, predictions are made on the test data following training. The resultant predictions are presented as arrays.

I used model_lstm and model_cnn to make predictions on the X_test data in our programme. The arrays y_pred_lstm and y_pred_cnn contain the predictions for the Bi-LSTM and CNN models, respectively.

F. Ensemble Averaging to Combine Predictions

Using ensemble averaging, the program combines the predictions of the Bi-LSTM and CNN models. This basic

method creates a new prediction by averaging the predictions of the two models.

In our program, y_{pred_lstm} and y_{pred_cnn} represent the predictions of the Bi-LSTM and CNN models, respectively. The ensemble prediction, $y_{pred_ensemble}$, is created by averaging y_{pred_lstm} and y_{pred_cnn} element-by-element. This ensemble forecast represents an aggregate estimate of the closing prices.

Consequently, $y_{pred_ensemble}$ comprises the predictions derived from the ensemble model created by aggregating the predictions of the individual Bi-LSTM and CNN models.

The concept behind ensemble averaging is that it can increase the accuracy of predictions by minimizing the impact of individual model errors.

G. Evaluation Metrics:

In this model, two important evaluation metrics are used to assess the accuracy of predictions:

- The Mean Absolute Error (MAE) is a metric used to quantify the average absolute deviations between the observed Ethereum prices and the corresponding anticipated values. This provides a glimpse into the extent of the model's inaccuracies. In the given model's context, a reduced Mean Absolute Error (MAE) signifies a higher level of accuracy in the price predictions made by the model.
- Root Mean Squared Error (RMSE): This metric quantifies the square root of the mean of the squared errors in predictions. The method assigns more significance to substantial errors, hence exhibiting sensitivity towards outliers. Root Mean Square Error (RMSE) is a valuable statistic for comprehending the dispersion of forecast inaccuracies. Like Mean Absolute Error (MAE), a lower Root Mean Square Error (RMSE) suggests that the model's predictions are more precise.

These evaluation metrics are essential for determining the precision and dependability of the ensemble model's Ethereum price forecasts.

In our research, each of the Bi-LSTM, CNN, and ensemble models' evaluation metrics are also generated. Included among the metrics are Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and a "Accuracy" score.

We provided three visualization graphs for understanding the efficacy of the models. It depicts actual close prices versus predicted close prices for the Bi-LSTM, CNN, and ensemble models

IV. RESULTS AND DISCUSSIONS

(Bi-LSTM) Model Performance

Based on our analysis, the Bi-LSTM model, which takes advantage of the sequential nature of the data, proved to be the most effective.

Crucial Performance Indicators: The model demonstrated exceptional performance, attaining a Mean Absolute Error (MAE) of 0.0061, a Root Mean Squared Error (RMSE) of 0.0091, and an accuracy of 98.47%. The performance of

this model highlights the remarkable ability of the Bi-LSTM to identify and forecast the intricate dynamics associated with the price fluctuations of Ethereum.

CNN Model Performance:

In our research, the CNN model, which is renowned for its ability to recognize spatial patterns in data, also demonstrated impressive performance.

Crucial Performance Indicators: With an MAE of around 0.010, an RMSE of around 0.012, and an accuracy of nearly 97.40%, the CNN model achieved these results. Although the CNN model offers significant insights into the price behaviour of Ethereum, its overall accuracy is marginally inferior to that of the Bi-LSTM model.

Ensemble Model Performance:

By employing an aggregating technique, our ensemble model amalgamated the predictive prowess of the Bi-LSTM and CNN models, resulting in improved forecasting capabilities. Crucial Performance Indicators: The accuracy of 97.98%, MAE of 0.008, RMSE of 0.012, and MAE of 0.008 produced by the ensemble method demonstrate an interaction between the sequential and spatial data processing capabilities of the individual models. While demonstrating an enhancement compared to the CNN model, the ensemble model falls short in comparison to the Bi-LSTM model.

Visual Verification of Predicted Models:

Bi-LSTM prediction: The discrepancy between the Bi-LSTM model's predictions and the actual Ethereum prices (as illustrated in Figure 1) underscores this model's superior accuracy and capacity to closely monitor price trends.

CNN Predictions: The efficacy of the CNN model is illustrated in Fig. 2, which compares its predictions to actual prices. However, it is worth noting that the CNN model exhibits a slightly lower precision in comparison to the Bi-LSTM model.

Ensemble Model Predictions: The predictions generated by the ensemble model are presented in Figure 3. This visual representation provides strong evidence of the model's ability to accurately forecast Ethereum price fluctuations, although it does not outperform the performance of the Bi-LSTM model.

TABLE-II RESULTS OF THE STUDY

Matric	Bi-LSTM	CNN	Ensemble
MAE	0.0061	0.010	0.008
RMSE	0.0091	0.015	0.012
ACCURACY	98.47%	97.40%	97.98%

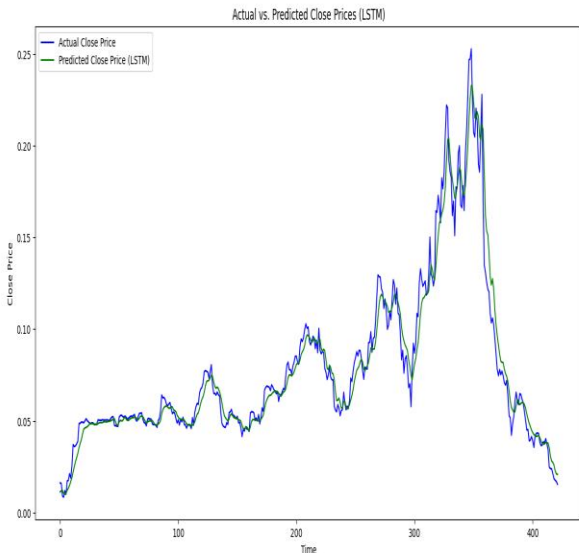


Fig. 1. Ethereum prediction with Bi-LSTM actual vs predicted

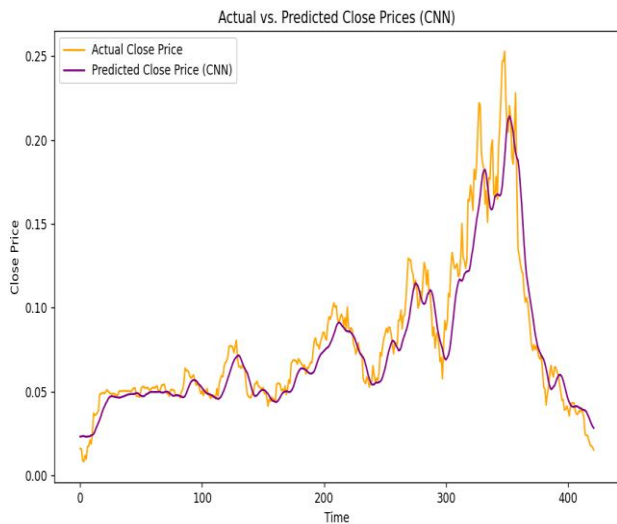


Fig. 2. Ethereum prediction with CNN actual vs predicted.

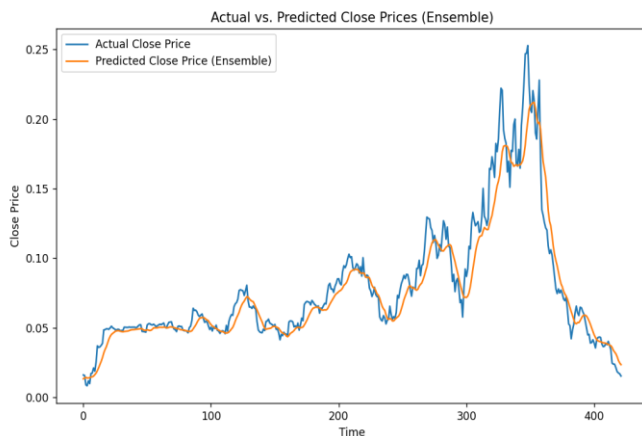


Fig. 3. Ethereum prediction with Ensemble learning model actual vs predicted.

V. CONCLUSION

Ethereum price forecasting is a complicated and intriguing undertaking for investors, traders, and financial experts in the ever-changing cryptocurrency market. This research endeavour aimed at simplifying the process of predicting Ethereum prices. We strived to find the best method for forecasting Ethereum prices by thoroughly analyzing several deep learning models, such as Bi directional Long Short-Term Memory (Bi-LSTM), Convolutional Neural Network (CNN), and a novel ensemble technique.

Our results show how the Bi-LSTM model outperformed the competition; it used sequential data analysis to predict Ethereum prices with unprecedented precision. With an astounding accuracy of about 98.47%, the model proved itself to be a formidable instrument for predicting cryptocurrency prices on its own.

The ensemble model, which integrates Bi-LSTM and CNN models, as well as the CNN model on its own, provided useful information about future price changes; nevertheless, they were unable to outperform the Bi-LSTM model in terms of prediction accuracy but it is showing that at a stage where CNN has stopped improving the results, by ensembling we achieved the better results than individual CNN. The significance of choosing a model that is tailored to the data and the forecasting objective is highlighted by this discovery.

Finally, our analysis proves that the Bi-LSTM model can accurately anticipate Ethereum prices, which is a huge step forward for cryptocurrency forecasting. It further highlights the possible advantages and disadvantages of using ensemble techniques in this setting. In light of the ever-changing cryptocurrency market, this study lays the groundwork for further research into improving prediction models by some more ensembling techniques and provide useful information for traders trying to make sense of Ethereum's intricate trading platform.

REFERENCES

- [1]. N.I. Sapankevych and R. Sankar, "Time Series Prediction using Support Vector Machines: A Survey," *IEEE Computational Intelligence Magazine*, 4(2):23–38, 2009.
- [2]. M. Saad, J. Choi, D. Nyang, J. Kim, and A. Mohaisen, "Towards Characterizing Blockchain-based Cryptocurrencies for Highly-Accurate Predictions," *IEEE Systems Journal*, 14(1):321–332, March 2020.
- [3]. J. Kim and J. Kim, "Demo: Light-Weight Programming Language for Blockchain," in *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Seoul, Korea, June 2019.
- [4]. M. Saad, L. Njilla, C.A. Kamhoua, J. Kim, D. Nyang, and A. Mohaisen, "Mempool Optimization for Defending Against DDoS Attacks in PoWbased Blockchain Systems," in *Proc. IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, Seoul, Korea, May 2019.
- [5]. A. Einstein, B. Podolsky, and N. Rosen, "Can Quantum-Mechanical Description of Physical Reality be Considered Complete?," *Phys. Rev.*, 47:777–780, 2018.
- [5] Umar, M., Su, C. W., Rizvi, S. K. A., & Shao, X. F. (2021). Ethereum: A safe haven asset and a winner amid political and economic uncertainties in the US?. *Technological Forecasting and Social Change*, 167, 120680
- [6]. Box, G.E.P.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Holden-Day: San Francisco, CA, USA, 1976.
- [7]. Wirawan, I.M.; Widiyaningtyas, T.; Hasan, M.M. Short Term Prediction on Ethereum Price Using ARIMA Method. In *Proceedings of the 2019 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, Indonesia, 21–22 September 2019; pp. 260–265. [CrossRef]
- [8]. Yamak, P.T.; Yujian, L.; Gadosey, P.K. A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, Sanya, China, 20–22 December 2019; Association for Computing Machinery: New York, NY, USA, 2020; pp. 49–55. [CrossRef]
- [9]. Roy, S.; Nanjiba, S.; Chakrabarty, A. Ethereum Price Forecasting Using Time Series Analysis. In *Proceedings of the 2018 21st International Conference of Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 21–23 December 2018; pp. 1–5. [CrossRef]
- [10]. Walther, T.; Klein, T.; Bouri, E. Exogenous drivers of Ethereum and Cryptocurrency volatility – A mixed data sampling approach to forecasting. *J. Int. Financ. Mark. Inst. Money* 2019, 63, 101133. [CrossRef]
- [11]. Maciel, L. Cryptocurrencies value-at-risk and expected shortfall: Do regime-switching volatility models improve forecasting? *Int. J. Financ. Econ.* 2021, 26, 4840–4855. [CrossRef]
- [12]. Mba, J.C.; Mwambi, S.M.; Pindza, E. A Monte Carlo Approach to Ethereum Price Prediction with Fractional Ornstein–Uhlenbeck Lévy Process. *Forecasting* 2022, 4, 409–419. [CrossRef]
- [13]. Derbentsev, V.; Babenko, V.; Khrustalev, K.; Obruch, H.; Khrustalova, S. Comparative performance of machine learning ensemble algorithms for forecasting cryptocurrency prices. *Int. J. Eng.* 2021, 34, 140–148.
- [14]. Chevallier, J.; Guégan, D.; Goutte, S. Is It Possible to Forecast the Price of Ethereum? *Forecasting* 2021, 3, 377–420. [CrossRef]
- [15]. Khedr, A.M.; Arif, I.; El-Bannany, M.; Alhashmi, S.M.; ; Sreedharan, M. Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. *Intell. Syst. Account. Financ. Manag.* 2021, 28, 3–34. [CrossRef]
- [16]. Brownlee, J. *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python; Machine Learning Mastery*: San Juan, PR, USA, 2018

Model-Based vs Model Free Deep Reinforcement Learning Models for Cancer Treatment: A Critical Analysis with a Solution DRL Model

Madeha Arif

Department of Computer Software
Engineering
College of Electrical and Mechanical
Engineering, NUST
Rawalpindi, Pakistan
madeha.arif15@ce.ceme.edu.pk

Usman Qamar

Department of Computer Software
Engineering
College of Electrical and
Mechanical Engineering, NUST
Rawalpindi, Pakistan
usmanq@ce.ceme.edu.pk

Abstract— In the field of artificial intelligence, deep reinforcement learning (RL) has grown to be one of the most talked-about issues. It has a wide range of applications, including end-to-end control, robotic control, recommendation systems, and systems for natural language communication. In this paper, we have critically reviewed model-based and model-free deep reinforcement models for the treatment of cancer patients and evaluated each model based on some parameters. Based on the evaluation, a critical discussion is carried out highlighting the limitations and drawbacks of all the existing models. The analysis also gives suggestions and marks the key indicators of future needs in this domain. In the end, a solution model is proposed that tries to cover all the shortcomings and addresses the issues encountered in the existing models. The findings indicate that we can get a 94% accuracy rate with three agents, and that increasing the number of agents has no further effect on accuracy.

Keywords: Deep Reinforcement Learning (DRL); Model-based learning; Model-free learning; Deep Learning; Neural Network (NN)

I. INTRODUCTION

Deep learning and large data have enabled dramatic advancements in the study of artificial intelligence. In the area of post-deep learning, interest in investigating new technologies is expanding. Particularly appealing is deep reinforcement learning (RL), which incorporates neural network modeling into conventional RL algorithms. Deep RL specifically determines which action to take to maximize the benefit in the face of a particular situation by solving decision optimization issues. As a result, deep RL analysis and application are receiving a lot of attention from both the academic community and industry. Between deep RL and conventional machine learning, there are significant variations in processing and analysis.

The current, widely accepted machine learning paradigm mostly gathers or creates dataset tags in advance and executes machine learning using static data that is already available. Contrarily, RL is a classic illustration of the closed-loop learning paradigm, which incorporates feedback signals into the learning process using dynamic data and tags. We attempt to give a summary of the state-of-the-art deep RL algorithms in this research. The first section is all about DRL and its basic information. In the next section, there is a brief description of Model-based vs model-free DRL algorithms and then there are past models that are examined and analysed based on certain

parameters followed by a discussion highlighting the shortcomings and benefits of those models. In the end, there is a proposed model that combines all the suggestions and covers all the shortcomings of previous models.

II. DEEP REINFORCEMENT LEARNING (DRL)

A. Model Free DRL

Development done in deep reinforcement learning is still in starting phase and a lot is needed to be done. Most academic researchers focus on the static and deterministic environment where states have been fully observed and are static as well [1]. Thus, the majority of the RL techniques are model-free. Model-free RL makes estimates of the system state, value function, and the reward function of an agent using a large number of samples for action policy optimization that is aimed to achieve more rewards. Due to the least complex implementation and open resources, model-free RL has attracted many more scholars for carrying out further research in this field [3].

B. Model-Based DRL

Things are made easier with known transition dynamics between states and future actions. Such dynamics are called models. Model-based methods include algorithms that learn the transitions to decide which new state (s_{t+1}) will be selected after performing an action a_t in current state s_t . These methods will figure out how to select the actions. In short, these algorithms learn models of system dynamics and then optimal control strategy for choosing such next actions. Model-based RL algorithms are developed from optimal control methods. In comparison with model-free RL methods, the model-based RL algorithms learn a value function or next policy using a data-efficient manner and they do not need to continuously interact with the environment. This may lead to difficulty in model identification and cause an inaccurate description of the real environment. However, it may suffer from the issue of model identification and lead to an inaccurate description of the real environment [4]. If there is a simulator available but the cost of performing the simulations is too much then opt for model-free off-policy algorithms e-g NAF, DDPG [2], and SQL [5]. There are multiple choices for all the policy-based model-free methods in RL that can be extra critic based or Q-Learning-based. In case we don't have any simulator, the main question arises that how long the

acceptable waiting period should be. In this case, if the waiting period is not much prolonged, then the model-based algorithms are a better option e-g probabilistic ensembles with trajectory sampling (PRTS) [6] and guided policy search (GPS) [7]. In another case, model-free off-policy algorithms can be used that make use of some assumptions and can be made more generic and less domain-specific.

III. PAST CONTRIBUTIONS

A. Model-Based DRL Algorithms

Many RL methods have been proposed in the past for solving many problems related to healthcare and specifically cancer treatment. Masoud et al. [8] proposed a method that is a combination of fuzzy sets and reinforcement learning called fuzzy reinforcement learning for controlling the growth of cancer cells. They have used two different drug dosages to reduce the population of cancer cells. The main limitations of the research are that they have used arbitrary parameters selected by the trial-and-error method.

Another research conducted by Shen et al. [9] proposed a single-agent deep reinforcement learning model for weight tuning. They used Epsilon greedy approach and solved the optimization problem by minimizing the sum of doses fed to four critical organs with doze balancing. Their proposed method is high-dose-rate brachytherapy (HDRBT) for cervical cancer.

In their research, Capizzi et al. [10] a hybrid model is proposed that combines a fuzzy system with a neural network for lung cancer nodule detection. The proposed model is only effective for certain types of data and patient samples. They used MATLAB toolbox to simulate their model and x-ray images of lung nodules as a dataset. Results gave 92.56% accuracy.

In another research, a Parameter Tuning Policy Network (PPTN) was trained using the procedure of end-to-end reinforcement learning [11]. They tried this method to improve the image quality of CT scan with this generic framework. The drawback was that this method had to wait for an iterative process to further tune its parameters. This method focused on policy optimization.

Another model uses Markov Decision Process for optimization of lung cancer detection by training a dynamic Bayesian network and then discovers an expert's decision-based reward function through the inverse reinforcement learning method. They unfortunately could not handle the stochastic nature of patient responses and the model is not suitable for lung nodule images taken at random frequencies. They used NLST data and simulated on MATLAB toolbox [12].

A Q-Ranking approach was used to detect cell lines' sensitivity to anti-cancer drugs. This method integrates various predictive algorithms and then chooses a suitable algorithm for a certain application. Batch reinforcement learning is used to identify the ranking policy [13]. The model has a limitation of not being scalable and is not generic. NCI-DREAM7 dataset is used for addressing policy optimization problems.

B. Model Free DRL Algorithms

A value-based single-agent reinforcement learning method with TD and Q Learning was proposed for tumor

localization of lung cancer [14]. The authors categorized lung cancer types and described the characteristics of each one. The most challenging part of the application of this RL method was to define a suitable reward function for updating the Q-value for each performed action.

Another Q-Learning-based approach was used to control the drug dosing during chemotherapy treatment. A scaled error value of reward function based on the count of cancer and normal cells. The authors Padmanabhan et al. [15] applied their model to patients from different age groups and for each case, a different RL agent was developed to control each case. The main limitation of this study was that the proposed model was not generic to be applied to all the cases, but they needed to be specific according to each patient's characteristics.

The model addressed the application of antiangiogenic therapy for the reduction of tumor volume [16]. The volume of tumor is considered as a reward function if the error value is equal to or less than 1. They tested their model on just a single patient record using Silico Simulations. Due to the higher complexity of this model, model-based controllers are impossible to use. Also, the model could not handle the stochastic nature of the patient's dynamics [17].

Huan et al. [18] also proposed a model for controlling automated radiation adaption for lung cancer. A DRL approach with three component-based neural network framework with a Deep Q-network is developed. A limited number of samples were used, and the reward function was customized for each patient. 114 NSCLC patient data was used.

In their research [19], proposed another model for lung cancer nodule detection with a combined framework of fuzzy systems and neural networks. Again, the model is only suitable for certain patient samples and not generic to be adopted for all patients. They used the MATLAB toolbox for carrying out model simulations and optimized the reward function.

Padmanabhan et al. [20] proposed another model for reward function optimization for modeling the optimal drug dosing in cancer treatment. 15 patients were simulated for model application with MATLAB Simulations. They also proposed a model by Padmanabhan et al. [21] using integral reinforcement learning for optimal drug dosing for a provided performance measure. Only 10 patients were simulated using MATLAB simulations. whereas stochastic parameters like nonlinearities, time delays, and nonnegative constraints are not handled with this model.

IV. ANALYSIS

In this section, many past related contributions have been discussed and analyzed based on certain parameters as shown in Table 1.

TABLE 1. MODEL-BASED DRL ALGORITHMS

Ref#	Algorithm	Application addressed	Limitations/ Assumptions	Gap Addressed
[8]	free model-based fuzzy reinforcement learning. Combination of integrated fuzzy sets and reinforcement learning.	Control Cancer Cells growth. reducing cancer cell population by using two different drugs dosages	The parameters mentioned in this algorithm are arbitrary and selected by trial and error.	Policy Optimization
[9]	deep reinforcement learning (DRL) based approach to accomplish the weight-tuning. Epsilon greedy process. The optimization problem minimizes a weighted sum of doses to four critical organs with doze balance.	high-dose-rate brachytherapy (HDRBT) for cervical cancer.	The VPN approach is potentially applicable to external beam therapy.	epsilon greedy for reward optimization.
[10]	model based on a composition of fuzzy system combined with a neural network.	Lung cancer nodules detection.	Model works good for a specific type of data and samples.	
[11]	Parameter Tuning Policy Network (PTPN) trained via an end-to-end reinforcement learning procedure	a general framework on the development of a strategy to improve CT image quality	only considered images with a relatively low resolution in a small number of cases. PTPN has to wait for the iterative process to finish, before it can adjust parameters	Policy optimization
[12]	Markov decision process that simultaneously optimizes lung cancer detection. trained a dynamic Bayesian network as an observational model and used inverse reinforcement learning to discover a rewards function based on experts' decisions.	Lung Cancer	a discrete time model may not be well-suited for instances of imaging observations at irregular frequencies. Stochastic nature of patient screening probabilities.	Reward Function
[13]	Q-Rank, to predict the sensitivity of cell lines to anti-cancer drugs Q-Rank integrates different prediction algorithms and identifies a suitable algorithm for a given application. models are automatically ranked based on non-scored meta-features. The ranking policy is identified using batch reinforcement learning. The top-ranked model(s) is (are) used to predict drug responses	Predict drug sensitivity for therapy of cancer.	Model is not scalable or generic.	Policy optimization

TABLE 2. MODEL-FREE DRL ALGORITHMS

Ref#	Algorithm	Application addressed	Limitations/ Assumptions	Gap Addressed
[14]	Value-based reinforcement learning approach (TD and Q-Learning)	Tumour localization of Lung cancer	the most challenging issue of applying deep reinforcement learning models to lung cancer treatment is to define an appropriate reward function that is used to update the Q-value for each action.	
[15]	Q-learning-based approach for the closed-loop control of drug dosing related to chemotherapy. Based on cancer and normal cells count, scaled value of the error is used in the reward function.	Cancer in different age groups. different RL agents are developed to address the drug-dosing control in each of these cases	different RL agents need to be trained to account for the patient characteristics of different patient groups.	
[16]	Q-Learning method for drug dosing closed-loop control. 30,000 training episodes are considered. Tumor volume is used in the reward function as an error less than or equal to 1.	Antiangiogenic therapy for Tumor volume reduction.	since the model has high complexity, it is impossible to use model-based controllers.	
[17]	RL based Q Learning model free method for closed loop control of cancer chemotherapy drug dosing.	control of cancer chemotherapy drug dosing.	Does not handle patient dynamics.	Policy Optimization
[18]	a three component neural networks framework with a deep Q-network (DQN) was developed for deep reinforcement learning (DRL) of dose fractionation adaptation.	Automated radiation adaptation in lung cancer.	customization of the reward function if individual cases were to be considered. Limited sample set.	Reward function customization to observe policy changes

[19]	model based on a composition of fuzzy system combined with a neural network.	Lung cancer nodules detection.	Model works good for a specific type of data and samples.	
[20]	Q-Learning algorithm using different reward functions to model different constraints in cancer treatment	Optimal chemotherapy drug dosage for cancer treatment	Specific reward function for certain scenarios. Scale value of error in reward function.	Reward function optimization
[21]	online integral reinforcement learning (IRL) algorithm is designed to provide optimal drug dosing for a given performance measure.	sedative drug dosing to maintain a required level of sedation.	Numerical results are presented using only 10 simulated patients. Stochastic nature like time delays, nonlinearities, and nonnegative constraints are not handled.	Reward optimization

V. DISCUSSION

This survey has been divided into two types of DRL Algorithms i-e Model Based DRL and Model Free DRL. Both DRL methods have achieved some levels of success but there are still a lot of optimizations that need to be done and there are many limitations of each finding that need to be addressed as well. There are many shortcomings of the past research in the form of limitations. These are summed up here to identify the research gaps for future use:

- Models proposed do not handle the stochastic nature of patient responses as they are tested on a few patients or small datasets with specific known body characteristics.
- Proposed models are not generic in nature i-e they do not handle a variety of cancer patients but are designed too specific for a few scenarios thus limiting the use of these models in vast levels of treatments.
- Finally, the models are all designed as single agents. They can also be implemented in a multiagent scenario to get a better performance.

Many future research directions can be concluded from these limitations of past DRL models for cancer treatment. Future DRL model should comprise of:

- Handle stochastic nature of cancer disease and handle noisy or incomplete states.
- Deal with credit assignment problem.
- Should be ideal in cost of exploration and exploitation.
- Reward function should be adaptive and not the misleading one.
- Algorithm should be model free and based on multi modal approach.

VI. PROPOSED SOLUTION MODEL

The proposed solution model addresses the following limitations as evaluated in our critical analysis:

- Multi agent creates a knowledge base that helps in reducing information overlaps and voting system creates more reliable and contributing agent is highlighted as well.
- Deep RL will provide more coverage to multidimensional environments with its NN learning the policies and optimizing the reward without huge Q-Table.
- Hyperparameter tuning using NN will help in balancing the cost of exploration and exploitation.

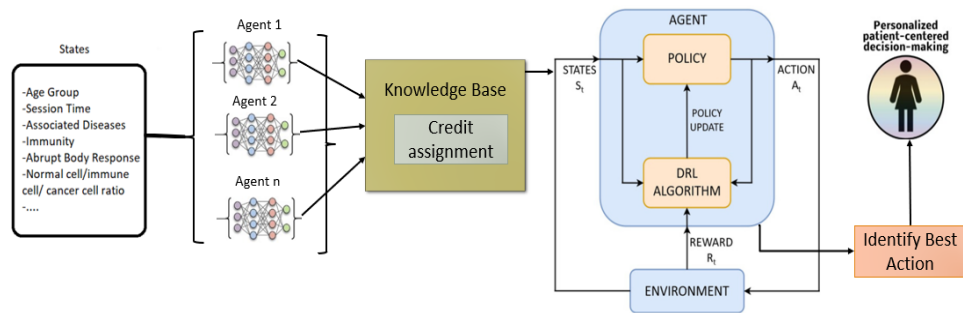


Fig 1. Proposed Multi-Agent DRL Model for Cancer Patients Treatment

A. Experiment

In this research, we have proposed a multi-agent DRL model that can support decision-making for the treatment of cancer patients in stochastic clinical scenarios. Stochastic scenarios are the ones that are partially observable and thus handled through stochastic policies where the agent assigns a probability to each action and selects the action based on the probability value. The approach used is a model-free Deep Q-Network algorithm. When the environment is straightforward the Q-Learning algorithm performs

exceptionally well. However, the database becomes too large, and tabular methods are no longer useful when there are billions of possible unique states and thousands of different actions for each of them. Mnih et al. [22] created the Deep Q-Networks (DQN) algorithm to address this. Deep neural networks (DNNs) and the Q-learning method are combined in this algorithm. DNN is a non-linear function approximator. Thus, it is used in DQN to approximate Q values by replacing the need of a table to store Q values

which is very complex and nearly impractical in the case of a large number of states [23].

B. Dataset

Dataset is taken from a famous repository Kaggle Dataset includes sociodemographic information about cancer patients, signs and symptoms of the disease, histological and imaging characteristics, and TNM stage information. The records also include the patient's diagnostic procedures and available treatments Our model is trained and tested with Dataset of Colorectal cancer patients.

C. State Space

State space consists of around 25 discrete values. These states for Colorectal Cancer patients are listed below:

TABLE 3. INPUT STATES DATASET1

Sno.	States	Data type	Value
1	Age	Integer	
2	Diarrhoea	Boolean	0 - 100
3	Constipation	Boolean	0,1
4	blood stool	Boolean	0,1
5	abdominal pain	Boolean	0,1
6	weight loss	Boolean	0,1
7	Fatigue	Boolean	0,1
8	Biopsy	Boolean	0,1
9	Colonoscopy	Boolean	0,1
10	Imaging	Boolean	0,1
11	US	Boolean	0,1
12	CXR	Boolean	0,1
13	Location	Ordinal Categorical	1,2,3,4
14	Hist_type	Ordinal Categorical	1,2
15	Hist_grade	Ordinal Categorical	1,2,3
16	TNM_stage	Ordinal Categorical	1,2,3,4
17	Clinical_stage	Ordinal Categorical	1,2,3
18	Lymph_node	Boolean	0,1
19	Vascular_Invasion	Boolean	0,1
20	Residual_Tumor	Boolean	0,1
21	CEA_baseline	Ordinal (0 to 10)	0 to 10
22	Dist_metastasis	Boolean	0,1
23	Liver	Boolean	0,1
24	Lung	Boolean	0,1
25	Peritonuem	Boolean	0,1

D. Action Space

Action space in this scenario consists of 3 discrete values and any of their combination will be an action taken to interact with the environment. There are three types of actions that can be taken to interact with the environment. The actions for the Colorectal cancer dataset include:

TABLE 4. ACTIONS SET DATASET1

Sno	Action	Data type	Value
1	Chemotherapy	Boolean	0,1
2	Surgery	Boolean	0,1
3	Radiotherapy	Boolean	0,1

E. Reward

The reward is considered a binary value of 0 or 1 in case if patient stays alive or does not survive.

F. Custom Environment Creation

To combine states and actions, we have created a custom environment with the help of the Python library OpenAI Gym. A consistent interface for engaging with environments is offered by OpenAI Gym, making it simpler to compare and replicate results across various algorithms and academic articles. Therefore, if everything has the same structure, you can easily train and test multiple settings with different algorithms [24]. OpenAI Gym is an extremely user-friendly and flexible library for creating real-time custom environments in addition to gaming environments offered by the gym [25]. In this environment a patient comes with some symptoms that are considered input states, those states are passed to the agent to train against the three actions decided above. Based on the reward function, policies are built and NN optimizes the training agent's abilities by helping out in learning faster and applying those built policies for test data.

VII. RESULTS

In this research, DRL agents have been passed through multiple trials and the returned accuracy values are recorded with varying numbers of agents as well. The results recorded have been stored in Table 5.

TABLE 5. TRIAL RESULTS

Parameters	Agents=2	Agents=3	Agents=4
Trials	500	500	250
Timestep	1000	2000	3000
Learning_rate	0.000237875	0.009897	0.004275607782
buffer_size	56288	39537	10495
Gamma	9102253735	0.902997	0.8796811109
exploration_fraction	0.4484000	0.1663777	0.14679955981
batch_size	1510	1749	735
Accuracy	0.91	0.94	0.94

Results show that using three agents we can achieve an accuracy value of 94% and there is no additional change in accuracy if we increase the no of agents.

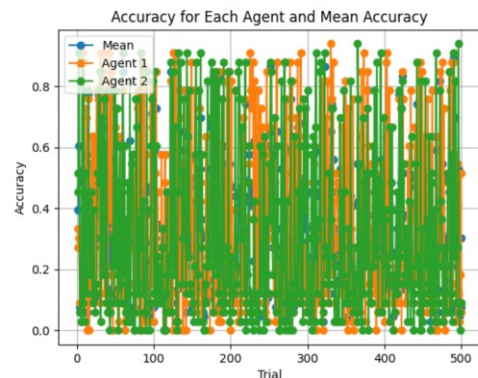


Figure 1. Accuracy Plot with 2 Agents

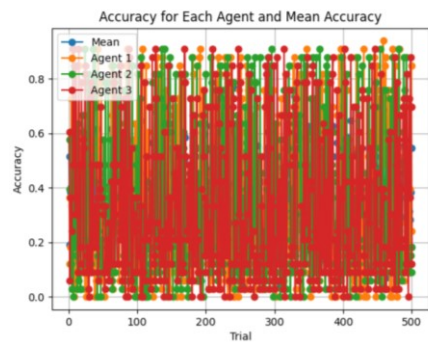


Figure 2. Accuracy Plot with 3 Agents

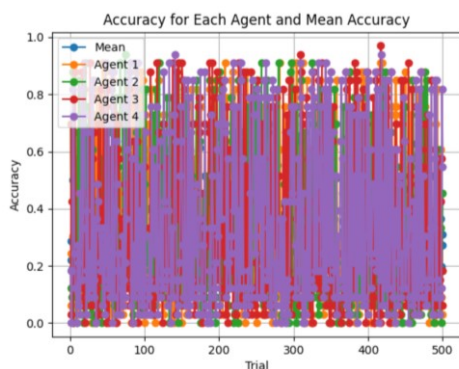


Figure 3. Accuracy Plot with 4 Agents

VIII. CONCLUSION

There are many methods proposed to solve automated disease detection and drug dosing through deep reinforcement learning. Most of the methods proposed so far are so immature or they are not generic with lots of limitations that we need to contribute much in this field. This analysis evaluates each past approach to deal with the stochastic nature of the DRL environment and treatment automation with the help of certain parameters. Finally, it proposes an experimented model that fills up the required gaps and this model proves to provide results with an accuracy of 94%.

REFERENCES

- Krizhevsky A, Sutskever I, Hinton GE, 2012. ImageNet classification with deep convolutional neural networks. Proc 25th Neural Information Processing Systems, p.1097- 1105
- Gu SX, Lillicrap T, Sutskever I, et al., 2016. Continuous deep Q-learning with model-based acceleration. Proc 33rd Int Conf on Machine Learning, p.2829-2838
- Sutton RS, Barto AG, 2018. Reinforcement Learning: an Introduction (2nd Ed.). MIT Press, Cambridge, MA, USA
- Hessel M, Modayil J, van Hasselt H, et al., 2018. Rainbow: combining improvements in deep reinforcement learning. <https://arxiv.org/abs/1710.02298>
- Haarnoja T, Tang HR, Abbeel P, et al., 2017. Reinforcement learning with deep energy-based policies. Proc 34th Int Conf on Machine Learning, p.1352-1361
- Chua K, Calandra R, McAllister R, et al., 2018. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. Proc 32nd Neural Information Processing Systems, p.4754-4765
- Levine S, Koltun V, 2013. Guided policy search. Proc 30th Int Conf on Machine Learning, p.1-9.
- Goharimanesh, M., Akbari, A., & Lotfi, B. (2015, October). On using fuzzy reinforcement learning to control the cancer cells. In 1st International Nastaran Cancer Symposium-2015.
- Shen, C., Gonzalez, Y., Jung, H., Chen, L., Qin, N., & Jia, X. (2018). Automatic Inverse Treatment Planning for Cervical Cancer High Dose-Rate Brachytherapy via Deep Reinforcement Learning. International Journal of Radiation Oncology, Biology, Physics, 102(3), e540.
- Capizzi, G., Sciuto, G. L., Napoli, C., Połap, D., & Woźniak, M. (2019). Small lung nodules detection based on fuzzy-logic and probabilistic neural network with bioinspired reinforcement learning. IEEE Transactions on Fuzzy Systems, 28(6), 1178-1189.
- Shen, C., Gonzalez, Y., Chen, L., Jiang, S. B., & Jia, X. (2018). Intelligent parameter tuning in optimization-based iterative CT reconstruction via deep reinforcement learning. IEEE transactions on medical imaging, 37(6), 1430-1439.
- Petousis, P., Winter, A., Speier, W., Aberle, D. R., Hsu, W., & Bui, A. A. (2019). Using sequential decision making to improve lung cancer screening performance. Ieee Access, 7, 119403-119419.
- Daoud, S., Mdhaffar, A., Jmaiel, M., & Freisleben, B. (2020). Q-Rank: Reinforcement Learning for Recommending Algorithms to Predict Drug Sensitivity to Cancer Therapy. IEEE Journal of Biomedical and Health Informatics, 24(11), 3154-3161.
- Liu, Z., Yao, C., Yu, H., & Wu, T. (2019). Deep reinforcement learning with its application for lung cancer detection in medical Internet of Things. Future Generation Computer Systems, 97, 1-9.
- Padmanabhan, R., Meskin, N., & Haddad, W. M. (2020). Reinforcement learning-based control of drug dosing with applications to anesthesia and cancer therapy. In Control Applications for Biomedical Engineering Systems (pp. 251-297). Academic Press.
- Yazdjerdi, P., Meskin, N., Al-Naemi, M., Al Moustafa, A. E., & Kovács, L. (2019). Reinforcement learning-based control of tumor growth under anti-angiogenic therapy. *Computer methods and programs in biomedicine*, 173, 15-26.
- Padmanabhan, R., Meskin, N., & Haddad, W. M. (2017). Learning-based control of cancer chemotherapy treatment. IFAC-PapersOnLine, 50(1), 15127-15132.
- Tseng, H. H., Luo, Y., Cui, S., Chien, J. T., Ten Haken, R. K., & Naqa, I. E. (2017). Deep reinforcement learning for automated radiation adaptation in lung cancer. Medical physics, 44(12), 6690-6705.
- Capizzi, G., Sciuto, G. L., Napoli, C., Połap, D., & Woźniak, M. (2019). Small lung nodules detection based on fuzzy-logic and probabilistic neural network with bioinspired reinforcement learning. IEEE Transactions on Fuzzy Systems, 28(6), 1178-1189.
- Padmanabhan, R., Meskin, N., & Haddad, W. M. (2017). Reinforcement learning-based control of drug dosing for cancer chemotherapy treatment. *Mathematical biosciences*, 293, 11-20.
- Padmanabhan, R., Meskin, N., & Haddad, W. M. (2019). Optimal adaptive control of drug dosing using integral reinforcement learning. *Mathematical biosciences*, 309, 131-142.
- Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M., 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G. and Petersen, S., 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540), pp.529-533.
- Vidyadhar, V., Nagaraj, R. and Ashoka, D.V., 2021. NetAI-Gym: Customized Environment for Network to Evaluate Agent Algorithm using Reinforcement Learning in Open-AI Gym Platform. *International Journal of Advanced Computer Science and Applications*, 12(4).
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. and Zaremba, W., 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Investigating Threats to ICS and SCADA Systems via Honeypot Data Analysis and SIEM

Tameem Ud Din
Computer Systems Engineering
UET Peshawar
Peshawar, Pakistan
20PWCSE1866@uetpeshawar.edu.pk

Usman Zia
Computer Systems Engineering
UET Peshawar
Peshawar, Pakistan
20PWCSE1868@uetpeshawar.edu.pk

Mahnoor
Computer Systems Engineering
UET Peshawar
Peshawar, Pakistan
20PWCSE1928@uetpeshawar.edu.pk

Dr. Laiq Hasan
Computer Systems Engineering
Univeristy of Engineering and Technology Peshawar
Peshawar, Pakistan
laiqhasan@uetpeshawar.edu.pk

Syed M. Ali Uddin Hafee
CCEE
NED University of Engineering and Technology
Karachi, Pakistan
hafee.pg3602185@cloud.neduet.edu.pk

Abstract—Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems (ICS) are crucial for managing essential infrastructure like energy, water, transportation, and manufacturing. As these systems become more connected to the internet, they face increasing cyber threats, such as viruses, ransomware, and targeted attacks. These threats can lead to significant consequences, including data loss and disruptions to operations. To improve cybersecurity for SCADA and ICS, it's important to understand and address these evolving threats. While traditional security methods are vital, they may not be enough against constantly changing threats. That's why we need innovative approaches to grasp how cyber attackers operate. Honeypot technology provides a useful tool for studying cyber threats by creating fake systems to attract attackers and observe their techniques. This is especially important for SCADA and ICS, where breaches can cause serious damage. Additionally, Security Information and Event Management (SIEM) systems help by gathering and analyzing data from different sources to detect and respond to threats in real-time. This paper aims to fill gaps in understanding cyber threats to SCADA and ICS by using honeypot data and SIEM tools. By simulating Programmable Logic Controllers (PLCs) through honeypots and connecting it with SIEM Solution, we can actively monitor and tackle security events, preventing potential harm before it happens.

Keywords—SCADA; ICS; OT; SIEM; SOAR; SOC; Critical Infrastructure Protection; Incident Response; Log Analysis; Threat Intelligence; Honeypot; Attack Detection.

I. INTRODUCTION

Supervisory control and data acquisition (SCADA) systems are considered a type of industrial control systems that allows users monitor, acquire data from, and control industrial processes locally or remotely through sensors and actuators [1]. Power plants and water treatment facilities are examples of traditional industrial systems that were designed to operate in highly controlled and separated settings. However, the recent exposure of Industrial Control Systems (ICS) to the Internet has made access and technological adaptation easier, which has led to the exploitation of security holes by attackers to launch attacks against ICS [2]. These attacks can significantly

impact the economics and national security of countries. To identify possible threats and comprehend the terrain of these assaults, ICS honeypots are deployed [3]. Honeypots are an interesting security concept; instead of keeping attackers out, you want to invite them in [4]. They are typically divided into three categories: low-, medium-, and high-interaction. The key distinctions between these types are their capacities for data collection, maintenance, and deployment [5]. Generally speaking, the higher the level of interaction, the more data the honeypot can capture. High interaction ones are most similar to real systems and can collect the most data. In industrial environments where attacks occur, such as ICS/SCADA, honeypots need to be very similar to real systems in order to reduce detection risk [6].

This paper investigates the threats to SCADA and ICS systems using open source honeypot deployment, log analysis, and integration with open source SIEM solutions to enhance threat detection capabilities and incident response. Furthermore, It studies the viability of putting open-source honeypots in the cloud to detect attacks on these systems. In addition, the study investigates how to integrate these honeypots with an open-source SIEM solution. This combination strategy tries to improve threat detection by connecting honeypot data with other security logs. Finally, the research extends beyond simply recognizing attackers. It digs into a deep examination of the collected traffic to determine attack kinds, sources, and other useful information for creating defensive tactics. The core contributions of this paper are:

- Evaluation of Open-Source Honeypot Deployment on Cloud
- Integration of Honeypot with Open-Source SIEM for Threat Intelligence
- In-Depth Analysis of Attack Patterns

This is how the remainder of the paper is organized. Section 2 investigates the concepts of SIEM and honeypots

and reviews related literature. Section 3 explains the approach and the experimental setting. The findings are examined and described in Section 4. Section 5 concludes the paper.

II. BACKGROUND AND RELATED WORK

This section gives an overview of honeypots, SIEM Solution, summarizes relevant efforts, and honeypot deployments.

A. Honeypots

Honeypots are software applications designed to mimic real systems, deceiving attackers while serving as a decoy. Honeypots are classified into two types: production honeypots, which are installed within a network to deceive insiders, and research honeypots, which are deployed on the internet to attract attackers from the outside. Furthermore, honeypots can be classed according to how closely they engage with attackers. Low-interaction honeypots simulate basic services but do not completely function, whereas high-interaction honeypots provide a full operating system with real services [7]. To set up honeypot, a specific honeypot image is deployed on the machine. Once configured, a port is assigned to the honeypot. Any attempt to access this port redirects the attacker to the honeypot, which appears to be a genuine system. Through this redirection, the attacker's machine ID, IP address, and other critical parameters are logged in the honeypot's log file, using these logs we can move to further analysis. We used conpot, an open-source, general-purpose ICS honeypot.

B. Conpot

The ICS honeypot we have used is called Conpot. Conpot is a SCADA honeypot that serves as a valuable tool for emulating SCADA systems and detecting potential threats within SCADA device networks. Developed by The HoneyNet Project, Conpot is designed to be easy to implement and provides simulation capabilities for protocols like HTTP, Modbus, and SNMP, as well as integration with programmable logic controllers (PLC). It features a logging system that monitors and records any unauthorized changes made by intruders, offering detailed event logs with millisecond accuracy. By mimicking the behavior of real SCADA systems, Conpot utilizes a logging system to monitor any changes that are made by intruders. The honeypot logs events of HTTP, SNMP and Modbus services with millisecond accuracy and offers basic tracking information such as source address, request type, and resource requested in the case of HTTP [8].

C. Security Information and Event Management

Security Information and Event Management (SIEM) is a vital security software/platform that analyzes security events. It works by examining log files sent to it and following the paths we set. SIEM analyzes log files using established rules and generates alerts depending on the results. Custom rules can be written for specific types of logs to generate alerts when certain circumstances are satisfied. To successfully analyze logs, we must create a decoder for any log types that do not follow the standard format. SIEM does, however, include preconfigured decoders for JSON structured logs.

D. Related Work

The use of honeypots as a security precaution and research tool has been extensively studied, including industrial control systems (ICS) and supervisory control and data acquisition (SCADA) for protecting Operational Technology (OT) environments. The following literature review focuses on significant studies and research efforts in the subject of honeypots and their use to secure critical infrastructure.

Nasution et al. concentrate on the low-interaction honeypot HONEYD's potential for enhancing security systems in their paper [9]. They go through how HONEYD is configured and deployed, as well as how it can identify and record assaults, giving security analysts important information. A thorough analysis of the application of honeypots and honeynets in Cyber-Physical Systems (CPS), Industrial IoT (IIoT), and Internet of Things (IoT) is provided by Franco et al. [10]. Their research emphasizes the benefits and drawbacks of setting up honeypots in these settings and stresses the significance of acquiring information on new dangers. Jicha et al. [11] investigate the characteristics of SCADA honeypots, with a special emphasis on the Conpot variation. Their research provides a thorough examination of Conpot's capabilities, including its ability to mimic numerous industrial protocols and collect important information on prospective assaults on SCADA systems. Sharma and Kaul [12] conduct a detailed investigation on intrusion detection systems and honeypot-based proactive security measures in Vehicular Ad-hoc Networks (VANETs) and VANET clouds. Their findings emphasize the necessity of honeypots for recognizing and mitigating risks in these dynamic and diverse contexts. Uitto et al. [13] describe the anti-honeypot strategies and procedures used by attackers to discover and avoid honeypots. Their research underlines the significance of establishing strong and resilient honeypot systems to fight such evasion strategies while maintaining effective threat monitoring and analysis. Cao et al. [14] describe DiPot, a distributed industrial honeypot system that monitors internet scanning and attack behaviors on industrial control systems. DiPot's strengths are its high-level modeling, broad data analysis capabilities, and accessible graphical frontend, which allow users to obtain insights into the present condition of ICS security. Lopez-Morales et al. [15] introduce HoneyPLC, a highly interactive, adaptable, and malware-collecting honeypot that supports a wide range of PLC models and manufacturers. Their trials show HoneyPLC's capacity to successfully disguise itself as a real device, enticing and misleading attackers while collecting vital data samples for further study. Radoglou-Grammatikis et al. [7] provide TRUSTY, a strategic custom honeypot deployment and analysis framework. It uses an adversarial game model between the attacker and the defense to optimize the number of honeypots deployed based on the attacker's behavior and the available computing resources.

E. Research Gap

These studies show the growing importance of honeypots in critical infrastructure security, notably for SCADA and ICS

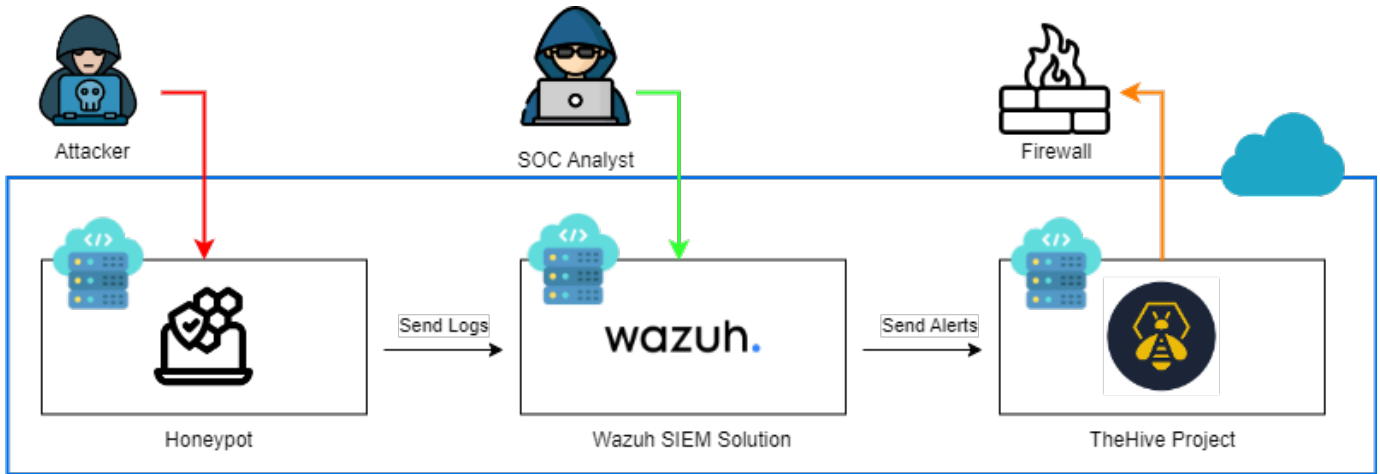


Fig. 1. Holistic Approach to SCADA/ICS Threat Investigation with Honeypot, Wazuh, and TheHive

systems. They provide useful insights on honeypot deployment, configuration, and analysis, as well as the problems and tactics involved in their effective use in detecting and mitigating cyber threats. While earlier research has examined the impact of honeypot deployment, there is still a gap in building a real-time threat analysis system. This system would use open-source and free tools to continually monitor honeypot data and extract threat intelligence in order to proactively mitigate threats.

III. EXPERIMENT APPROACH

This study offers a comprehensive approach to investigating cyber hazards to SCADA and ICS systems. As shown in Fig. 1, it uses honeypot technology in conjunction with Wazuh, an open-source SIEM system, to analyze threats and attacks in real time. For one week, the honeypot was deployed in a containerized environment on a cloud platform and exposed to the internet in order to collect logs. A persistent storage solution within the container ensured log retention for Wazuh's real-time analysis. Wazuh alerts were also linked into TheHive project, allowing for the building of playbooks. These playbooks managed the blocking of new malicious IP addresses detected by the honeypot using TheHive's SOAR platform, automating the response to possible threats.

A. Experiment Setup

The experiment was set up by installing and configuring the Conpot honeypot, emulating a SCADA environment with various industrial protocols such as Modbus, SNMP, and HTTP. The honeypot was then exposed to the internet to attract potential attackers and gather real-world threat data. In parallel, the Wazuh SIEM agent was installed on the same system hosting the Conpot honeypot. The Wazuh agent was configured to monitor and process the log files generated by the honeypot, enabling the analysis of security events and potential threats in the dashboard of Wazuh. Further, the Wazuh was integrated with TheHive for security orchestration and

automation purposes. The details of each stage are discussed in the subsections below.

1) Conpot Deployment:

a) *Selection of Honeypot:* The Conpot honeypot was used for this study because it is open source and can efficiently simulate SCADA systems. Conpot supports a variety of industrial protocols, including Modbus, SNMP, and HTTP, making it ideal for replicating a realistic SCADA system.

b) *Installation and Configuration:* The Conpot honeypot was packed into a Docker container and placed on a dedicated system, with all necessary dependencies such as Python, Twisted, Scrapy, and PyYAML. The honeypot was then set up to imitate the needed SCADA system characteristics, such as supported protocols and services. The IEC 60870-5-104 protocol was disabled, and the default template for the configuration was utilized.

c) *Exposure to the Internet:* To attract real-world cyber-attacks and collect vital threat intelligence, a honeypot with a small footprint (Conpot) was deliberately exposed to the internet. This included opening critical ports and making the honeypot accessible via external networks. We mapped the typical ports used by common SCADA protocols to the Docker container running Conpot, leaving its default settings for a realistic attack surface.

2) SIEM Solution Installation and Configuration:

a) *SIEM Solution Selection:* The Wazuh SIEM Solution was chosen for this study because it is open-source, scalable, and compatible with a variety of log formats, including JSON, the major format utilized by the Conpot honeypot.

b) *Installation and Configuration:* The Wazuh agent was installed on the server that ran the Conpot honeypot, and the Wazuh SIEM was configured to monitor and process not only the Conpot log files but also monitor the systems logs of the server running conpot. Furthermore, the Wazuh indexer server and dashboard was installed on a separate server.

c) *Rule Definition:* Custom rules were built within the Wazuh SIEM to help with the examination of the Conpot log

files. These rules were designed to detect specific patterns, abnormalities, or occurrences of interest associated with SCADA system attacks and vulnerabilities.

3) Data Collection and Analysis:

a) *Logs Monitoring:* The Conpot honeypot logs were constantly monitored, and any interactions or attacks were recorded with millisecond delay, including source IP addresses, requested resources, and any attempt to make unauthorized changes by possible attackers.

b) *Logs Analysis through Wazuh Dashboard:* The Wazuh SIEM dashboard was used to examine the log data collected from the Conpot honeypot, making use of specified rules and decoders. Figure 2 illustrates the Wazuh SIEM dashboard, which enables real-time monitoring and analysis of security alerts and events. This investigation sought to uncover potential threats, attack patterns, and vulnerabilities associated with SCADA systems.

c) *Threat Intelligence Gathering:* The analysis of the Conpot honeypot logs and SIEM alerts provided valuable threat intelligence, including the tactics, techniques, and procedures (TTPs) employed by malicious actors targeting SCADA systems. Additionally, the public IP addresses of potential threat actors were collected for further investigation or potential blocking.

4) Integration with TheHive:

a) *Installation and Configuration:* TheHive integration involved setting up a dedicated server for its deployment. Within TheHive, a new organization was created specifically for managing security incidents. A dedicated user account was also created and assigned to this organization. To establish communication between TheHive and Wazuh, an API-based connection was configured. This two-way communication channel allows Wazuh to automatically send security alerts to TheHive for further analysis and response.

b) *Writing Playbooks:* TheHive’s playbook functionality was leveraged to automate response actions based on incoming alerts. Playbooks were written to orchestrate specific actions, such as blocking malicious IP addresses identified by the honeypot or SIEM system directly on the firewall. This automated response streamlines the security workflow, allowing for faster threat mitigation and minimizing potential damage to SCADA/ICS systems.

When the honeypot or SIEM detects a potential threat, such as a hostile IP address attempting to exploit a vulnerability, theHive playbooks can be automatically activated. These playbooks would then execute a predetermined set of steps, such as blocking the malicious IP address at the network layer or isolating infected systems. This quicker response helped to reduce the impact of cyberattacks and prevent potential harm to SCADA/ICS systems.

IV. RESULTS AND DISCUSSION

A. Observed Attacks and Threats

The analysis of the Conpot honeypot logs and Wazuh SIEM alerts provided valuable insights into the tactics, techniques,

and procedures (TTPs) employed by malicious actors targeting SCADA systems. Figure 3 showcases the top tactics and techniques observed in attacks, categorized according to the MITRE ATT&CK framework. The following subsections provide a detailed overview of the findings.

1) *Types of Attacks:* The study identified several types of attacks attempted against the simulated SCADA environment, including:

a) *Brute Force Login Attempts:* A significant number of brute force attacks (7,548) were observed, where attackers attempted to gain unauthorized access to the system by trying multiple combinations of non-existent usernames and passwords.

b) *Reconnaissance and Vulnerability Scanning:* Numerous attempts were made to probe the honeypot for open ports, services, and potential vulnerabilities. These reconnaissance efforts are often precursors to more advanced attacks.

c) *Unauthorized Access Attempts:* Attackers tried to access the system without proper credentials, potentially aiming to exploit known vulnerabilities or misconfigurations. A total of 23 such unauthorized access attempts were recorded.

d) *Exceeding Authentication Limits:* In some instances, attackers exceeded the maximum allowed authentication attempts, indicating their persistence in trying to gain unauthorized access.

2) *Attacker’s Region:* The analysis of the source IP addresses revealed that the attacks originated from various regions across the globe. Table I presents the distribution of attack sources by country or region. As shown in Table I,

TABLE I
DISTRIBUTION OF ATTACK SOURCES BY COUNTRY/REGION

Country/Region	Number of Unique IP Addresses
United States (US)	201
China (CN)	12
United Kingdom (GB)	10
Germany (DE)	10
Pakistan (PK)	8
Netherlands (NL)	8
Russia (RU)	8
Others	52

the majority of attacks originated from the United States, with 201 unique IP addresses involved. Other notable sources include China, the United Kingdom, Germany, Pakistan, the Netherlands, and Russia.

3) *Protocols:* The honeypot was configured to emulate various industrial protocols commonly used in SCADA systems. Table II presents the distribution of attacks based on the targeted protocols and the corresponding port numbers. As

TABLE II
DISTRIBUTION OF ATTACKS BY PROTOCOL AND PORT NUMBER

Protocol	Port Number	Number of Attacks
HTTP	80	1168
SNMP	161	150

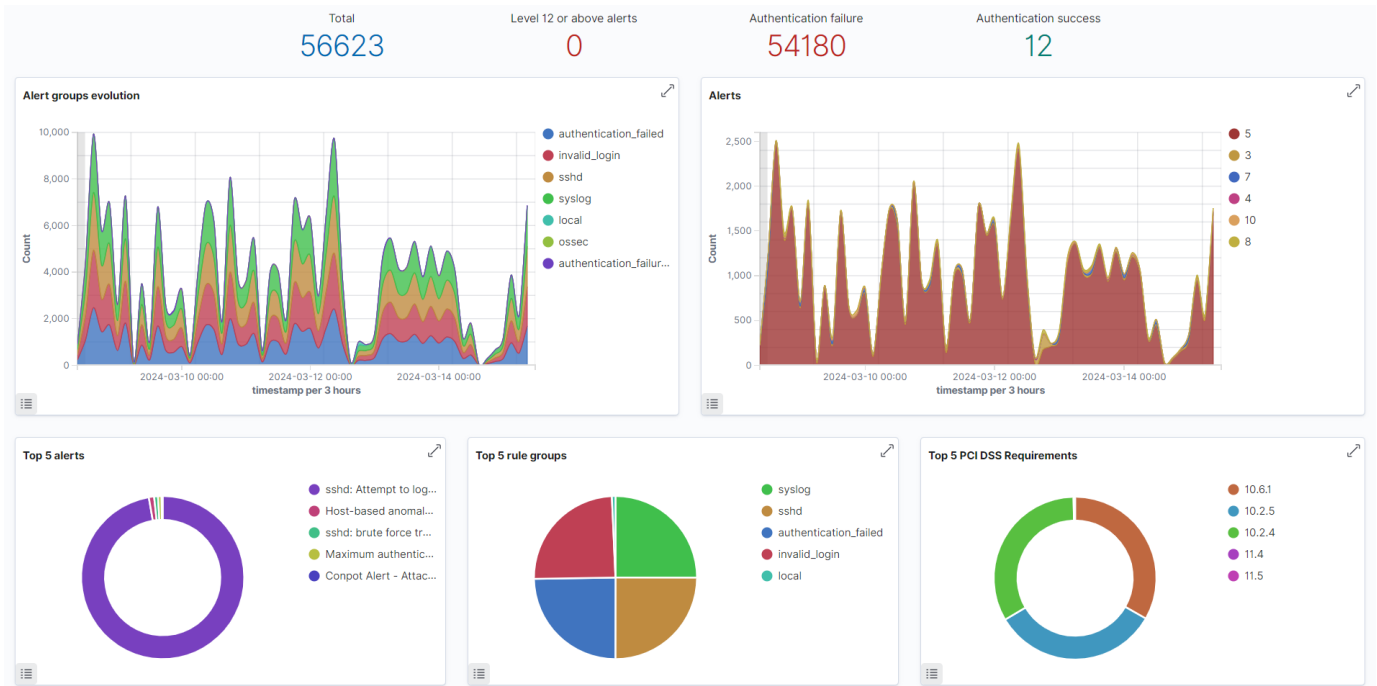


Fig. 2. Wazuh SIEM Dashboard: Real-time Monitoring and Analysis of Security Alerts and Events

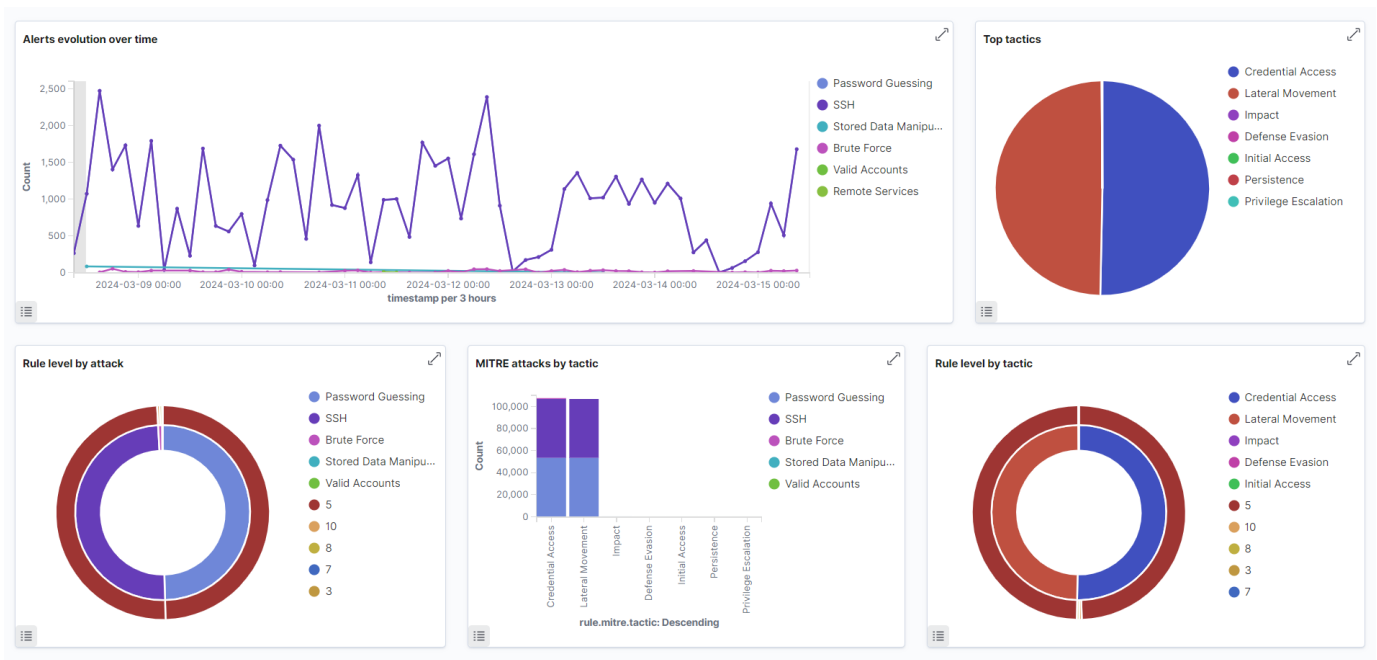


Fig. 3. Wazuh SIEM Dashboard: Top Tactics and Techniques Observed in Attacks Based on the MITRE ATT&CK Framework

shown in Table II, the majority of attacks (1,168) targeted the HTTP protocol on port 8800, while 150 attacks aimed at the SNMP protocol on port 16100. It is important to note that the honeypot was intentionally exposed to the internet to attract potential attackers and collect real-world threat data. The observed attacks and their distribution provide valuable insights into the tactics, techniques, and procedures (TTPs) employed by malicious actors targeting SCADA and ICS systems.

B. Discussion

The study revealed a significantly higher number of attacks targeting the HTTP protocol on port 8800 compared to other protocols like SNMP. This observation can be attributed to several factors. Firstly, HTTP is a widely adopted protocol, and many attackers attempt to exploit web-based vulnerabilities or misconfigurations, making it an attractive target for reconnaissance and potential exploitation.

Another potential reason for the high HTTP traffic could be the detectability of the default Conpot web interface. While Conpot aims to emulate a realistic SCADA environment, the default configuration and web interface may have been recognized by attackers as a honeypot, prompting them to focus their efforts on the HTTP service. To enhance the authenticity of the simulated environment and attract a more diverse range of attack patterns, customizing the Conpot configuration to better mimic specific SCADA systems or industrial environments could be beneficial.

Furthermore, the study did not explicitly mention attacks targeting the SSH protocol, which is often used for remote administration. Leaving the SSH port open, although not directly related to SCADA protocols, could potentially attract unwanted brute-force attacks or unauthorized access attempts, which may not be relevant to the study's scope. It is recommended to avoid exposing unnecessary ports to the internet when deploying honeypots or simulating SCADA environments to maintain a focused and relevant attack surface.

Lastly, the relatively lower traffic observed for protocols like SNMP compared to HTTP could be due to the difficulty in accurately emulating the behavior and nuances of industrial protocols within a low interaction honeypot environment with default configuration template. To address this challenge, future studies could explore incorporating more realistic industrial protocol implementations, potentially through the use of virtualized or emulated hardware components, and integrating additional data sources or logs from real SCADA systems (with appropriate anonymization and security measures) to enrich the honeypot's behavior and log patterns.

V. CONCLUSION

This study successfully investigated threats to SCADA/ICS systems by combining honeypots, SIEM, and security orchestration. A honeypot like Conpot acted as a decoy, collecting real-world attack data when exposed on a cloud platform. This data was then analyzed by Wazuh, a SIEM solution, to identify attack patterns and sources. Finally, TheHive, a

security orchestration platform, automated responses based on predefined protocols, such as blocking malicious IPs and isolating infected systems. This study demonstrates the effectiveness of this combined approach, and importantly, the value of open-source tools (Conpot, Wazuh, TheHive) in protecting critical infrastructure.

Future work could focus on enhancing the realism of the honeypot deployment, exploring the detection and analysis of more advanced attack techniques, investigating the scalability and performance of the proposed solution, integrating with external threat intelligence platforms, and fostering collaboration and information sharing among organizations and security communities.

REFERENCES

- [1] M. Mesbah, M. S. Elsayed, A. D. Jurcut, and M. Azer, "Analysis of ics and scada systems attacks using honeypots," *Future Internet*, vol. 15, no. 7, p. 241, 2023.
- [2] A. Nechibvute and H. Mafukidze, "Integration of scada and industrial iot: Opportunities and challenges," *IETE Technical Review*, pp. 1–14, 2023.
- [3] A. Ara, "Security in supervisory control and data acquisition (scada) based industrial control systems: Challenges and solutions," in *IOP Conference Series: Earth and Environmental Science*, vol. 1026, no. 1. IOP Publishing, 2022, p. 012030.
- [4] H. Project, *Know your enemy: Revealing the security tools, tactics, and motives of the blackhat community*. Addison-Wesley Professional, 2001.
- [5] S. Maesschalck, V. Giotsas, and N. Race, "World wide ics honeypots: A study into the deployment of conpot honeypots," in *Industrial Control System Security Workshop*, 2021.
- [6] S. Chamotra, J. Bhatia, R. Kamal, and A. Ramani, "Deployment of a low interaction honeypot in an organizational private network," in *2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*. IEEE, 2011, pp. 130–135.
- [7] P. Radoglou-Grammatikis, A. Liatifis, E. Grigoriou, T. Saoulidis, A. Sarigiannidis, T. Lagkas, and P. Sarigiannidis, "Trusty: A solution for threat hunting using data analysis in critical infrastructures," in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2021, pp. 485–490.
- [8] A. Jicha, M. Patton, and H. Chen, "Scada honeypots: An in-depth analysis of conpot," in *2016 IEEE conference on intelligence and security informatics (ISI)*. IEEE, 2016, pp. 196–198.
- [9] A. M. Nasution, M. Zarlis, and S. Suherman, "Analysis and implementation of honeyd as a low-interaction honeypot in enhancing security systems," *Randwick International of Social Science Journal*, vol. 2, no. 1, pp. 124–135, 2021.
- [10] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2351–2383, 2021.
- [11] A. Jicha, M. Patton, and H. Chen, "Scada honeypots: An in-depth analysis of conpot," in *2016 IEEE conference on intelligence and security informatics (ISI)*. IEEE, 2016, pp. 196–198.
- [12] S. Sharma and A. Kaul, "A survey on intrusion detection systems and honeypot based proactive security mechanisms in vanets and vanet cloud," *Vehicular communications*, vol. 12, pp. 138–164, 2018.
- [13] J. Uitto, S. Rauti, S. Laurén, and V. Leppänen, "A survey on anti-honeypot and anti-introspection methods," in *Recent Advances in Information Systems and Technologies: Volume 2 5*. Springer, 2017, pp. 125–134.
- [14] J. Cao, W. Li, J. Li, and B. Li, "Dipot: A distributed industrial honeypot system," in *Smart Computing and Communication: Second International Conference, SmartCom 2017, Shenzhen, China, December 10-12, 2017, Proceedings 2*. Springer, 2018, pp. 300–309.
- [15] E. López-Morales, C. Rubio-Medrano, A. Doupe, Y. Shoshitaishvili, R. Wang, T. Bao, and G.-J. Ahn, "Honeyplc: A next-generation honeypot for industrial control systems," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 279–291.

SkinScan: Cutting-edge AI-Powered Skin Cancer Classification App for Early Diagnosis and Prevention

Maria Sial

Departemtn of Computer Systems Engineering
University of Engineering and Technology,
Peshwar, Pakistan
20pwcse1910@uetpeshawar.edu.pk

Salman Shakeel

Departemtn of Computer Systems Engineering
University of Engineering and Technology,
Peshwar, Pakistan
20pwcse1925@uetpeshawar.edu.pk

Muhammad Asim

Departemtn of Computer Systems Engineering
University of Engineering and Technology,
Peshwar, Pakistan
20pwcse1944@uetpeshawar.edu.pk

Amaad Khalil

Departemtn of Computer Systems Engineering
University of Engineering and Technology,
Peshwar, Pakistan
amaadkhalil@uetpeshawar.edu.com

Muhammad Abeer Irfan

Department of Computer Systemss Engineering
University of Engineering and Technology,
Peshwar, Pakistan
abeer.irfan@uetpeshawar.edu.pk

Atif Jan

Department of Electrical Engineering
University of Engineering and Technology,
Peshwar, Pakistan
atifjan@uetpeshawar.edu.pk

Abstract — Mobile health applications (mHealth) use machine learning (AI)-based algorithms to classify skin lesions; nevertheless, the influence on healthcare systems is unknown. In 2019, a large Dutch health insurance provider provided 2.2 million people with free mHealth software for skin cancer screening. To evaluate the effects on dermatological care consumption, the research conducted a practical transitional and population-based study. To evaluate dermatological needs between the two groups throughout the first year of free access, the research compared 18,960 mHealth users who completed at least one successful evaluation with the app to 56,880 controls who did not use the app. The odds ratios (OR) were then computed. A cost-effectiveness analysis was conducted in the near term to find out the expense for each extra diagnosed premalignancy. Here, results indicates that mHealth users had a three-fold greater incidence of requests for benign tumors on the skin and the nevi (5.9% vs 1.7%, OR 3.7 (95% CI 3.4–4.1)), and they had greater numbers of claims for (pre)malignant skin cancers as groups (6.0% vs 4.6%, OR 1.3 (95% CI 1.2–1.4)). Compared to the existing standard of care, the expenses associated with using the app to detect one additional (pre) malignant skin lesion were €2567. These results suggest that AI in mHealth may help identify more dermatological (pre)malignancies, but this could be weighed against the current greater rise in the need for care for benign tumors of the skin and nevi.

Keywords: Skin Cancer; AI-Powered; Skin Lesions; Skin Cancer Types (Basal Cell Carcinoma, Squamous Cell Carcinoma, Melanoma); Image Classification

I. INTRODUCTION

One of the most popular forms of cancer, skin cancer is becoming more common and more widespread, which puts considerable pressure on healthcare systems [1-4]. Medical technology advancements like tele dermatology and artificial intelligence (AI)-powered mobile health apps are being incorporated into medical treatments as a potential way to reduce this load. AI application could relieve stress on physicians and further minimize associated healthcare expenses by reducing the number of appointments for benign skin cancers and boosting the possibility of early identification of skin cancer [5].

The idea of using AI-based algorithms to identify skin cancer has a lot of assurance, but it has only been studied in controlled laboratory environments. Furthermore, even though AI is comparable to Dermatologists can identify cancers of the skin on dermoscopy-based images [6-8], but it's not yet clear how or for whom to use this technique in clinical care. The public can now use this method because AI-based algorithms for identifying skin cancer have been used in several mHealth apps in recent years [9]. The

Netherlands is in a unique situation because of the quick advancement of AI-based mobile health apps in population-based settings. A mobile app for skin care is being paid for by multiple major health insurance carriers. Cancer detection for their consumers [10], allows consumers to assess whether to see a GP (general practitioner) for a potentially malignant skin lesion using an AI-based mHealth app. By evaluating how these real-world data affect doctors and their patients when they utilize such an app, we can gain a better understanding of the potential impact on healthcare consumption [11]. Thus, the purpose of this study is to assess how a mHealth app for questionable skin lesions affects the use of dermatological healthcare in a population-based scenario.

II. RELATED WORK

Skin cancer is a global issue affecting one in five people, with millions of new cases reported annually. It occurs when normal cells grow uncontrolled and disorderly, leading to contact inhibition of proliferation. There are two main categories of skin cancer: non-melanoma and melanoma. Melanoma skin cancer is further divided into two subtypes: squamous cell carcinoma (SCC) and basal cell carcinoma (BCC). Deep learning algorithms have become popular for early skin cancer detection. Tahir et al. introduce the DSCC_Net, a deep learning-based skin cancer classification network, aiming to identify different types of skin cancer. The model is tested on three standard datasets and provides a robust evaluation framework for six deep networks. DSCC_Net outperforms other models in skin cancer diagnosis, achieving a 99.43% AUC and outperforming baseline models in accuracy, recall, precision, and F1-score, making it a valuable tool for dermatologists and healthcare professionals. [12] Maad M. Mijwil's paper "Skin cancer disease images classification using deep learning solutions" focuses on image classification using deep learning models, which is a fundamental pillar of medical progress. The study uses a convolutional neural network (ConvNet) to detect skin cancer pictures, covering over 24,000 instances. Three architectures—InceptionV3, ResNet, and VGG19—are carefully used to identify the most effective architecture for the classification of benign and malignant cancer types with high accuracy. The research methodology focuses on artificial intelligence, specifically machine, and deep learning, with self-learning algorithms forming the foundation of artificial neural networks. TensorFlow, a flexible open-source library, is used for machine and deep learning tasks. The study uses a large dataset of high-pixel images from the ISIC collection from 2019-2020. The authors propose

an enhanced framework for early skin cancer detection using the well-known CNN-based architecture, VGG-16. The model operates on the foundation of the VGG-16 network but introduces enhancements to improve accuracy. The International Skin Image Collaboration dataset is used for assessment. [13]

The paper "Detection and optimization of skin cancer using deep learning" by Balambigai, Elavarasi, Abarna, Abinaya, and Arun Vignesh focuses on improving Convolutional Neural Network (CNN) models for skin cancer classification. The authors used a dataset from Kaggle and applied random search optimization for hyper-parameter selection, resulting in an improved accuracy of 77.17%. In the article "Classification of Skin Cancer Lesions

Using Explainable Deep Learning," Muhammad Zia Ur Rehman introduced a unique technique by incorporating extra convolutional layers into pre-trained models. The modified DenseNet201 model showed a remarkable accuracy of 95.50%, demonstrating state-of-the-art performance compared to other approaches. The study highlights the importance of computer-aided diagnostic solutions in enhancing the detection process and supporting dermatologists in making informed decisions for early intervention and better patient outcomes.[14]

III. Types of Skin Cancer

Skin cancer can be of three main forms. Melanoma, squamous cell carcinoma, and basal cell carcinoma are the three types of these varieties.

3.1 Basal Cell Carcinoma

This is the type of skin cancer that occurs in the basal cells. Basal cells are present in the lower part of the **outer layer** called the epidermis. On the skin, basal cell carcinoma appears as a tiny, usually shiny bump or scaly, flat patch that slowly expands which we can see in Figure 1.

3.2 Squamous Cell Carcinoma

Squamous cell carcinoma (SCC), also known as cutaneous carcinoma of squamous cells (CSCC), is the second most common skin cancer which we can see in in figure 2, primarily affecting the **outermost layer** of the skin, often found in areas exposed to the sun.

3.3 Melanoma

Melanoma, also known as "Black Tumor," is a dangerous skin cancer caused by melanocytes, which produce **melanin**, a black substance that gives skin shade. It spreads easily and expands quickly in Figure 1.



Figure 1: Main Types of Skin Cancer (Basel Cell Carcinoma, Squamous Cell Carcinoma, Melanoma)

AI algorithms are being developed to improve early skin cancer diagnosis and user experience. The **SkinScan app** will be tested for accessibility, engagement, and impact on preventative healthcare measures. Ethical considerations like user consent and data privacy will be considered when releasing AI-powered health apps. The "SkinScan" project combines advanced technology and user-centered design to improve skin cancer prevention and detection. It uses artificial intelligence algorithms, machine learning models, and ethical guidelines to ensure early diagnosis and tracking of health outcomes.

IV. METHODOLOGY

The proposed skin cancer detection system consists of the following features.

4.1. User Login/Register

Doctors and patients can start by creating an account and entering basic information like their username and password. Users returning can safely log in. The system confirms user identity with a strong authentication procedure, ensuring allowed access to SkinScan features.

4.2. SkinScan Interface

SkinScan interface offers skin cancer screening services, including scanning body areas, viewing results, learning about forms, taking pictures, uploading, and providing details on different types of skin cancer.

4.3. Selecting Body Part

Users select a body area for skin cancer analysis using their device's camera. They can capture or upload images, which are preprocessed for analysis. These images are then imported into the SkinScan program for further examination.

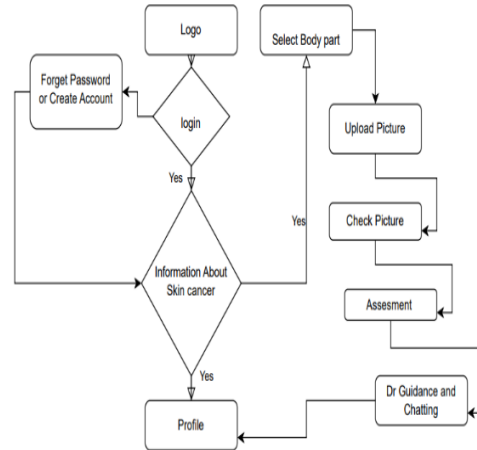


Figure 2: SkinScan Flow Diagram

4.4. View Result

Users obtain the results of a skin scan, which may include details regarding anomalies found, potential forms of cancer of the skin, and suggested courses of action. To help users comprehend scan results, the system may indicate areas of concern on uploaded photos using remarks or visualizations and the flow diagram can be seen in Figure 2. This figure explains the functionality of the overall proposed system for skin cancer.

4.5 Connecting Patients and Doctors

The application's chat feature facilitates easy communication between doctors and patients, promoting a patient-centered approach to healthcare delivery, enhancing access, and empowering patients. Patients can easily communicate with the doctor for further guidance.

4.6. Model Architecture

In our investigation, we utilized an enhanced version of the Swin Transformer architecture. The process of fine-tuning involves retraining a pre-existing model on a specific dataset or task, thereby enhancing its effectiveness for that particular objective.

The model classifies various dermatological conditions, including actinic keratoses, basal cell carcinoma, benign keratosis-like lesions, dermatofibroma, melanocytic nevi, melanoma, and vascular lesions which we can also see in figure 4, aiding early diagnosis and treatment planning.

4.7. Architecture of Swin Transformers

With a linear computing complexity resulting from self-attention processing within each local window, the Swin Transformer is a Vision Transformer that may be used for image identification and classification tasks. It builds hierarchical feature maps by combining picture areas in layers whose classification can be seen in Figure 3.

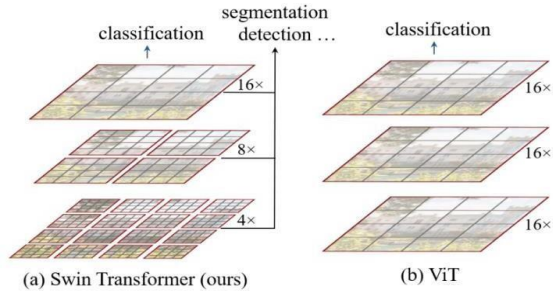


Figure3: Swin Transformer

4.8 DataSet

The HAM10000 ("Human Against Machine with 10000 training images") dataset, consisting of 10,015 images, aims to address the scarcity of dermatoscopic images for the automated diagnosis of pigmented skin lesions. It covers diagnostic categories like Actinic keratoses, Basal cell carcinoma, benign keratosis-like lesions, dermatofibroma, melanoma, and vascular lesions which can be seen in figure 4 with all types of skin cancer.

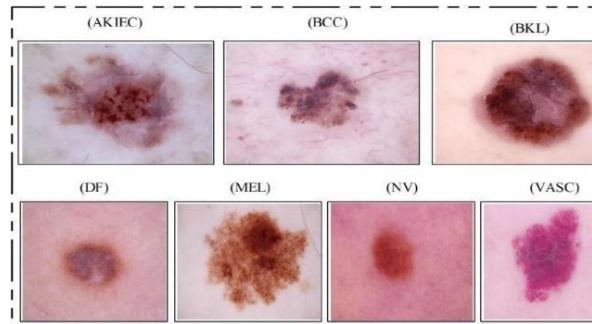


Figure 4: Types of Skin Cancer

4.9. Training Procedure

We carefully chose and adjusted several hyperparameters during the training process to maximize the effectiveness of our skin cancer classification model

4.10 Training Hyperparameters

1. Learning Rate:

The number of steps taken during optimization to update the model's parameters is determined by the learning rate which is 5e-05 in this case.

$$\text{New Weight} = \text{Old Weight} - \text{Learning Rate} \times \text{Gradient} \quad (1)$$

2. Train Batch Size:

The number of training examples used in each training cycle is determined by the batch size. The train batch size of this model is 32.

3. Eval Batch Size:

It determines the number of examples utilized for evaluation in each iteration, much as the train batch size. The Eval batch size of this model is 32.

4. Seed

For reproducibility, the random number generator is initialized with a seed in Figure 4, which is a random number. The seed in this model is 42.

5. Gradient Accumulation Steps:

In advance of changing the model's parameters, gradient accumulation permits gradients to build across several batches. The value is 4.

6. Total Train Batch Size:

The product of train batch size and gradient accumulation steps yields the effective batch size that is used for parameter updates. The total train batch size is 128.

$$\text{Train Batch Size} \times \text{Gradient Accumulation Steps} = \text{Total Train Batch Size} \quad (2)$$

Table 1: Training Results & Accuracy of Swin Transformer model

Training Result & Accuracy of Model	
Training Loss	0.3261
Epoch	1.0
Step	3.30
Validation Loss	0.2744
Accuracy	0.8984
Precision	0.9030
F1	0.8985
AUC-Roc Curve	0.9935

7. Optimizer:

The Adam optimizer, with betas = (0.9, 0.999) and epsilon=1e-08, uses adaptive methods to adjust learning rates and momentum, ensuring stability during parameter updates.

8. *Learning Rate Scheduler Type:* A linear learning rate scheduler adjusts the learning rate linearly during training epochs.

9. Learning Rate Scheduler Warmup Ratio:

The value of 0.1 indicates that **10%** of the epochs are allocated for the learning rate warmup.

10. Number of Epochs:

The model is trained for one epoch, completing one pass through the entire training dataset.

In our proposed system the training loss after one epoch is 0.3261, while the validation loss is 0.2744 as shown in table 1. The training method achieved an accuracy of 89.84%, precision of 90.30%, F1 score of 89.85%, and Area Under the Curve (AUC ROC) of 99.35%. Additionally, it implies that the value at the given stage is 3.30 as given in Table 1. These indicators collectively reflect the model's performance and efficacy after the stipulated training period, providing information about its capacity to make accurate predictions and manage the given data.

4.11. Working

The application features a logo screen, login screen, and password reset options. It raises consumer awareness about skin cancer and provides access to important information. Users can create accounts, customize profiles, and upload pictures for skin assessment. They can preview submitted photos before starting the assessment via the Check Picture screen in figure 2.

The SkinScan program offers a user-friendly interface with distinct functions, allowing users to create accounts, access instructional materials, upload photos, and receive detailed skin scan findings after processing supplied photographs.

V. RESULTS

Presenting SkinScan, a cutting-edge AI-powered software for prevention and early detection of skin cancer. We have reached significant goals through thorough research and testing, proving the app's performance in precisely detecting possible skin problems. The artificial intelligence model utilized in SkinScan has exhibited remarkable efficiency, attaining elevated levels of precision in identifying skin cancer in various datasets. The measures for accuracy, recall, and F1-score consistently show how reliable our model is in differentiating between various kinds of skin lesions. Early feedback from users and engagement data demonstrate how wellliked SkinScan's functionality and user experience is. Users value the app's ability to raise awareness and its simplicity of use, as well as its clear and instructional content regarding skin cancer. One of the main concerns with SkinScan has been its picture

processing pipeline efficiency which can be seen in Figures 5 and 6, we have provided it with images, and it has shown us the results. Our findings demonstrate that the program offers quick analysis, guaranteeing a smooth user experience. The app's usefulness for daily use is enhanced by its real-time processing capabilities.

The image was classified using a model, predicting it to be a Melanoma or Melanocytic-nevi, with lower confidence values for other classifications, aiding in skin lesion diagnosis in Figures 5 and 6, based on the precision and accuracy results we conclude that this is the desired type of cancer.



Figure 5: Skin Cancer Classification Output

VI. DISCUSSION

There was a 32% rise in reports for (pre)malignant lesions of the skin among users of the app in comparison to a comparable number of those who avoided using the mHealth app. However, a three- to four-fold increased likelihood of claims for benign skin cancers and nevi among mHealth users also offset this effect. Based on the previously published diagnostic precision of the analyzed app [15,16], these results were anticipated. Additionally, they align with other highly recognized population-based cancer screening programs that strike a balance between accurately diagnosing malignancies and producing false positive results [17], as well as contemporary clinical dermatological practice, which excises about 8 nevi for every melanoma [18].



Figure 6: Classification Results for Skin Cancer

Using a mHealth app could be an option even though traditional skin cancer screening according to a full body check by a qualified healthcare provider is not advised [19]. a stage in between to think about focused screening for high-risk lesions. According to this study, using the app was associated with a rise in (pre)malignancy claims; as a result, it may be a useful first step in enhancing skin cancer detection. But as it stands, the app can now detect any cutaneous (pre)malignancies, including actinic keratosis, keratinocytic carcinomas, and melanomas. The morbidities and fatalities of these (pre)malignancies vary greatly [20–23], and early diagnosis of cutaneous premalignancies such as actinic keratosis is medically less significant. The incorrect diagnosis and hence inefficient use of limited healthcare resources could be a major drawback of implementing these kinds of apps across the population [24].

VII. CONCLUSION

The SkinScan app, using the Swin Transformer model, is a groundbreaking solution for early skin cancer detection and prevention. With an accuracy rate of 0.89, it facilitates user-friendly skin examinations, enabling timely medical interventions and potentially life-saving treatments. SkinScan also raises awareness and promotes proactive health behaviors, potentially improving patient outcomes and reshaping skin cancer management.

REFERENCES

[1]. Schreuder, K., de Groot, J., Hollestein, L. M., Louwman, M. huidkankerrapport IKNL, <https://iknl.nl/nieuws/2019/steedsvaker-huidkanker,-nationaal-plan-nodig> (2019).

[2]. Tokez, S., Hollestein, L., Louwman, M., Nijsten, T. & Wakkee, M. Incidence of multiple vs first cutaneous squamous cell carcinoma on a nationwide scale and estimation of future incidences of cutaneous squamous cell carcinoma. *JAMA Dermatol.* 156, 1300–1306 (2020).

[3]. Lomas, A., Leonardi-Bee, J. & Bath-Hextall, F. A systematic review of worldwide incidence of nonmelanoma skin cancer. *Br. J. Dermatol.* 166, 1069–1080 (2012).

[4]. Chen, S. T., Geller, A. C. & Tsao, H. Update on the epidemiology of melanoma. *Curr. Dermatol. Rep.* 2, 24–34 (2013).

[5]. Janda, M. & Soyer, H. P. Can clinical decision making be enhanced by artificial intelligence? *Br. J. Dermatol.* 180, 247–248 (2019).

[6]. Tschandl, P. et al. Comparison of the accuracy of human readers versus machinelearning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study. *Lancet Oncol.* 20, 938–947 (2019).

[7]. Esteva, A. et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542, 115–118 (2017).

[8]. Haenssle, H. A. et al. Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Ann. Oncol.* 29, 1836–1842 (2018).

[9]. Freeman, K. et al. Algorithm based smartphone apps to assess risk of skin cancer in adults: systematic review of diagnostic accuracy studies. *BMJ* 368, m127 (2020).

[10]. CZ. SkinVision reimbursement CZ, <https://www.cz.nl/vergoedingen/skinvision> (2023).

[11]. Rajpurkar, P., Chen, E., Banerjee, O. & Topol, E. J. AI in health and medicine. *Nat. Med.* 28, 31–38 (2022).

[12]. Tahir, M., Naeem, A., Malik, H., Tanveer, J., Naqvi, R. A., & Lee, S.-W. (2023). DSCC_Net: Multi-Classification Deep Learning Models for Diagnosing of Skin Cancer Using Dermoscopic Images.

[13]. Mijwil, M. M. (2021). Skin cancer disease images classification using deep learning solutions.

[14]. Muhammad Zia Ur Rehman, Fawad Ahmed, Suliman A. Alsuhibany, Sajjad Shaukat Jamal, Muhammad Zulfiqar Ali, & Jawad Ahmad. (Year). Classification of Skin Cancer Lesions Using Explainable Deep Learning.

- [15]. Udrea, A. et al. Accuracy of a smartphone application for triage of skin lesions based on machine learning algorithms. *J. Eur. Acad. Dermatol. Venereol.* 34, 648–655 (2020).
- [16]. Sangers, T. et al. Validation of a market-approved artificial intelligence mobile health app for skin cancer screening: a prospective multicenter diagnostic accuracy study. *Dermatology*, 1–8, <https://doi.org/10.1159/000520474> (2022).
- [17]. Taksler, G. B., Keating, N. L. & Rothberg, M. B. Implications of false-positive results for future cancer screenings. *Cancer* 124, 2390–2398 (2018).
- [18]. Nelson, K. C., Swetter, S. M., Saboda, K., Chen, S. C. & Curiel-Lewandrowski, C. Evaluation of the number-needed-tobiopsy metric for the diagnosis of cutaneous melanoma: a systematic review and meta-analysis. *JAMA Dermatol.* 155, 1167–1174 (2019).
- [19]. Johansson, M., Brodersen, J., Gotzsche, P. C. & Jorgensen, K. J. Screening for reducing morbidity and mortality in malignant melanoma. *Cochrane Database Syst. Rev.* 6, CD012352 (2019).
- [20]. Adamson, A. S., Suarez, E. A. & Welch, H. G. Estimating overdiagnosis of melanoma using trends among black and white patients in the US. *JAMA Dermatol.* <https://doi.org/10.1001/jamadermatol.2022.0139> (2022).
- [21]. Boniol, M., Autier, P. & Gandini, S. Melanoma mortality following skin cancer screening in Germany. *BMJ Open* 5, e008158 (2015).
- [22]. Stang, A. & Jockel, K. H. Does skin cancer screening save lives? A detailed analysis of mortality time trends in SchleswigHolstein and Germany. *Cancer* 122, 432–437 (2016).
- [23]. Welch, H. G., Mazer, B. L. & Adamson, A. S. The rapid rise in cutaneous melanoma diagnoses. *N. Engl. J. Med.* 384, 72–79 (2021).
- [24]. Adamson, A. S. & Welch, H. G. Machine learning and the cancer-diagnosis problem - no gold standard. *N. Engl. J. Med.* 381, 2285–2287 (2019).

Combatting Illegal Logging with AI Powered IoT Devices for Forest Monitoring

Dr. Nasru Minallah
*Department of Computer System
 Engineering*
*University of Engineering and Technology,
 Peshawar Pakistan*
n.minallah@uetpeshawar.edu.pk

Abdullah Khan
*Department of Computer
 System Engineering*
*University of Engineering and
 Technology, Peshawar Pakistan*
20pwce1916@uetpeshawar.edu.pk

Maham Jadoon
*Department of Computer
 System Engineering*
*University of Engineering and
 Technology, Peshawar Pakistan*
20pwce1875@uetpeshawar.edu.pk

Abstract — This research presents a comprehensive strategy for tackling illegal logging by leveraging Artificial Intelligence (AI) and Internet of Things (IoT) technologies. In high-risk forestry areas, sensors-equipped Internet of Things devices are used to continuously monitor and detect sound of surrounding. The AI component uses machine learning methods to identify potential unlawful logging activities by accurately detecting and distinguishing sound patterns associated with chainsaw and logging operations such as tree cutting and also detect the natural disaster like the wildfire. When such activities are detected by these smart AI powered IoT device installed in forest, real-time notifications are generated after such activity which allowing surrounding enforcement agencies, such as the forest department, to intervene promptly. By providing a targeted and prompt solution to the issue of illicit logging, this strategy supports biodiversity preservation and sustainable forest management.

Keywords: Forest Monitoring; AI Against Illegal Logging; Real-time Alerts; Environmental Conservation; IoT for Anti-Logging

I. INTRODUCTION

Forests are woven in our history. It is the source of fresh air, strong materials for houses, fresh water, and fertile soil for growing crops. But forests are being challenged. Illegal logging driven by uncontrolled greed for money challenges them. It throws the nature's balance off give and take. The results are damaging to our ecosystems causing disasters like habitat losses as well as social entity indifferences. Deforestation initiates climate imbalance messing up the weather regime as well as disturbs the process of equilibrium in the ecosystem. All this careless act performed risks the same resources for its gain.

Big-time cutting of the trees combined with illegal chopping of the wood provisions to the problems in saving worldwide vine variance, keeping equal ecology, and protecting homes of countless wild beings. This destroys the system of world waters, reduces nature distinction by destruction of home and

creates a lot of conflicts and other effects on the social matters. Alarmingly, every year an expanse of country-sized land is afflicted by deforestation either from illegal logging or fires, especially in growing countries like Brazil and Indonesia.

The records of vast forests have always been a challenge in the history books. Walking through dense woods is risky, requires numbers of people, and usually leaves out far places. Regular surveillance like cameras can help, but they don't capture everything and can miss things as an action is taking place. The quiet noises in the forest, often covered by leaves and distance, can slip on by. This allows lasting damage to happen before we can step in.

What if tech could boost these soft sounds, turning them into warnings of unfairness and signals of hope for our vanishing woods? Imagine hearing - really hearing - the hidden messages in the moving leaves? This is the big dream driving our project. We try to appeal to the cutting edge of Internet of Things (IoT) and voice classification power. We want to stand guards for peaceful green defenders.

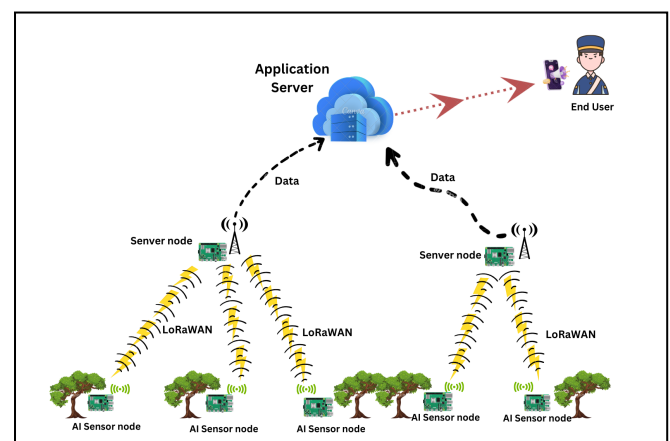


Figure 1: Diagram of Proposed Project.

Now let's imagine intelligent sensors scattered in a forest, as shown in figure 1. They're run by the chip ESP32, a powerful, widely-used engine. Their job is to beckon and learn, with advanced tech that recognizes voices. They're always on the lookout for injurious things like noisy chainsaws, falling logs,

or people talking up to no good. Before, these damaging noises would go unnoticed. Now they are translated into useful information. Straightaway, this information fires through the unseen paths of the IoT to a main cloud base.

The soft sounds would rise into a loud noise in the woods, slowly but surely. Alerts go off immediately, leading forest rangers exactly where the action was improper. Now, law breakers who prowl and come unseen are seen; their quiet deeds demand responsibility loudly and effect action, reverberating in the digital world.

II. AIMS AND OBJECTIVES

Illegal logging is a pressing threat to global forests, imperiling their sustainability. A solution leveraging artificial intelligence (AI) and Internet of Things (IoT) devices is proposed to combat this issue. The project aims to reduce unauthorized logging through real-time monitoring, deploying IoT devices strategically. AI algorithms will analyze data, identifying patterns and potential logging incidents.

Preserving biodiversity is a key focus, preventing deforestation from illegal logging. The project enables data-driven decision-making for forest management, emphasizing collaboration among stakeholders. Capacity building initiatives prioritize enhancing local communities' capabilities in monitoring and combating illegal logging.

Awareness initiatives highlight the impacts of illegal logging, emphasizing forest conservation. Specific objectives include deploying IoT devices, implementing AI-powered analysis, and fostering collaboration. Advocacy for supportive policies is integral to combat illegal logging.

Community engagement involves training local communities in monitoring and protecting forests. Ongoing evaluation ensures the effectiveness of AI-powered IoT devices in combating illegal logging.

III. LITERATURE REVIEW

In the world of audio classification, which has made great strides since the last decades of the 20th century, applications have been found for distance learning and digitized libraries. The new generation methodologies that have been pioneered include Hidden Markov Models by Lu Jian with its intricate sorting of audio files and also label-free approach by Zhao Xueyan. Li, S.Z provides a refined sound identification system using Support Vector Machines (SVMs) and Mel

Frequency Cepstral Coefficients (MFCC) in the domain of sound understanding.

Features from the time and frequency domain form the focus area of the study for sorting the sounds. Using Short-Time Average ZCR and Short-Time Energy in the time domain, the important features for sound or silence determination are extracted on the paper. In the frequency domain, there exist repeated features like Centroid of Audio Frequency Spectrum and Sub-Band Energy Ratio that give useful pointers. The idea of the suggested method of sound sorting is grounded in SVM, and thus it requires the careful preprocessing of audio samples and many features to be extract, including MFCC. The system proved effectual as it was tested for a mixed set of 2500 sound pieces and obtained an average commendable accuracy of 92.14%. In conclusion, this article reveals some insight into the historical context of audio sorting, a deep sound's sorting system mixing time and frequency traits, augmented with SVM processing.

Tiny Machine Learning (TinyML) challenges the problem of pitching machine learning into microcontrollers integration in domains such as sound recognition, image classification and motion classification/anomaly detection. In sound recognition, most powerful keyword spotting (KWS) takes the center stage whereby individuals gain hands-free access control for better user accessibility. In projects that depict exemplary TinyML application, such as the project with XIAO nRF52840 Sense, implementation of TinyML applications as feasible is demonstrated even in cases of limited resources. Regarding other projects, TinyML has application in image classification where XIAO ESP32S3 Sense additionally supports camera access and therefore can be used in projects related to object recognition too. TinyML enhanced by an EnsensoPico for motion classification/anomaly detection enabled the real-time analysis of sensor data and has been exemplified in an innovative cargo damage prevention tool utilizing XIAO nRF52840 Sense. These, as well as those purpose-developed in this area, illustrate the transforming potential of TinyML hence empowering low-power embedded systems across various sectors.

We provide audio classification in this literature review, from understanding basic processes towards application of advanced machine learning models. It focuses voice conversion into quality digital format, stress on analysis through spectrogram and other tool selection like Mel-spectrogram and Short- Time Fourier Transform (STFT). Within the study, the loudness differences are normalized using techniques, and ideas of Data Augmentation and also the use of mechanisms such as Depthwise Separable

Convolutional Neural Networks (DS-CNN) and FastGRNN are presented to improve on the overall capabilities of listening to audio tasks.

This investigation into Enhanced Sound Recognition (ESR) with the application of Deep Learning (DL) techniques underscores the emergent interest to leverage DL in enhancing Machine Learning (ML) models. This study will review the different approaches towards sound recognition from spectrograms/MFCCs as inputs to ANN/CNN classifiers and raw waveform directly. In one of the most illustrative experiments, sound classifying in a Wireless Sensor Network (WSN) with Raspberry Pi (RPi) nodes is performed using a Convolutional Neural Network (CNN). The research has proven that even feature extraction and sound classification can be executed on embedded high-level devices, so it highly accentuates searching computational capabilities.

Addressing the dearth in discourse of the benefits and the feasibility for the introduction to machine learning techniques of sound classification. Other proposed solutions for the Internet of Things (IoT) are specially designed models like CNN, with separate focus on model inference time and resource efficiency on low-power microcontrollers. Hardware accelerators such as Tensor Processing Units (TPUs) and Field Programmable Gate Arrays (FPGAs) are exploited to enhance the performance of audio applications of Deep Learning. Lastly, the study concludes in a better approach for looking damage identification and location in multiple classes by considering the various level of severity of damages as well as for scenarios. These techniques along with the windowing data augmentation and a unique majority voting technique accompanied with the global CNN-1D model prove to be adequate for dealing with the problems of limited data.

IV. METHODOLOGY

The methodology consists of a series of related steps from the data collection to the real-world deployment designed to develop an effective system to fight against the issue of illegal logging with the help of the AI enabled IoT devices. In our purposed methodology the cutting-edge technology is used by utilization the low power microcontroller and make it the powerful tool with the use of Machine learning. Like in the first step through the data collection and it preprocessing before train the supposed model.

A. Data Collection and Preprocessing

First, a rich dataset comprising 75 samples each of the following audio classes - Silence, Axe, Chainsaw, and Fire has been collected from

different website. Each audio recorded voices has 5 seconds and the total data collected is shown in Table [1]. The process involved in data collection uses Mel-filter bank energy features that are very important in audio classification. These features provide vital information about environmental sounds and extract the speech signals for use in recognition tasks like associated with illegal logging.

TABLE 1. Data Collection

<i>Data collected</i>	<i>Train / Test split</i>
25m 0s	79% / 21%

B. Training Deep Learning Model

In this step, the convolutional neural network model (CNN) is trained and selecting filter-bank energy features (MEF) as a feature to detect the surrounding voices and make the meaningful insight form recorded voices. In Step 1, obtain a balanced dataset where each class would contain at least 75 samples: Silence, Axe, Chainsaw. The audio samples consisted of files with durations of exactly 5 seconds each. This dataset is important in training a strong classifier for the accurate classification of environmental sounds associated with the activities of illegal logging.

1) *Mel-filter bank Energy Features (MEF)*: The Mel-filter bank energy features (MEF) have been used as the choice feature for this audio classification. The method of the extraction of the features used in this work is to capture the frequency content of the audio signals. This process can be achieved by using a filter bank to divide this audio signal into several narrow frequency bands. The resulting energy values across these bands serve as essential input features for the machine learning model.

Parameters for MEF are listed below

- i. Frame Length: 0.02 seconds
- ii. Frame Stride: 0.01 seconds
- iii. Filter Number: 40
- iv. FFT Length: 256
- v. Low Frequency: 0

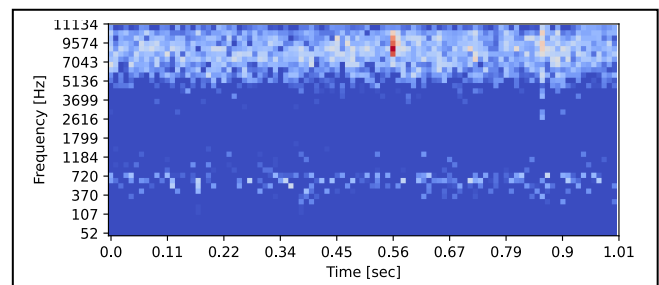


Figure 2: Mel Energies (DSP Output)

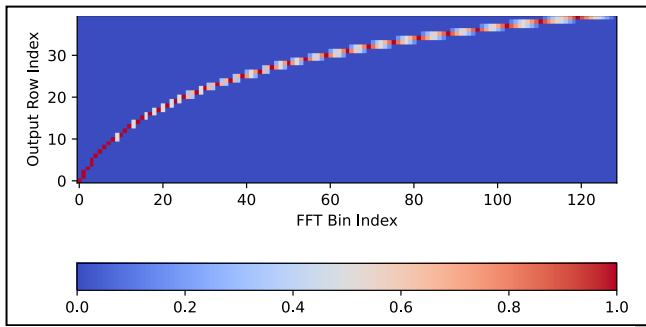


Figure 3: FFT Bin Weighting

The figure 2 and 3 illustrated the Mel Eng Mel Energies and FFT Bin Weighting of some of voice from data set. These parameters should be chosen very carefully in such a way as to optimize the extraction of Mel filter bank energy features from the audio data, such that ultimately a set of very representative and informative features is obtained for the purpose of training the deep learning models.

2) *Neural Network Architecture*: The neural network architecture was designed to effectively process the Mel- filter bank energy features classifying the audio into their corresponding classes. The architecture includes:

- i. **Input Layer (3,960 features)**: The model reshapes the features so that it can be processed by the model.
- ii. **Reshape Layer (40 columns)**: 40 filters that are part of the MEF.
- iii. **1D Conv/Pool Layer (8 neurons, 3 kernel size, 1 layer)**: First incidence of convoluting.
- iv. **Dropout (Rate 0.25)**: Introduces regularization to prevent overfitting.
- v. **1D Conv/Pool Layer (16 neurons, 3 kernel size, 1 layer)**: Extra convolutions for better feature extraction.
- vi. **Dropout (Rate 0.25)**: Further regularization.
- vii. **Flatten Layer**: It converts output into a one-dimensional array/tensor for next series of layers.
- viii. **Additional Layer**: Introduces complexity and abstraction for improved model performance.
- ix. **Output Layer (4 classes)**: Represents the four classes (Silence, Axe, Chainsaw, Fire).

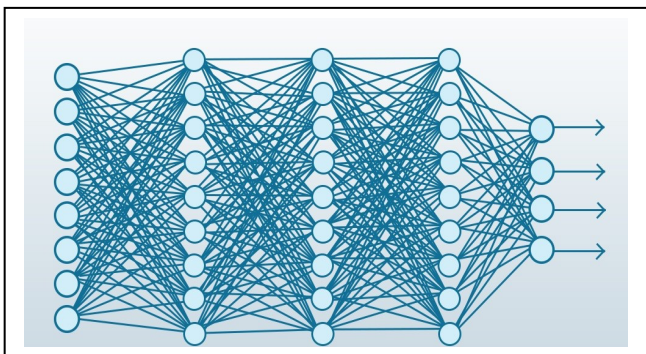


Figure 4: Model Hidden layers Structure of CNN

Table 2: Model Accuracy on Training set

<i>Accuracy</i>	<i>Loss</i>
96.2%	0.10

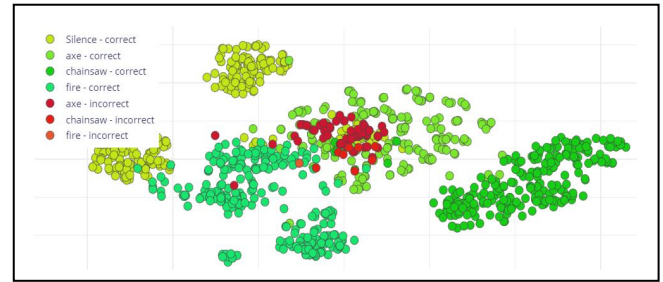


Figure 5: Data Explorer

Training Settings

- i. **Number of Training Cycles**: 100
- ii. **Learning Rate**: 0.005

In processing that, it is possible to realize the consensus and accuracy during the training process to see that the network learns effectively the Mel-filterbank energy features. The figure 5 below is showing the model training accuracy which also shows the incorrect classification with red points.

C. Integration Model with ESP32 Microcontroller as a Sensor Node

Using the trained model which we have develop in previous section, the model is exported as a arduino and deployed to the ESP32(ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth) microcontroller functioning as a sensor node. For recording of audio data of surrounding, an enabled INMP441 microphone has been used which The INMP441 is a high-performance, low power, digital-output, omnidirectional MEMS microphone. The recorded audio data, therefore undergoes processing by use on device machine learning model. This integration ensures a classification of sounds which falls into categorization related to the activities of illegal logging by ESP32 autonomously.

Table 3: On-device Performance (ESP32 board)

<i>Interfacing Time</i>	<i>Peak RAM usage</i>	<i>Flash Usage</i>
21ms	10.6Kb	32.6Kb

In table 3, on device performance is illustrated. On ESP32 SoC. The latter uses LoRa communication for a secure transmission of the voice identified, and node location (longitude, latitude). In this scenario, the ESP32 being a node sensor, sends its data, using LoRa technology to the central server node which is another ESP32 and work as server as its is handling the database of that project. This

way of communication is needed to establish networking abilities over wide forested areas so that monitoring can occur in real time.

D. Server Node (ESP32) and Database Integration

In this step, ESP32 works as the server node and is to receive signals from the ESP32 sensor nodes. The server node connects Firebase which a cloud platform by Google. Firebase is used for real-time data storage, updating node locations and other relevant information that exists in the field of operations. This integration ensures a centralized and accessible repository of data.

The mobile application is built to interface with the Firebase database and give real-time notifications on detected activities and node locations. It is the user interface for the app, where different users like members of the local community, law enforcement agencies, as well as environmental groups interact.

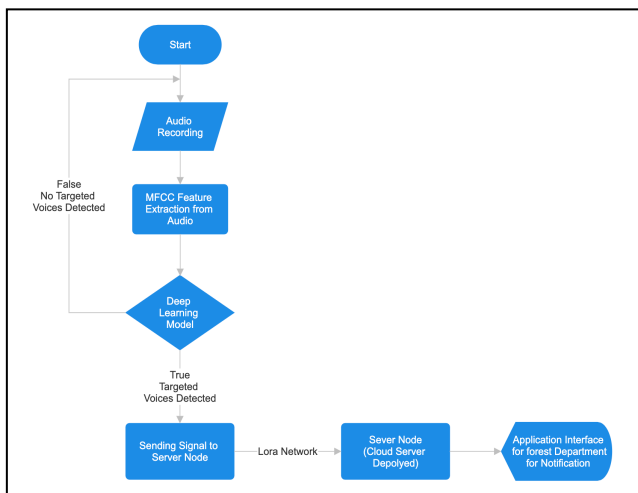


Figure 6: Flowchart to project

Figure 6 is illustrating the project top overview and clear flow of our project. This is the real world-based scenario which is used to test how good the model is that has trained on several localization and classification tasks in a real-world case. The testing phase also tests whether LoRa communication is robust as well as ensuring data moves from LoRa to Firebase successfully.

V. RESULTS

The model performed exceptionally well in various aspects. It achieved an accuracy of 96.2% on the validation set along with low loss, that is, 0.10. The confusion matrix reflected accurate different class ability of the model, high percentages for SILENCE, AXE, CHAINSAW and FIRE. The corresponding efficiency in terms of F1 scores was further

validated through table below.

	<i>Silence</i>	<i>Axe</i>	<i>Chainsaw</i>	<i>Fire</i>	<i>Uncertainty</i>
<i>Silence</i>	73.5%	9.8%	0%	0%	16.7%
<i>Axe</i>	1.8%	70.2%	0%	4.0%	24.1%
<i>Chainsaw</i>	0%	2.9%	95.2%	0%	1.9%
<i>Fire</i>	0%	1.2%	0%	98.4%	0.4%
<i>F1 score</i>	0.84	0.77	0.98	0.96	

Table 4: Accuracy 82.57%

During on-device performance, the model showed efficient inferencing that took very short of 21 ms. Minimal resource usage was identified by peak RAM at 10.6K and flash usage at 32.6K as already shown in table 3. This makes for deployment in real-time constrained devices.

Table 5: Model's Efficiency in utilizing

	<i>MFE</i>	<i>Classifier</i>	<i>Total</i>
<i>Latency</i>	450ms	21ms	471ms
<i>RAM</i>	20.4K	10.6K	20.4K
<i>Flash</i>		32.6K	
<i>Accuracy</i>			

In each of the testing scenarios, the model maintained a high level of accuracy at 82.57%. The performance behavior of the classification algorithm in classifying into groups the audio samples under the different classes including instances of uncertainty was depicted by the confusion matrix. The F1 scores for each class also confirmed the reliability of the model under different setup conditions during testing.

The latencies distribution during the inference process of the model display as shown in table 5 a well-distributed figure, with an overall of 471 ms made up of 450 ms that are contributed by the Mel-filterbank energy features (MFE), another 21 ms for the classifier. Resource usage in terms of RAM and FLASH stood at 20.4K and 32.6K respectively unveiling further efficacy of the model in memory allocation as well storage.

Overall, the results underscore strong generalization ability by the model showing it as being promising in a real-world setting combating illegal logging with its effective audio classification on IoT devices.

VI. CONCLUSION

The application of Internet of Things (IoT) and Artificial Intelligence (AI) technology offers an achievable approach for combating illegal logging, which is a major danger to world forests and biodiversity. We have demonstrated an

innovative strategy for real-time monitoring and detecting actions related to illegal logging, such as chainsaw running and tree cutting down, through the deployment of smart sensors installed with AI algorithms.

By harnessing advanced machine learning techniques which include convolutional neural networks (CNNs) trained on Mel-filter bank energy features (MEF), the developed model attained an spectacular accuracy of 96.2% on the validation set. The model efficient inferencing process with a minimal resource usage on low-powered microcontrollers like the ESP32 which spectacle its suitability for real-world deployment in forest environments.

Integration with IoT devices like ESP32 microcontroller serving as a sensor node which enables autonomous classification of environmental noises related to illegal logging activities. Employing LoRa communication technology, data transmission to a centralized server node deployed with Firebase database make sure the realtime data storage and accessibility.

The model's overall accuracy of 82.57% as well as successful memory allocation and memory resource usage have been confirmed by the testing scenarios' results, which also highlight the model's robustness and generalization capacity. The model is a useful tool for monitoring and preventing illicit logging in real-world forest situations because of its low-latency inference process and ability to reliably classify audio samples into distinct classifications.

In conclusion, our research demonstrates the effectiveness of AI-powered IoT devices in preserving biodiversity and promoting sustainable forest management and also safeguarding against the dangerous effects of illegal logging. By providing timely detection and intervention capabilities so that this strategy contributes to the conservation of forests, ecosystems, and the a lot of benefits they provide to humanity and the planet as a whole.

REFERENCES

- [1] Anjanappa, C., Parameshwara, S., Vishwanath, M.K., Shrimali, M. and Ashwini, C. (2022). AI and IoT based Garbage classification for the smart city using ESP32 cam. *International journal of health sciences*, pp.4575–4585. <https://sciencescholar.us/journal/index.php/ijhs/article/view/6905>
- [2] Yang, X., Xing, H. and Su, X. (2022). AI-based sound source localization system with higher accuracy. *Future Generation Computer Systems*. <https://linkinghub.elsevier.com/retrieve/pii/S0167739X22003387>
- [3] Ali, M., Young Sig Kwon, Lee, C.-H., Kim, J. and Kim, Y. (2015). *Current Approaches in Applied Artificial Intelligence*. Springer.
- [4] Ahmad, I., Shaqiff Emir Shariffudin, Aizat Faiz Ramli, Mohamad, M., Mansor, Z. and Khan, S. (2021). *Intelligent Plant Monitoring System Via IoT and Fuzzy System*. <https://ieeexplore.ieee.org/document/9526312/>
- [5] Merenda, M., Porcaro, C. and Iero, D. (2020). Edge Machine Learning for AI-Enabled IoT Devices: A Review. *Sensors*, 20(9), p.2533. <https://www.mdpi.com/1424-8220/20/9/2533>
- [6] Calo, S.B., Touna, M., Verma, D.C. and Cullen, A. (2017). Edge computing architecture for applying AI to IoT. *2017 IEEE International Conference on Big Data (Big Data)*. <https://ieeexplore.ieee.org/document/8258272>
- [7] Roggen, D., Cobden, R., Arash Pouryazdan and Muhammad Zeeshan (2022). *Wearable FPGA Platform for Accelerated DSP and AI Applications*. *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. <https://ieeexplore.ieee.org/document/9767398>
- [8] edgeimpulse.com. (n.d.). Edge Impulse. [online] Available at: <https://edgeimpulse.com/>
- [9] Espressif (2023). ESP32 Overview | Espressif Systems. [online] www.espressif.com. Available at: <https://www.espressif.com/en/products/socs/esp32>
- [10] Chatterjee, R., Mazumdar, S., Sherratt, R.S., Halder, R., Maitra, T. and Giri, D. (2021). Real-Time Speech Emotion Analysis for Smart Home Assistants. *IEEE Transactions on Consumer Electronics*, 67(1), pp.68–76. <https://ieeexplore.ieee.org/document/9352018>
- [11] Madikeri, S. R., & Murthy, H. A. (2011). Mel Filter Bank energy-based Slope feature and its application to speaker recognition. <https://ieeexplore.ieee.org/document/5734713>

Exploring Hardware Security: Ring Oscillator Based PUF Implementation on Xilinx 7 Series FPGAs

Habib Ullah

Computer Systems Engineering
University of Engineering and Technology
Peshawar, Pakistan
habib.ullah@uetpeshawar.edu.pk

Bilal Habib

Computer Systems Engineering
University of Engineering and Technology
Peshawar, Pakistan
bhabib@uetpeshawar.edu.pk

Waleed Khan

Computer Systems Engineering
University of Engineering and Technology
Peshawar, Pakistan
khanwaleed247@uetpeshawar.edu.pk

Abstract—In recent years, hardware security has become increasingly critical in various applications ranging from IoT devices to critical infrastructure. Physically Unclonable Functions (PUFs) have emerged as a promising solution for providing unique hardware identifiers and enhancing device security. Among different implementations of PUFs, ring oscillator (RO) based PUFs offer simplicity and effectiveness. This paper presents the implementation and characterization of a RO-based PUF on Xilinx 7 Series FPGAs. The design methodology, implementation details, and characterization results are discussed to evaluate the performance and security of the proposed solution. The findings demonstrate the feasibility and effectiveness of using RO-based PUFs for hardware security on Xilinx 7 Series FPGAs.

Keywords: Hardware Security; Ring Oscillator (RO); Frequency Variation; Challenge-Response Protocol.

I. INTRODUCTION

In an era marked by increasing concerns over cybersecurity threats and the integrity of digital systems, the demand for robust hardware security mechanisms has never been greater. The proliferation of connected devices, ranging from Internet of Things (IoT) gadgets to critical infrastructure components, underscores the need for reliable means of device authentication and data protection. Physically Unclonable Functions (PUFs) have emerged as a promising solution to address these challenges by leveraging the inherent variations in hardware properties to generate unique device identifiers. Among the various implementations of PUFs, ring oscillator (RO) based designs offer simplicity, efficiency, and

compatibility with FPGA architectures. This paper explores the implementation and characterization of a RO-based PUF on Xilinx 7 Series FPGAs, aiming to provide insights into its performance, security, and suitability for hardware security applications.

All the Xilinx ZYNQ Architectures are divided into PL and PS as shown in Figure 1. PL stands for Programmable Logic, which is an FPGA. In a ZYNQ device, the PS or processing system (cortex a9) is connected to the programmable logic (PL) using the AXI buses. That means peripherals implemented in PL can communicate easily with software running on the CPU.

II. LITERATURE REVIEW

The concept of Physically Unclonable Functions was first introduced by Pappu et al. [1], who proposed the use of optical scattering properties of random materials for generating unique identifiers. Since then, PUFs have been extensively studied and applied in various domains, including device authentication, secure key generation, and anti-counterfeiting measures. Among the different implementations of PUFs, RO-based designs have gained attention due to their simplicity and effectiveness. Gassend et al. [2] introduced the concept of silicon physical random functions, demonstrating the feasibility of using ring oscillators for generating unique device identifiers. Subsequent research by Suh and Devadas [3] further explored the application of PUFs for device authentication and secret key generation, highlighting their advantages over traditional cryptographic methods. Maes and Verbauwhede [4] conducted a comprehensive study on the state of the art and future research directions in physically unclonable functions, emphasizing the need for robust and reliable hardware security mechanisms.

In the context of FPGA-based security, Xilinx 7 Series FPGAs have gained popularity for their high-performance, low-power, and versatile architecture. The inherent reconfigurability of FPGAs makes them ideal platforms for implementing hardware security solutions, including PUFs. However, limited research has been conducted on the implementation and characterization of PUFs specifically targeting Xilinx 7 Series FPGAs. This paper aims to bridge this gap by presenting a detailed exploration of RO-based PUF implementation on Xilinx 7 Series FPGAs and evaluating its performance and security characteristics.

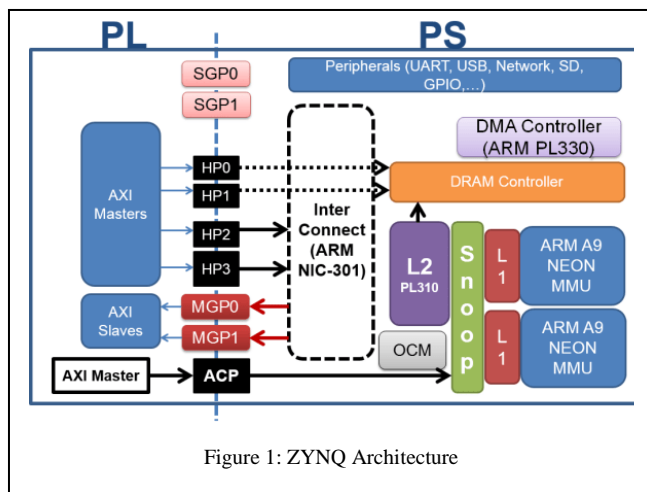


Figure 1: ZYNQ Architecture

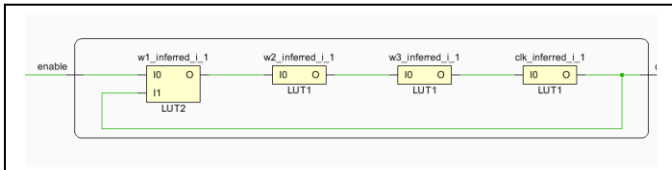


Figure 2: Ring Oscillator after Implementation in Vivado

III. METHODOLOGY

An RTL design is implemented in Programming Logic (PL) and interfaced with the ZYNQ processing system (PS) using AXI interfaces. The top module comprises of a Datapath and a Controller.

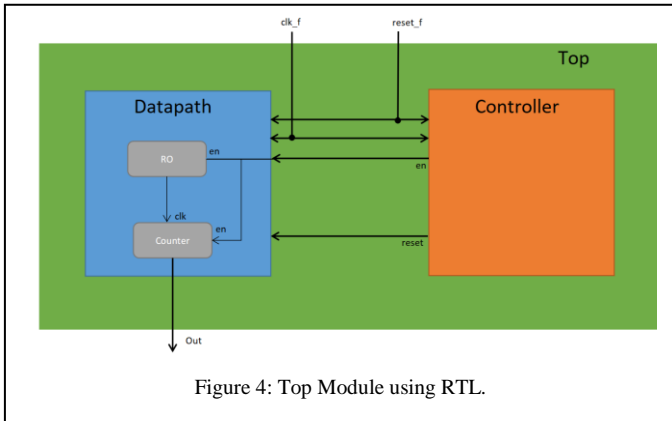


Figure 4: Top Module using RTL.

The design is implemented in Vivado 17.4 and tested on a Digilent Zybo featuring an XC7Z010 BGA SoC which includes an Artix-7 FPGA and two hardcore ARM Cortex A9 processors.

Inside the Datapath 256 Ring Oscillators (RO) are instantiated which each RO drive a counter as shown in Figure 3 and Figure 4. Only one RO runs at a time and the controller resets the counter after reading another RO.

The Controller is a Moore FSM. Which enables the RO for 0.5ms using a 100MHz Clock. So, for a period of 10ns, it keeps incrementing a 16-bit counter and see if it reached 50,000. If it is, it set enable = 0 meaning that 0.5ms has passed. For the next RO the Controller resets the counter.

As shown in Figure 2, Constraints are applied on the RO so that the Vivado tools do not optimize the ring oscillators into a single LUT, otherwise it would simply defeat the purpose of an RO.

Five Instances (D1, D2, D3, D4, D5) of Datapath are instantiated into the top module out of which only the output of one is selected as the counter value using 5x1 Multiplexer. The counter will give 256 values for a single Datapath as it has 256 ROs. The External signal sel which is 8 bits wide, is used to select each RO out of 256 ROs in each Datapath.

The methodology for implementing and characterizing the RO-based PUF on Xilinx 7 Series FPGAs involves several key steps:

A. Design and Synthesis

The RTL design of the RO-based PUF is developed using hardware description languages (HDLs) such as Verilog or VHDL. The design is synthesized using Xilinx Vivado Design Suite to generate the corresponding FPGA configuration bitstream.

B. Implementation

The synthesized design is mapped onto the target Xilinx 7

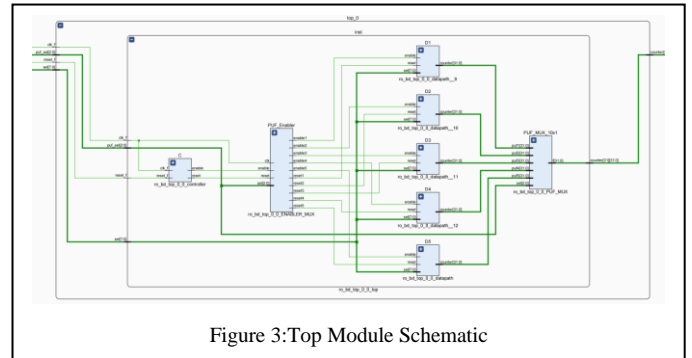


Figure 3: Top Module Schematic

Series FPGA device, considering resource utilization, timing constraints, and routing considerations. Careful attention is paid to optimizing the design for the target FPGA architecture to ensure efficient utilization of hardware resources.

C. Verification

The implemented design undergoes rigorous functional verification to ensure correctness and compliance with the intended specifications. Simulation-based verification techniques are employed to validate the functionality of the RO-based PUF under various operating conditions and input stimuli.

D. Testbench

Using a testbench to simulate the proposed design, the counter gets the value 50000 after running an RO for 0.5ms (500,000ns) as shown in Figure 5.

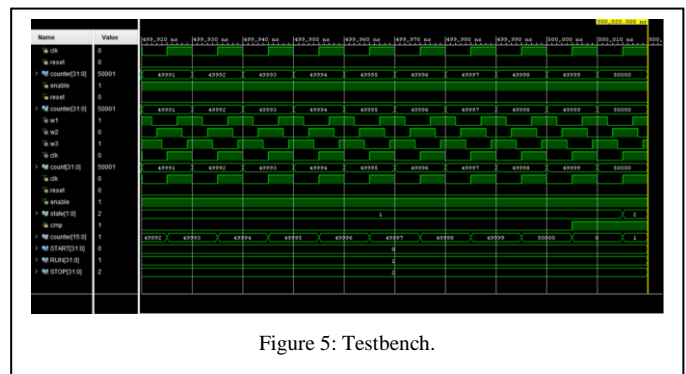


Figure 5: Testbench.

E. Placement and Routing

As shown in Figure 6, the placement of all ROs is constraint into to SLICEL slices only and the rest of the parts of the block design are placed by the tool mostly in SLICEM slices as we have used all the SLICEL.

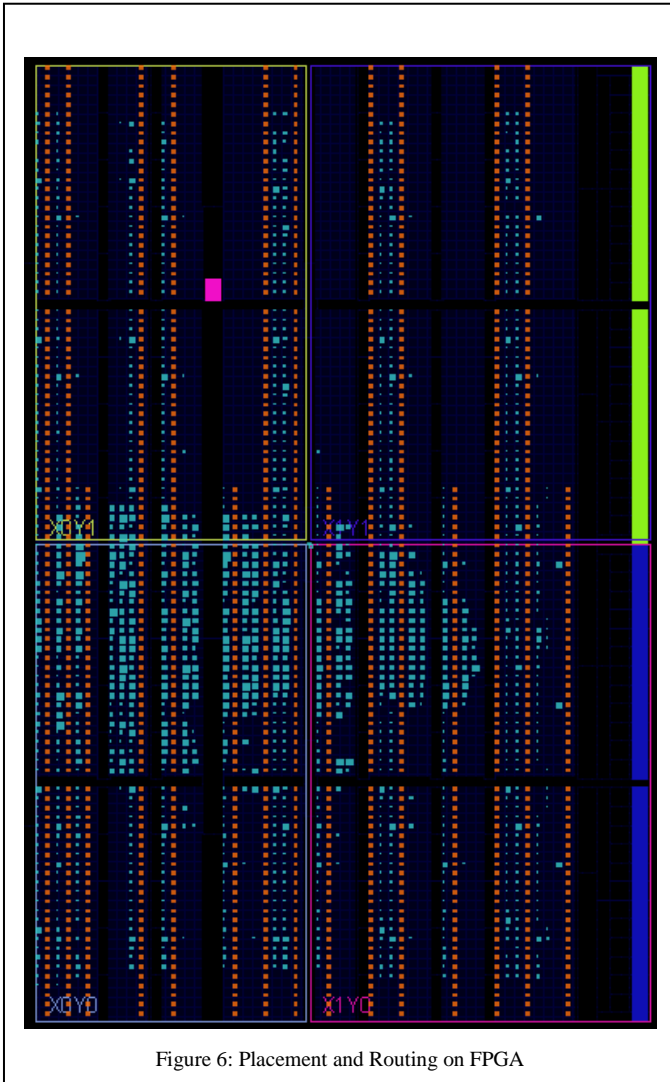


Figure 6: Placement and Routing on FPGA

F. Characterization

Once the design is verified, a series of characterization tests are conducted to evaluate its performance and security properties. These tests include frequency measurements, stability analysis, and challenge-response evaluations, as described in the previous section.

By following this methodology, the performance and security of the implemented RO-based PUF on Xilinx 7 Series FPGAs can be thoroughly evaluated, providing valuable insights into its suitability for hardware security applications.

IV. RESULTS AND DISCUSSION

The characterization results demonstrate the effectiveness of the implemented RO-based PUF on Xilinx 7 Series FPGAs. The frequency measurements reveal significant variations among individual ring oscillators, ensuring the uniqueness of the response patterns. Stability analysis indicates consistent performance of the PUF across different operating conditions, highlighting its reliability. Challenge-response evaluations confirm the unpredictability and security of the generated

identifiers, making the PUF suitable for secure authentication and key generation applications.

V. CONCLUSION

In conclusion, this paper presents a comprehensive exploration of hardware security using ring oscillator based PUFs on Xilinx 7 Series FPGAs. The implementation and characterization results highlight the feasibility and effectiveness of leveraging PUFs for enhancing device security. Future research directions may include optimizing the PUF design for specific performance metrics and exploring advanced security features for FPGA-based systems.

ACKNOWLEDGMENTS

I extend my heartfelt gratitude to the Almighty for granting me the strength, confidence, and patience to pursue my academic endeavors. I am immensely thankful to Dr. Bilal Habib for his invaluable supervision and guidance in Embedded Systems, FPGAs, and Hardware Security. His expertise significantly contributed to enhancing my skills and preparing me to face future challenges.

I would like to express my appreciation to the Xilinx community forum for enriching my understanding and problem-solving capabilities in FPGA-related topics. The insights and discussions provided invaluable assistance in the implementation of my work.

I am grateful to my family members and parents for their unwavering support throughout my academic journey. Lastly, I extend my thanks to UET Peshawar, particularly the NCBC Lab and the Computer Systems Engineering department, for providing a conducive environment for learning and growth.

REFERENCES

- [1] Pappu, R., Recht, B., Taylor, J., & Gershenfeld, N. (2002). Physical one-way functions. *Science*, 297(5589), 2026-2030.
- [2] Gassend, B., Clarke, D., van Dijk, M., & Devadas, S. (2002). Silicon physical random functions. In *Proceedings of the 9th ACM conference on Computer and communications security* (pp. 148-160).
- [3] Suh, G., & Devadas, S. (2007). Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual Design Automation Conference* (pp. 9-14).
- [4] *International Conference on Field Programmable Logic and Applications*, 2011, pp. 151-156. doi: 10.1109/FPL.2011.35.
- [5] [8] A. Lotfy, M. Kaveh, D. Martín, and M. R. Mosavi, "An Efficient Design of Anderson PUF by Utilization of the Xilinx Primitives in the SLICEM," *IEEE Access*, vol. 9, pp. 23025-23034, 2021, doi: 10.1109/ACCESS.2021.3056291.
- [6] [9] B. Habib, J.-P. Kaps, and K. Gaj, "Efficient SR-Latch PUF," in *Applied Reconfigurable Computing*, K. Sano, D. Soudris, M. Hübner, and P. C. Diniz, Eds., Cham: Springer International Publishing, 2015, pp. 205-216.
- [7] [10] B. Habib, K. Gaj, and J.-P. Kaps, "FPGA PUF Based on Programmable LUT Delays," in *2013 Euromicro Conference on Digital System Design*, 2013, pp. 697-704. doi: 10.1109/DSD.2013.79.
- [8] [11] S. Tian, A. Krzywosw, I. Giechaskiel, and J. Szefer, "Cloud FPGA Security with RO-Based Primitives," in *2020 International Conference on Field-Programmable Technology (ICFPT)*, 2020, pp. 154-158. doi: 10.1109/ICFPT51103.2020.00029.
- [9] [12] K. Dayananda and N. Karimian, "When Physical Unclonable Function Meets Biometrics." 2020.
- [10] [13] Y. Hu, Y. Jiang, and W. Wang, "Compact FPGA Ring Oscillator Physical Unclonable Functions Circuits Based on Intertwined

- Programmable Delay Paths,” Prepr. <https://doi.org/10.36227/techrxiv>, vol. 14214401, p. v2, 2021.
- [11] [14] A. Aghaie and A. Moradi, “Inconsistency of simulation and practice in delay-based strong PUFs,” *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, pp. 520–551, 2021.
- [12] [15] M. Fakroon, F. Gebali, and M. Mamun, “Multifactor authentication scheme using physically unclonable functions,” *Internet Things*, vol. 13, p. 100343, 2021.
- [16] Bilal Habib, “GMU PUF Scripts.” Cryptographic Engineering Research Group. [Online]. Available: <https://cryptography.gmu.edu/puf/>

GAMEBRAINS: NPCs Intelligence Using Neural Network Brains

Mosaddiq Billah
Computer Systems Engineering
University of Engineering and
Technology
Peshawar, Pakistan
20pwcse1863@uetpeshawar.edu.pk

Ahmed Bahar
Computer Systems Engineering
University of Engineering and
Technology
Peshawar, Pakistan
20pwcse1884@uetpeshawar.edu.pk

Anoora Seher
Computer Systems Engineering
University of Engineering and Technology
Peshawar, Pakistan
20pwcse1869@uetpeshawar.edu.pk

Engr. Dr. Muniba Ashfaq
Computer Systems Engineering
University of Engineering and
Technology
Peshawar, Pakistan
muniba@uetpeshawar.edu.pk

Engr. Abdullah Hamid
Computer Systems Engineering
University of Engineering and
Technology
Peshawar, Pakistan
abdullahhamid@uetpeshawar.edu.pk

Abstract— This paper aims to develop the foundational knowledge about the Unity game development engine embedded with AI for the development of a hypercasual game that has intelligent NPCs, which operate strategically in the environment. The targeted audience comes in the class of those who are pursuing their career in the niche of game development. Moreover, in this paper, different reinforcement learning methods have been discussed, which have been implemented in the game that produce the optimal results for the behavior of NPCs. AI games are the future of this industry to enhance the gaming experience for the users and provide a multiplayer effect for them. Hence, this paper tends to represent a glimpse into the future perspective of the gaming industry in hypercasual gaming platforms.

Keywords: Unity, Reinforcement Learning, AI Games, NPCs, and agents.

I. INTRODUCTION

Unity Engine is a cross-platform game development developed by Unity Technologies that supports 2D games, 3D games, Augmented Reality applications, and Virtual Reality applications. It was released in 2005. The engine extended to support a variety of devices such as desktops, mobiles, and different console devices. It works on both iOS and Android devices. It is considered a beginner-friendly application for game developers who want to pursue their career in the domain of game development, where most indie game developers or independent game developers are mostly found using this engine. Furthermore, this game engine is utilized by different organizations beyond game development, such as engineering, architecture, and film [1].

This paper presents the integration of AI into Game Development, which is a glimpse into the future of the industry of game development. However, this paper tends to introduce AI to the platforms of hypercasual games, since AI is mostly found in high-level games such as Red Dead Redemption 2. Our paper aims to develop a hypercasual game, which is a lightweight game with fewer mechanics

that would have intelligent NPCs. The NPCs in the game would operate strategically in the game, hence enhancing the gaming experience for the users by providing them with a multiplayer game effect through the intelligent NPCs.

A. Research Paper Aim

To develop a foundation of understanding of Unity being integrated with AI for the development of AI games that will uplift the gaming industry. The following objectives and queries are developed that will achieve this aim.

B. Research Paper Objectives

- To study Unity and AI for the development of innovative AI hypercasual games.
- To examine the best optimal Reinforcement Learning methods for the agents.
- To outline the integration of AI with Unity.

C. Research Paper Questions

- What are the benefits of AI games?
- To what extent AI games can shape the gaming industry?
- Which Reinforcement Learning methods are optimal for the game agents to operate intelligently in the game?

The research paper has been divided into five sections in order to achieve the aim and goals. Section 1 elaborates the introduction, Section 2 covers the Literature Review,

Section 3 covers the methodology, Section 4 covers the discussion and Section 5 covers the conclusion.

II. LITERATURE REVIEW

The literature review, in this section of the proposed paper, examines the works that have been conducted previously.

Unity is a cross-platform game engine, used for 2D and 3D games. It supports a variety of platforms such as desktops and mobiles. Furthermore, Unity is one of the most popular engines among the other game development engines due to its flexibility, efficiency, and convenience. The gaming engine comprises various tools that are convenient tools for modifying your project. The feature of real-time play mode with smart previews allows you to monitor the modifications instantly [1]. Unity engine supports the deployment of its projects on different operating systems such as Mac, Linux, and Windows, along with artist-friendly tools by allowing developers to develop efficient games [3].

Additionally, the navigation system in Unity enables NPCs to logically move around the environment of the game [4]. However, this feature only limits the movement of the NPCs or agents to make decisions in terms of movement around obstacles in the environment.

On the scripting aspect of the Unity engine, it supports C#, which is an object-oriented programming language developed by Microsoft. There are two ways to design C# scripts in the Unity engine. The first one is object-oriented design, which is referred to as the traditional approach and is used by the majority of developers. The second is data-oriented design, which is also supported by Unity [5].

Therefore, based on the studies covered in the literature review, a conceptual framework is shown below (See Fig 1).

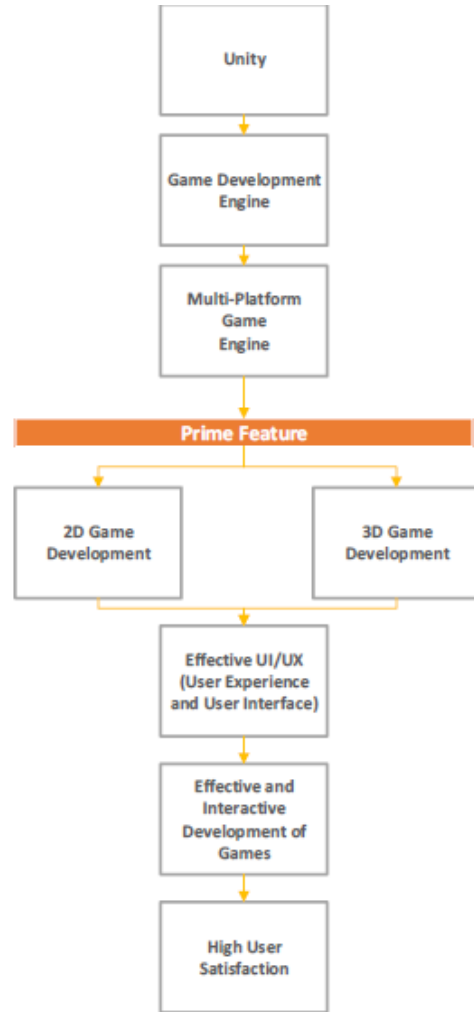


Figure 1 - Conceptual Framework

III. METHODOLOGY

The following methodology is adopted in this paper in which data is gathered from different sources, which relates to the domain of this paper. The collection of data for this work has been gathered from the following tools:

1. Unity Engine
2. C#
3. Python (Anaconda Software)

A. Training Environment

The environment was created for the hypercasual game to test the reinforcement methods. The environment is based on a football game where two teams play against each other by scoring the ball into the located goals. The important feature used in the game is “ray cast” which collects data. Raycast is used to detect the object from a distance by the agent to make intelligent decisions based on the values collected. The agent receives the reward after completing their desired objective of scoring a goal.



Figure 2 – Hypercasual Game Design

B. Reinforcement learning methods

Multiple RL methods exist in this case for the purpose of training these agents, whereas the following RL methods were implemented in order to attain the aim of leading it to an AI game where the agents or NPCs operate intelligently. Furthermore, the training of the neural network was implemented parallel to enhance the training process.

The following methods were utilized for the agents:

1. Proximal Policy Optimization (PPO)
2. Random network distillation
3. Behavior Cloning (BC)
4. Curiosity-driven exploration by self-supervised prediction
5. Generative Adversarial Imitation Learning (GAIL)

C. Collecting training data

Statistics were collected from the feature of ray-cast, which is a component in Unity that detects objects from a distance when it is implemented on an object. It retains information of an object through an invisible laser, which is visible in the editor mode of Unity Engine.

D. Agent's Behavior Parameters

In this game, the NPCs or agents were assigned with various ray-casts lasers for detecting various objects;

Agent or NPCs, for observation space, have been assigned with 11 ray-casts forward and 3 ray-casts backward in order to collect data for the training model in order to take intelligent decisions through reinforcement learning.

IV. RESULT & DISCUSSION

In this section of the proposed paper, the research objectives have been discussed. Clear themes have been identified to assist in discussing the research goals effectively in order to attain the main aim of the proposed paper.

A. To study Unity and AI for the development of innovative AI hypercasual games.

Unity 3D offers several features, which has not been discussed yet. Unity supports asset tracking, scripting, processing and physics which eases the development time for developers for their projects. According to a paper [6], Unity engine supports 27 diverse platforms and devices in an environment that is an user-friendly development.

Regarding to AI in the game industry, it is a broad topic. AI comprises of machine learning and decision making abilities specifically for the behavior of agents in the game. It is a significant aspect in contemporary game development. Developing an AI game can have an immersive impact on game play [7].

The first AI game, called the “Spacewar!”, was developed at MIT in the early 1960s [7]. It was a multiplayer space combat genre game that had AI-controlled opponents. Nevertheless, AI games evolved over the years since then for the purpose of enhancing the gaming experience. During those days, PCs did not exist as large mainframes or computers were mostly found in institutions.

In recent times, large gaming organizations have started to implement AI into their games such as Red Dead Redemption 2, which is a popular game. But, it must be taken into notice that such games are mostly high level and requires specific consoles or upgraded PCs.

In this case, hypercasual games, such as Candy Crush game or Subway Surfers, are known for their simple features and mechanics, convenient gameplay and minimum design, enabling to be accessible to a broad audience of gaming genre.

To convey AI games to a broad audience of such genre, hypercasual games are mostly played by people. By introducing AI to the platform of such games, it can uplift the gaming industry to a greater extent.

B. To examine the best optimal Reinforcement Learning methods for the agents.

According to a paper, there are various training tools for the evaluation of reinforcement learning algorithms [8]. There is a reinforcement learning algorithm known as the NEAT or NeuroEvolution of Augmenting Topologies. NEAT has various applications in several domains, including game development. However, the shortcoming of NEAT, which is compared to the reinforcement learning methods implemented in this paper, is that it does not inherit a memory system. It focuses on evolving the structure and linkage of neural networks instead of having a mechanism for remembering information over period.

However in this paper as mentioned above, the following reinforcement learning methods have been utilized for the training agents in the game environment:

a. Proximal Policy Optimization (PPO)

The default core method is the Proximal Policy Optimization (PPO) reinforcement algorithm, which was first presented by [9]. By limiting policy updates to a limited range of $[1 - \epsilon, 1 + \epsilon]$, PPO is renowned for its stable and user-friendly architecture. This prevents huge policy changes from gradient ascent, which can be damaging. Deviations of $rt(\theta)$ from 1 in policy changes are penalized by the objective function.

Mathematically, the function of PPO is represented by;

$$L^{\text{CLIP}}(\theta) = \hat{E}_t [rt(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t]$$

- ϵ is a hyperparameter, (usually 0.1 or 0.2).
- \hat{E}_t is the expectation over timesteps.
- rt is the probability ratio of new and old policies.
- \hat{A}_t is the estimated advantage at time t .
- θ is the policy parameter.

The Proximal Policy Optimization (PPO) reinforcement algorithm is a stable and user-friendly method that is ideal for training agents in game environments since it makes use of the clipped objective function.

b. Generative Adversarial Imitation Learning (GAIL)

Among model-free imitation algorithms, Generative Adversarial Imitation Learning (GAIL) is unique in that it performs better than previous model-free techniques, particularly in complex contexts [10]. This approach depends more on learning from the environment and requires more contact with it than model-based approaches because it is model-free. The method is ad hoc action exploration to find out which activities lead to a policy shift in favor of the expert's policy.

With GAIL, agents can learn behavior without explicitly knowing the reinforcement signal by watching the expert's activities. The expert records and saves its actions in a demonstration file with associated states and actions. With this framework, behavior may be directly extracted from the given example.

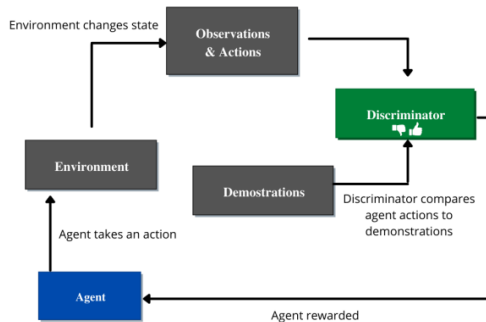


Figure 3: Generative Adversarial Imitation Learning

c. Curiosity-driven exploration by self-supervised prediction

According to [11], the reports have observed that in real-world settings, rewards are frequently insufficient or nonexistent. In such situations, agents might be encouraged to explore their surroundings and learn new abilities that are necessary to accomplish their goals by using their curiosity as a motivator. Here, the agent's curiosity is defined as its ability to predict the results of its actions.

According to [11], the reports have explained that curiosity-driven exploration encourages agents to explore more effectively, lowering the requirement for significant contact with the environment to accomplish objectives. This method also allows the agent to forecast how its actions will turn out in scenarios that haven't been witnessed before. One notable benefit of curiosity-driven exploration is its capacity to motivate agents to discover new places in games.

d. Behavior Cloning (BC)

The need to accurately mimic human behavior in intelligent entities is addressed by behavior cloning, a concept Hussein [12] introduced. This method allows behavior-cloned agents to mimic human behavior in similar circumstances. Autonomous vehicles, helper robots, and computer-human communication are just a few of the domains in which it finds use. As a basic aspect of behavior cloning, imitation learning depends on insights from the environment and the agent's actions.

Imitation learning has intrinsic limitations, despite its usefulness. It is contingent upon the availability of expert demonstrations of a high caliber, and the agent's competence is limited to what people can demonstrate. Cloning behavior is very useful for agents that need to closely mimic the examples given. One disadvantage of such agents is that they are not able to improvise when choosing what to do.

Behavior cloning is a useful method for giving agents human-like behavior, and it has applications in many different fields. Although it works well for faithful imitation, it has limitations in terms of the caliber of expert demonstrations and the incapacity of agents to perform at levels higher than those of humans.

e. Random network distillation (RND)

Burda [13] introduced Random Network Distillation, which modifies the reinforcement learning techniques by adding an exploration incentive. The bonus is ascertained by calculating the prediction error of a neural network (NN) feature acquired from observations using a fixed, randomly initialized NN. This strategy serves as a directed exploration tool with the primary goal of addressing the problem of scant rewards. Essentially, the agent is rewarded for discovering new things when interacting with the

environment or for venturing into locations that haven't been explored before.

Setting itself apart from a lot of curiosity-based approaches, the RND methodology shows persistence in the face of obstacles like becoming stuck when subjected to random noise, like static on a TV screen. The method's dependence on an exploration bonus instead of absolute prediction error accounts for this robustness. As a result, RND shows great promise as an algorithm, especially for agents whose job it is to navigate complicated environments where the observation data obtained is highly noisy.

In reinforcement learning, Random Network Distillation presents a fresh viewpoint on exploration and efficiently handles situations with plenty of noise and scant rewards. Because of its unique methodology, it is positioned as a viable algorithm for agents exploring complex and noisy scenarios.

With the training of each algorithm, PPO gave a positive response with higher accuracy in terms of reinforcement learning, where the agents strategically operated throughout the game. Other algorithms gave a response, while PPO turned out to have higher accuracy of 95% as compared to the other RL algorithms, which were less in terms of their accuracy and response by the agents in the environment.

```

trainer_type: ppo
hyperparameters:
  batch_size: 128
  buffer_size: 2048
  learning_rate: 0.0003
  beta: 0.01
  epsilon: 0.2
  lambda: 0.95
  num_epoch: 3
  learning_rate_schedule: linear
  beta_schedule: linear
  epsilon_schedule: linear
network_settings:
  normalize: False
  hidden_units: 256
  num_layers: 2
  vis_encode_type: simple
  memory: None
  goal_conditioning_type: hyper
  deterministic: False
reward_signals:
  extrinsic:
    gamma: 0.99
    strength: 1.0
  network_settings:
    normalize: False
    hidden_units: 128
    num_layers: 2
    vis_encode_type: simple
    memory: None
    goal_conditioning_type: hyper
    deterministic: False
init_path: None
keep_checkpoints: 5
checkpoint_interval: 50000
max_steps: 2000000
time_horizon: 64
summary_freq: 60000
threaded: False
self_play: None
behavioral_cloning: None
  
```

Figure 4: Training Behaviour of PPO algorithm in the Soccer Game using Anaconda Prompt integrated with the Unity Engine's file project

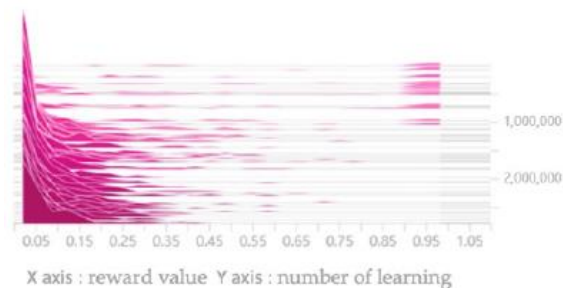


Figure 5: Cumulative-reward-value graph using the PPO algorithm

C. To examine the integration of AI with Unity

Unity cross-platform has the feature of supporting integration with other applications. Although, it has the library of AI that supports the Navigation Mesh, as discussed in the above section, which is for agents to move intelligently in the environment by avoiding obstacles. Unity contains the API that allows the integration of Python with the agents in the environment by assigning them the algorithm depending upon the purpose.

The Unity Machine Learning API package in Unity enables the developers to integrate Python with Unity Editor in order to train the behaviors of agents (See Figure Below) [14]. The API package contains of the following three components. The components are agent, python API and trainer. The agent is the character model in the game to which the model has been assigned. It communicates the training model through the Python API via a communicator in Unity. Hence, the agent utilizes the communicator in order to connect with the trainers through the Python API.

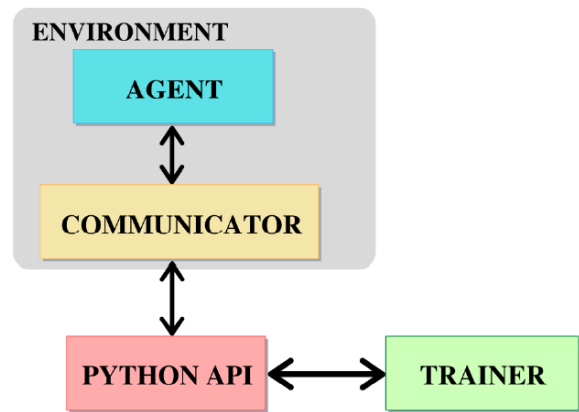


Figure 6: API Framework Model

V. CONCLUSION

To conclude, the paper seeks to enhance the gaming experience for users by using hypercasual game platforms with intelligent NPCs or agents. With the discussed methodology and discussion, the algorithm utilized for the game is PPO due to its stability and policy improvement. Furthermore, it can be observed that introducing AI to hypercasual games will not only uplift the game industry but encourage other independent developers to enter the domain of game development due to its modern perks of introducing AI with optimal reinforcement algorithms. Three objectives were formulated to attain these goals such as to study Unity and AI for the development of innovative AI hypercasual games, to examine the best optimal Reinforcement Learning methods for the agents, and to outline the integration of AI with Unity. Associated with these objectives, three questions were also formulated for the same purpose of attaining these goals what are the benefits of AI games? To what extent AI games can shape the gaming industry?

Which Reinforcement Learning methods are optimal for the game agents to operate intelligently in the game?

Hence, AI games are the future of the gaming industry and it has been rising since the 1960s with the development of the "SpaceWars!" game to contemporary games like Red Dead Redemption 2. It will start appearing as a common feature in every game in the future.

REFERENCES

- [1] Juliani, Arthur, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. "Unity: A general platform for intelligent agents." arXiv preprint arXiv:1809.02627 (2018).
- [2] Haas, John K. "A history of the unity game engine." (2014).
- [3] Canossa, Alessandro. "Interview with nicholas francis and thomas hagen from unity technologies." In-Game Analytics, pp. 137-142. Springer, London, 2013.
- [4] Becker-Asano, Christian, Felix Ruzzoli, Christoph Hölscher, and Bernhard Nebel. "A multi-agent system based on unity 4 for virtual perception and wayfinding." *Transportation Research Procedia* 2 (2014): 452-455.
- [5] Malete, Thabo N., Kabo Moruti, Tsaone Swaabow Thapelo, and Rodrigo S. Jamisola. "EEG-based Control of a 3D Game Using 14-channel Emotiv EPOC+." In 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), pp. 463-468. IEEE, 2019.
- [6] Craighead, Jeff, Jennifer Burke, and Robin Murphy. "Using the unity game engine to develop sarge: a case study." In *Proceedings of the 2008 Simulation Workshop at the International Conference on Intelligent Robots and Systems (IROS 2008)*. 2008.
- [7] Wexler, J. (2002). *Artificial intelligence in games*. Rochester: University of Rochester.
- [8] Juliani, A., Berges, V. P., Teng, E., Cohen, A., Harper, J., Elion, C., ... & Lange, D. (2018). Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627.
- [9] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.
- [10] Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. arXiv preprint arXiv:1606.03476.
- [11] Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. 34th International Conference on Machine Learning, ICML 2017, 6, 4261–4270.
- [12] Hussein, A., Gaber, M. M., Elyan, E., Jayne, C., & Gaber, M. M. (2016). A Imitation Learning: A Survey of Learning Methods A:2 A. Hussein et al. *ACM Computing Surveys*, V, 1–35.
- [13] Burda, Y., Edwards, H., Storkey, A., & Klimov, O. (2018). Exploration by random network distillation. ArXiv, 1–17.
- [14] Juliani, A., Berges, V. P., Teng, E., Cohen, A., Harper, J., Elion, C., ... & Lange, D. (2018). Unity: A general platform for intelligent agents. arXiv preprint arXiv:1809.02627.

Artificial Intelligence Based Seed Germination, Prediction, and Quality Assessment

Engr. Shahzad Khan
Computer Systems Engineering
University of Engineering & Technology
Peshawar, Pakistan.
15pwcse1342@uetpeshawar.edu.pk

Dr. Nasru Minallah
Computer Systems Engineering
University of Engineering &
Technology
Peshawar, Pakistan.
15pwcse1342@uetpeshawar.edu.pk

Abstract— Testing seeds germination is a big challenge to the seed companies because till now we are only able to germinate seeds or test seeds germination by using a manual method of seed germinator machine which takes approximately one week for 50% of seeds germination. So, a lot of time and seeds are wasted because of this manual testing of seeds dormancy and germination so here is the germination testing technology based on Artificial Intelligence, Deep learning and RGB data. Using this technology, we can find the germination so seeds within seconds using seeds embryo images.

Our recommended machine learning-based methods can improve and is used to boast up the seed germination method to check whether a seed can be germinated or not. Different CNN architecture with regional proposal has been trained by using the method of Transfer Learning which automatically review the seeds present in the petri dish and to predict the germination status of different seeds so that the seeds will germinates or not.

Keywords: FSC&RD: Federal Seeds Certification & Registration Department, AUC-ROC: Area Under ROC Curve, LSTM” Long Short-Term Memory.

I. INTRODUCTION

Germination defined as “the creation and then the development of specific biological structures and shape, having ability to grow into a normal plant under specific and favorable conditions”. The importance of seeds began with the dawn of agriculture, about 12000 years ago. The word Germination is a main seeds quality characteristics having direct impact on product yield and quality. The interest in seeds started with the very start of agriculture field. Germination is a complicated process begins with watering and ended with emergence of one part of embryo. Seeds must be assimilated to active its metabolism and germinate under its applicable environmental condition [1]. Seed germination is an important mechanism by which the seed breasts and starts to grow into a new structure and plant. It is an important process for plant easy growth and reproduction, it plays an important role while maintaining balance of the ecosystem.

Seed germination is the primary method of propagating plants. It enables the growth of new plants from the seeds of the parent plant. Germination is essential in agriculture for producing crops. Farmers depend on seed germination to grow food and feed populations. Seed germination is also crucial for ecological restoration. It allows the regeneration of plant

populations in areas that have been damaged by natural or human-made disasters. The germination of seeds contributes to the diversity of plant species, which is essential for maintaining the balance of the ecosystem. Germinating seeds improve soil health by producing roots that hold the soil together, preventing erosion and nutrient depletion.

Changes occur in seeds shape when a seeds became non-germinating due to temperature change and moisture are abnormal growth of embryo mobilization of endosperm storage and weakness of tissues [2]. To remove or reduce the manual steps in seeds testing very modern image analysis technology is used to detect seeds. Seed Vigor Imaging System Processes Red Green Blue pixels values of an image to measure the length of seeds [3]. In contrast and in Deep Learning techniques, specially CNNs is the best technique for image processing [4]. CNNs automatically derive the relevant properties from crude images and applied it to enormous number of image classification problems. CNNs are already applied to check and calculate germination rate of rice seeds [5]. Data enhancement is one of the techniques used for the upgradation of our training data by the method of rotating of data or flipping of the given data or resizing of the given data or the trained pictures. Method like this is mostly useable overcome and to avoid the overfitting which is good in the promotion and performance of classifier [6]. Different Evaluation metrics were used to check the accuracy and results of data models which are already trained. The mostly useable metric for evaluating the efficiency of computational object detection techniques will be Mean Average Precision. This technique will be useful for comparison of different computational object detection techniques instead of their algorithm [7]. Artificial intelligence can be used to check and predict seed germination by checking various data points that effects seed germination, i.e its temperature, humidity, soil type, seed quality, and age. Different algorithms of Artificial Intelligence can check large amounts of data to identify pictures, pattern and different trends that affect seed germination. This analysis can be used to develop such models which can predicate the forecast likelihood of seed germination based on various environmental factors. AI can also be used to analyze seed quality, such as seed viability and vigor, to predict germination rates. This analysis can help ensure that high-quality seeds are used for planting, which can improve germination rates and crop yields. AI can be used to optimize the environmental conditions for seed germination, such as adjusting temperature and humidity levels, to maximize germination rates and crop yields and can also predicate the length of seeds. AI can be used to monitor and predict the effects of environmental stressors on seed germination, such as drought or extreme temperatures. This analysis can help farmers and agricultural researchers develop

strategies to mitigate the effects of these stressors and improve seed germination rates. The purpose of the study is decrease time consumption and to developed improved germination predication.

II. LITERATURE REVIEW

Agriculture and crops are said to be the backbone of a country which play an important role in development of a country. Seeds germinations play a main role in crops increasing and its quality. According to Baskin CC and Baskin JM [8] knowing about how to increase the germination and its methods should provide the success of germination is the main and important category with very amazing consequences in the development of economy and research areas.

Nikolaeva et al. [9] explained the classification of first dormancy methods containing only two types of dormancies: endogenous and exogenous. In the first one for less germination the seed embryo is responsible, on the other hand some seed structures and chemicals are responsible for seed dormancy. According to author [10] by the achievement in Computer Vision Technology, Artificial Intelligence Technology and Computation Power Technology. Several tries are done in past years to consolidate seeds and quality determination by making it automated, harmless, predication of per seed operation rather than per mathematical values for checking seed quality characteristics like seed health and safety checking the physical and genetic purity and verification of seeds and chemical composition and its germination. For detection of seeds, we choice faster R-CNN because according to Ren S, and Girshick R [11] faster R-CNN having larger accuracy and not need of any real-time predication. Day by day the population of the world increase as in 2008 it was 6.7 million and it will reach to population of 8.0 million in 2025. We need large amount of agriculture to tackle the increase in population. There are four major global crops of agriculture which are Wheat, Rice, Maize, and Soyabean. These crops are responsible to provide 43% of dietary energy to global directly.

For seeds germination we have to use seeds germination trays which are kept in open environment. For seeds germination sunlight and water is the necessary object which kept the embryo soft and it is easy to be germinated. An important ecological and agronomic characteristic that determines when plants enter environments is the extent of seed germination. [12]. As a result, it is tightly controlled while both internal and external stimuli that establish the condition of dormancy and the likelihood of germination (measured as the ultimate % of germination).

III. RESEARCH METHODOLOGY

The methodology that we will follow is Computer Vision Annotation Tool (CVAT) for annotating objects. We use open-source libraries of python to implement AI algorithm and Faster R-CNN for the detection of multiple seeds inside in an image. we using flatbed scanner (SVIS) which stands for Seed Vigor Imaging System examines Red Green and Blue (RGB) pixels of scanned images to measure the size of different seeds and also using mean Average Precision (mAP)

metric for analyzing the efficiency of Computational Object Detection techniques.

Image collection

Figure 1 shows dataset model is used for generation of data while tested seeds were placed in the petri dish to capture wide range of different germination images. Inside petri dish a black cloth is placed below the seeds for high contrast to avoid background issues. The black cloth was watered to ensure germination and the petri dish was covered to avoid *evaporation*. All the images were taken in room temperature and the images were taken after every two hours to check the status of germinated seeds.

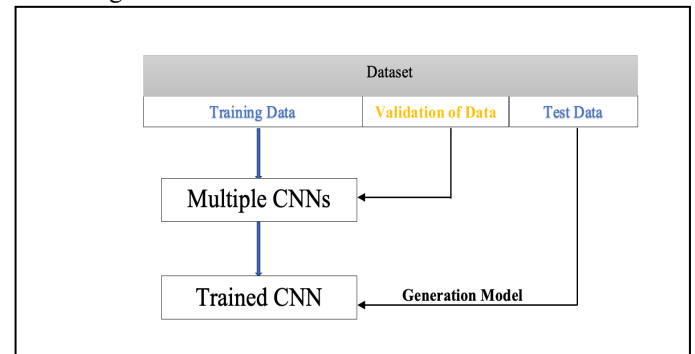


Figure 1: Dataset Generation Model

Data Processing and Object Detection

Open free software CVAT is used for drawing boundaries around each seed and check the status of germination of each one. One is classified as germinated and other is classified as non-germinated. For this we used more than 500 seeds to train it as a germinated and nongerminated classification.

For object detection we used Faster R-CNN to detect multiple seeds in a petri dish. Faster R-CNN have two types of neural networks, Regional Proposal Network and Convolutional Neural Network. RPN is used for Region of Interest where to find the location of seed in an image and CNN is used to distinguished between germinated seed & non-germinated seeds in an image of petri dish. Faster R-CNN is multiple stage algorithm which takes more time but is more accurate than single stage algorithm like YOLO and SSD that's why we choose Faster R-CNN for our work. For reduction of time, we used Transfer Learning which can reduce the computational cost and hence less time consumed to detect the object and to get the results.

Model Training

Below Figure 1 shows the dataset model in which firstly the data were spilt into 80% of trained data, 10% of validated data and 10% of tested data accordingly. To reduce the risk of overfitting we used stratification of data because between 4 to 32 hours the germination status of the seeds might not be changed. We also used data24 augmentation by rotating, flipping and resizing the trained data to reduce the overfitting. We have to trained four types of neural networks and for optimization of hyperparameter we used internal random search. The validation of data was used to select the model

with the best performing hyperparameter pairs. Finally, the best performing models for each of the four networks and for each species were applied, and their performance and generalisation potential were estimated using the never-before-used testing data. Tensorflow's Object Detection API and Python 3.6 were both used to implement each model.

In deep learning, an evaluation metric is a measure used to assess the performance of a machine learning model. It quantifies how well the model is performing on a given task, such as classification, regression, or other types of machine learning problems. Evaluation metrics are essential for comparing different models, fine-tuning hyperparameters, and understanding the model's strengths and weaknesses. The choice of evaluation metric depends on the specific task at hand. Choosing an appropriate evaluation metric is crucial, as it can affect the model development process and the understanding of the model's performance on the target task. For instance, in a binary classification problem with imbalanced classes, accuracy might not be the best metric, and one might focus more on precision, recall, or AUC-ROC to assess the model's performance effectively.

Mean Average Precision (mAP) is a regularly used metric for comparing computational object identification methods, and it may be used to evaluate various computational object detection methods without interacting with the underlying algorithms. When the overlap between the bounding boxes of the prediction and the ground truth reaches a predetermined threshold value, the overlap is assessed using the intersection over union approach and is judged true.

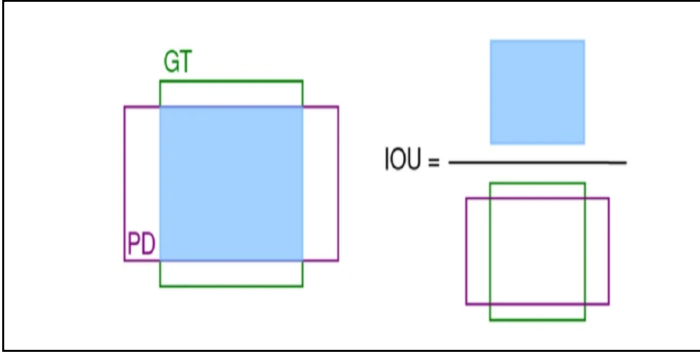


Figure 2: Intersection Over Union

Figure 2 shows intersection over union phoneoma which will be equated as

$$IOU (GT, PD) = \frac{|GT \cap PD|}{|GT \cup PD|} = \frac{\text{Area of the overlap}}{\text{Area of the Union}} \quad (1)$$

Where GT is the ground-truth's bounding box and PD is prediction's bounding box. Now we can measure the final germination percentage which show that at the end of the experiment how many percent of seeds can be germinated. The equation is following.

$$g = \frac{N_g}{N_t} * 100 \quad (2)$$

where N_g is quantity of germinated seeds while N_t is quantity of total seeds which can be used at the start of the experiment. We can also calculate median germination time by using the formula which can calculate the time in which half of the seeds can be germinated i.e 50%. In this work Coolbear is used to calculate median germination time that is:

$$t50 = T_i + \frac{\left(\frac{N+1}{2} - N_i\right) (T_j - T_i)}{N_j - N_i} \quad (3)$$

Here N is the quantity of germinated seeds, N_i & N_j is total quantity of adjacent germinated seeds by the time T_i and T_j . Now we have to find germination uncertainty which estimate the synchronization of germination across all time points measured. Following is the formula to find out the germination uncertainty.

$$U = \sum_{i=1}^K f_i - \log_2 f_i \quad (4)$$

Where f_i is the relevant germination's frequency, i is the time interval & k is the total time interval and if the value of U is equal to zero, it means the germination is synchronized perfectly in each and every time interval having no uncertainty. In this work for performance of deep learning model we performed two experiments. The aim of first experiment to check the prediction of germination and detection abilities of different model of deep learning and therefore we used mAP as a performance metric based on whole test set. The second part of the experiment is germination curves in the test set for ground truth and manual checking of different time intervals which can compare various germination indices.

IV. RESULTS AND DISCUSSIONS

In germination detection and predication part we used Faster Region based Convolutional Neural Network for four pre-trained architectures of Convolutional Neural Networks. On the validation set I selected the best performing architecture.

Here are some following tables which are of validation and test sets and also having some additional statistics of dataset. The tables contained result model evaluation of our designed model and having summary of indices of germination level from our trained dataset. Testing seeds germination is a big challenge to the seed companies because till now we are only able to germinate seeds or test seeds germination by using a seed germinator machine which takes approximately one week for 50% of seeds germination. Testing of seed germination is an important task for seed scientists to measure seed's quality and performance. Moreover, regular electricity is also required for seed germinator machine which is also a big issue in our country. So, a lot of time and seeds are wasted because of this manual testing of seeds dormancy and germination so here is the germination testing technology based on Artificial Intelligence, Deep learning and RGB data. Using this technology, we can find the germination so seeds within seconds using seeds embryo images.

This work will contribute towards Artificial Intelligence for testing of seeds germination and its dormancy. Currently the seeds companies and government use the seeds germinator machine which costs a lot and waste much time. We are planning to make a model using Artificial Intelligence

which can give us the seeds germination percentage using seeds embryo images.

TABLE I. CONFUSION MATRIX

	Non-germinated	Germinated	background
Non-germinated	67.3	0.4	0
Germinated	3.7	27.8	0
Background	0.2	0.7	-

Green: Seeds germination state is correctly classified.

Yellow: Germination state misclassified.

Orange: Consider/misclassified background as a seed.

Red: Incorrect seeds detection as background detected less seeds present in petri dish.

TABLE II. DATASET MODELS

ResNet50		ResNet101		Inception V2		Inception ResNet-V2	
Value	Test data	Value	Test data	Value	Test data	Value	Test data
96.21	96.29	96.54	96.69	95.81	95.62	97.48	97.90

In the above table the result in percentage is indicated by value while the test set is indicated by test. The value is approximately nearer to the test value which shows the predication and accuracy of the model. The major goal of this project is to evaluate and automate seed germination in order to save government departments' time and reduce the use of manual processes. For the initial training dataset, this experiment still requires some manual procedures for annotation generation. Since germination is a function of time, other machine learning techniques, like convolutional LSTM (Long Short-Term Memory) networks or long-term recurrent convolutional networks, may be able to use the time the image was taken to produce better models and a higher mAP. Similar research topics, such the detection of seedling development, have shown to have high prediction accuracy. In our research, all seed kinds were exclusively examined in petri plates with uniform backdrops. It could be more difficult and involve more testing to determine the germination condition for various greenhouse media. Furthermore, a bounding box only approximates an object's true location, which is particularly unsuitable for seeds with a round form. Modern feature extraction techniques, such as Mask RCNN, which use pixel-accurate locations rather than bounding boxes, could improve precision while decreasing the cost of annotation. Last but not least, smaller items tend to reduce the accuracy of Faster R-CNN models. This problem could be resolved by filming the germination process with higher resolution cameras along with more advanced feature extraction techniques. The availability of GPUs is essential for modern deep learning algorithms. Typically, our

suggested approach can be retrained using transfer learning on different seed types on a workstation with one or more GPUs. Modern single-board systems, like the NVIDIA Jetson Nano (which is comparable to a Raspberry Pi and has an integrated GPU), will make it possible to identify and rate germinated seeds right away without sending data to a robust computing server. This is particularly helpful when scientists only intend to utilise pretrained models, and it can make it simple to set up a high-throughput workflow in greenhouses. Cloud services like Google Colab and Amazon Web Services are another option to physical machines. This has the benefit of making capacity scaling up simple in the event of a requirement. Our tests used a Raspberry Pi computer and a cheap RGB camera module as the picture acquisition system. To explore the photoperiodism of various seeds, photos could potentially be taken using more advanced camera systems, such as hyperspectral or NoIR camera.

V. CONCLUSION AND FUTURE WORK

Modern convolutional neural network architectures were used in our suggested strategy to recognise individual seeds with high precision and distinguish between seeds that had germinated and seeds that hadn't. The models successfully complete the Zea mays hold-out test set with a mAP of over 97%. Furthermore, compared to manual evaluations, the predictions of our algorithm can be used to generate single-value germination indices more precisely. In comparison to traditional and manual methods, our model can assist in accelerating the seed annotation process with reduced error rates and a higher performance for bigger germination studies. Transfer learning is used to retrain the previously pretrained models so that our method may be modified to work with different seed kinds, petri dish material, or illumination conditions.

Currently the model is working for only Zea Mays and in future we are trying to develop a model which will works for all seeds and will be good for society. By finding exact good conditions for seeds testing like a seeds germinator which we install camera and take images and decide whether the seed is germinated or not.

REFERENCES

- [1] J. Bewley, K. Bradford, H. Hilhorst and H. Nonogaki, "Seeds Physiology of Development, Germination and Dormancy," *Seed Science Research*, vol. 23, p. 289, 2013.
- [2] R.Finkelstein, W.Reeves, T.Ariizumi and C.Stebe, "Molecular aspects of seed dormancy," *Annual Review of Plant Biology*, vol. 59, pp. 387-415, 2008.
- [3] T. King, M. Cole, J. M. Farber, G. Eisenbrand and e. al, "Food safety for food security: Relationship between global megatrends and developments in food safety.," *Trends in food science & technology*, vol. 168, pp. 160-175, 2017.
- [4] Y.LeCun, Y.Yann and Y.Bengio, "Convolutional networks for images, speech, and time series," *The Handbook Brain Theory Neural Network*, 1995.
- [5] T. T. Nguyen, V. -N. Hoang, T. -L. Le, T. -H. Tran and H. Vu, "A vision based method for automatic evaluation of germination rate of rice seeds," in *1st international conference on multimedia analysis and pattern recognition (MAPR)*, 2018.
- [6] C.Shorten and T.M.Khoshgoftaar, "A survey on image data augmentation for deep learning," *J Big Data*, vol. 6, 2019.

- [7] M. T.Lin, S.Belongie, P. J.Hays, D. Ramanan and C. Zitnick, "Microsoft COCO: common objects in context," *pringer International Publishing*, vol. 8693, 2014.
- [8] C. Fu, S. Xing, D. Z. D. Zhao, Y. Wei, L. Zhao and X. Yu, "A Study on the Aseptic Germination Method for Rosa rugose Seeds," *Agricultural Sciences*, vol. 8, pp. 150-162, 2017.
- [9] M. G. Nikolaeva, "Factors controlling the seed dormancy pattern," *Physiology and Biochemistry of Seed Dormancy and Germination*, vol. 1977, pp. 51-74, 2013.
- [10] B. Boelt, S. Shrestha, Z. Salimi, J. R. Jorgensen, M. Nicolaisen and J. M. Carstensen, "Multispectral imaging – a new tool in seed quality assessment," *Seed Science Research*, vol. 28, pp. 222-228, 2018.
- [11] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *EEE Trans Pattern Anal Machine Intell*, vol. 39, pp. 1137-1149, 2016.
- [12] M. J. Holdsworth, L. Bentsink and W. J. J. Soppe, "Molecular networks regulating Arabidopsis seed maturation, after-ripening," *dormancy and germination*, 2008.

Drinking Water Monitoring: Computer Vision Kit for Early E.coli Detection

Mansoor Khan
 Department of CS & IT
 UET Peshawar, Pakistan
 mansoor.khan@uetpeshawar.edu.pk

Samad Riaz
 Department of Electrical Engineering
 UET Peshawar, Pakistan
 samadriaz@uetpeshawar.edu.pk

Gul Muhammad Khan
 Department of Electrical Engineering
 UET Peshawar, Pakistan
 gk502@uetpeshawar.edu.pk

Abstract—This work presents an easy-to-use and accurate method to find up to 1 coliform unit (CFU) of a pathogenic bacterium i.e., *Escherichia coli* (*E. coli*) in 100ml of drinking water in 6-8 Hours of the incubation period. A larger number of CFUs is easy to detect and incubation time is reduced to 5-7 Hours for the testing samples containing more than 20 CFUs. Normally in laboratories up to 1 ml of a water sample is spread on an endo agar medium and incubated for about 24 Hours, and the *E. coli* coliform in metallic green color becomes visible through the naked eye. Which has a limitation of finding 1 CFU in just 1 ml of water and a limitation of a large amount of time. In the proposed work Membrane filtration method is used for experiments and a microscopic camera with deep learning algorithms i.e., yolov5 and yolov8 is used for the early detection and counting of *E. coli* colonies. This system is generalized on the field data of 8k images taken from different cities' water samples in Pakistan. Yolov5s model achieved a mean average precession (mAP@0.5) of .949, while the latest release version yolov8 achieved mAP@0.5 of 0.950. An automatic imagery system is developed that takes the images just by placing a petri dish in it processes those images through Raspberry Pi, and shows the detected colonies on the screen, while remote users can use a low-cost microscopic camera manually with a developed mobile application.

Keywords: Water Quality; *Escherichia coli*; Computer Vision.

I. INTRODUCTION

Water contamination is a global health issue. According to the World Health Organization (WHO), at least 1.7 billion people worldwide rely on contaminated drinking water. Microbial-contaminated water serves as a transmission source for various pathogens, including bacteria and harmful chemicals, and poses the greatest risk to drinking water safety. It can transmit diseases such as diarrhea, cholera, dysentery, typhoid, polio, and a range of digestive, respiratory, and neurological issues. It is estimated to cause approximately 505,000 diarrheal deaths each year [1]. Therefore, safe drinking water is essential for protecting public health and reducing the risk of waterborne diseases. To ensure its safety, drinking water must be free from harmful levels of pathogens and contaminants. *E. coli* is a type of bacteria that is commonly found in the environment, including in soil, water, and the intestinal tracts of animals and humans. Some strains of *E. coli* can cause illness in humans, particularly when they are present in drinking water. *E. coli* can enter drinking water sources through the feces of infected animals or humans [2]. *Escherichia coli* (*E. coli*) pathologies are classified based on the ailments it causes. Each pathotype results in a unique set of illness symptoms. *E. coli* O157:H7 is a strain of *E. coli* that produces a toxic

substance called Vero toxin or Shiga toxin, which can damage the lining of the small intestine and cause severe diarrhea [3]. In light of illness causes, it is crucial to quickly and accurately identify *E. coli* bacteria to stop epidemics and lower the death rate.

One common method for detecting *E. coli* bacteria is the use of agar media, which is a solid growth medium that contains nutrients and other components that are necessary for the growth and metabolism of microorganisms [4]. This method is limited by time, energy, human error, and testing of very small water samples. However, this method can be used for the creation, isolation, identification, counting, and sensitivity testing of microorganisms, as well as the testing of clinical specimens, food, water, and environmental controls. Deep learning approaches like convolution Neural Networks (CNN) can learn complex patterns in the data and play important roles in measuring the concentration of cyanobacteria in water detecting water impurities [5], etc. In [6] CNN is used with microscopic images for the classification of *E. coli* and *Vibrio cholera* (*V. cholera*) in water waste which usually contains a large number of bacteria and the dataset is collected in the laboratory with a high-resolution microscope from the wastewater samples. In [7] a faster-RCNN algorithm is used for the detection of *E. coli* bacteria on an endo agar medium using a simple camera in 6-10H, which is useful for the laboratory's experimental purposes. This is limited by the testing very small testing water poured on the solid agar medium, and the culture color change to dark which shows the presence of *E. coli*, but it doesn't measure the concentration or number of *E. coli* CFUs present in the testing water sample.

The proposed system uses the membrane filtration method in which 100ml of a water sample is passed from the Millipore membrane filter using a simple filtration assembly and then placed the membrane filter is on an absorbent pad containing methylene lauryl sulfate broth (MLSB) and kept in the incubator. This method aims to find out 1 CFU in 100ml of drinking water. Data is collected both manually and by using an automatic imagery system, from the samples taken from various cities in Pakistan to make a well-generalized dataset of about 33k images. A low-cost microscopic camera is used on which a specific pattern is defined for capturing images manually. Following this pattern while taking images helps in finding colonies that grow anywhere in the petri dish. This is helpful for the remote users using our developed system who have no access to our automatic kit. YOLO is a computationally efficient and accurate framework for object detection unlike two-stage detectors like RCNN, faster-

RCNN does object detection in a single pass [8]. A state-of-the-art computer vision model yolov8 is used for the detection and counting of colonies, which is generalized on our dataset. And finally, both the automatic offline and manual online systems are developed in which a trained model is deployed. The automatic kit contains a Raspberry Pi for image processing, a servo motor for moving the petri dish, and a microscopic camera installed for capturing images from the whole area of the petri dish automatically.

II. LITERATURE REVIEW

Numerous studies have been conducted in this field using various methodologies and datasets of images to train their models. The process for the rapid detection of *e. coli* along with a mobile application is automated and used in [9] by a convolutional neural network (CNN). The CNN model achieved a high accuracy of 96% and was able to predict each sample in just 458ms. The overall process takes 12 to 24 hours and is limited by information on the number of CFUs present in the sample. In [10] a CNN was used to classify and count *E. coli* and *Vibrio cholera* (*V. cholera*) bacteria in wastewater from microscopic images. The CNN had an accuracy of 93.01% and 97.0% for classification and counting, with better performance for the RGB color model. Sensitivity analysis showed that adding Gaussian noise to the images decreased the accuracy of the CNN. A deep learning method (Faster RCNN) using the TensorFlow framework that is 99% accurate and is based on the color variations between images was able to reduce the detection time of *E. coli* bacteria to 6-10 hours [7]. This is limited by counting CFUs and testing a very small testing sample of pure *E. coli* in the laboratory because the color variation occurs with other bacteria presence as well. Further classification of bacterial growth in agar plates was carried out with Coherent microscopy and deep neural networks [10]. This system detection can detect up to 1 CFU per 1000ml (1CFU/L) in 9 hours. This system is complex and the experimental setup is expensive, not portable, and more resources consumable. Two different media two different times before and after incubation, manual filtration process after 5 hours, keeping again another media, and then in the imagery system consumes one expert time while experimenting.

R. Patil et al. (2020) detected viable bacterial cells in water samples within a period of 2 hours with LOD of 1-10 CFU/ml using a cell splitting method and developed a neural network-based system that uses time-lapse microscope images with the microscope (Labomed Lx 300i) to detect and quantify viable bacterial cells in water samples [11]. A ResNet50 is used to detect *E. coli* in images of the optical microscope of water samples collected by lay community workers using a mobile app and field protocols [12]. While the field protocols and mobile app were successful and received positive feedback, the images generated by a low-cost microscope in field conditions were not of sufficient quality for AI detection. The preliminary AI algorithm performed with 94% accuracy in identifying *E. coli* in lab-derived images compared to a gold-standard method, and additional low-cost technologies are being explored to improve image quality. The Correlation parameters, with the help of artificial

intelligence, can accurately detect *E. coli* in water samples in a short time[13]. Neural network-based method for identifying *E. coli* in groundwater samples utilizing physio-chemical water quality factors. In [14] titration and spectroscopic techniques were used to examine the water samples for any physical, chemical, or microbial changes. An artificial neural network (ANN) was used to predict *E. coli* levels in groundwater based on water quality parameters. The best-performing model included Turbidity, pH, Total Dissolved Salts, and Electrical Conductivity as inputs and was optimized using a Bayesian Regularization training algorithm. The superposition-based learning algorithm (SLA) based on Grover's algorithm was effective in accurately predicting *E. coli* levels and could potentially automate real-time bacterial monitoring. L. Lechowicz et al. (2013) used Infrared spectra to classify 109 uropathogenic *E. coli* strains based on their susceptibility/resistance to cephalothin using ANN [15]. Bacteria strains were cultured on LB agar medium at 37°C for 24 hours before IR spectra measurement. The best-designed ANN achieved an error rate of 5% and an accuracy of 83.43% in classifying the strains. Infrared spectroscopy and ANN can be used to classify bacteria based on their antibiotic susceptibility. V. Chandramouli et al. (2020) neural network model was developed to predict *E. coli* levels at six select Lake Michigan beaches using water quality observations and tributary discharge data as inputs [16]. An Excel sheet tool was developed based on the best model to facilitate real-time decision-making by beach managers. The model, developed using historical data and the Bayesian Regularization Neural Network training algorithm, had an average prediction accuracy of 87% in predicting *E. coli* classes. M. Stocker et al. (2022) Several machine learning models were evaluated for predicting *E. coli* concentrations in agricultural pond waters in Maryland over three years [17]. The random forest model provided the lowest root mean squared error in almost all cases and important predictors of *E. coli* included turbidity, dissolved organic matter content, specific conductance, chlorophyll concentration, and temperature, all the process requires 2 hours at 37°C. Model performance did not significantly differ when using 5, 8, or 12 predictors, indicating that additional measurements did not significantly improve the predictive accuracy of the evaluated algorithms. support vector machines, k-nearest neighbor, and stochastic gradient boosting models also performed well in predicting *E. coli* concentrations. An artificial intelligence-based system for quasi-real-time water quality monitoring, specifically focusing on detecting early chemical or bio-contamination [18]. The *E. coli* grow for 1 - 4 hours and the accuracy of the model depends on the time for which *E. coli* grow. The system used advanced pattern recognition algorithms such as Support Vector Machines (SVM) and ANN, as well as innovative sensing technology, to identify anomalies in the water quality parameters of free chlorine concentration, pH, alkalinity, and total organic carbon. In [19] rapid *E. coli* detection method using membrane lauryl sulfate broth (MLSB) employs an indirect impedance technique. MLSB medium is prepared, inoculated with water

samples or bacterial strains, and sealed in glass cells for 24-hour incubation at 44°C, producing characteristic impedance patterns for detection.

III. METHODOLOGY

The methodology of the system can be explained in 4 steps i.e., choosing a method for water testing, data collection, model training, and deployment.

A. Method For Water Testing

Various kits are used to test drinking water in the field during different emergency cases. In [20], three kits are compared i.e., Delagua, Colilert, and Petrifilm based on the accuracy, experimental process, and cost. According to this paper, during an emergency, Colilert MPN should be considered first next to Petrifilm, and last the Delagua, because one can incubate the sample from the human body in the Coliart and petrifilm kits. On the other hand, based on the accuracy, the Delagua kit is the most accurate of the three, followed by Colilert and Petrifilm. Figure 1 shows the performance of the three methods on different numbers of CFUs. To make an accurate system a Delagua kit methodology is used for water testing in which the water sample is passed through a Millipore membrane filter shown in Figure 2 and placed on the absorbent pad containing media and then placed in the incubator for incubation.

B. Setup For Data Preparation

The deep learning model required a large amount of well-generalized data to learn. Collecting a sufficient amount of data through a microscopic camera is hard, time-consuming, and less efficient. The main challenge is the microscopic camera can't cover the whole area of the petri dish in a single image. It is possible to cover the whole area either by moving the microscopic camera or the petri dish. We have made two setups, automatic and manual for moving the petri dish because moving the camera leads to blur, loss of focus, and moving effects in the images.

1) Automatic System for Capturing Images

A simple automatic 2D motion system is designed to move the petri dish both horizontally and vertically shown in Figure 3. It smoothly moves the Petri in front of the microscopic camera without losing the focus on the area. This design is implemented in the final developed prototype shown in Figure 6.

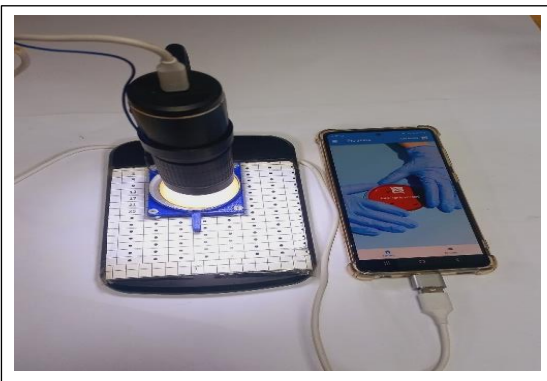


Figure 4: Manual Setup for Data Collection

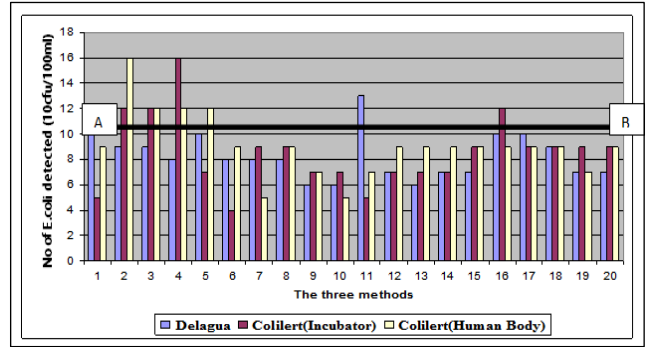


Figure 1: Graphical comparison of results from the three methods at low-level (3rd) dilutions [20].

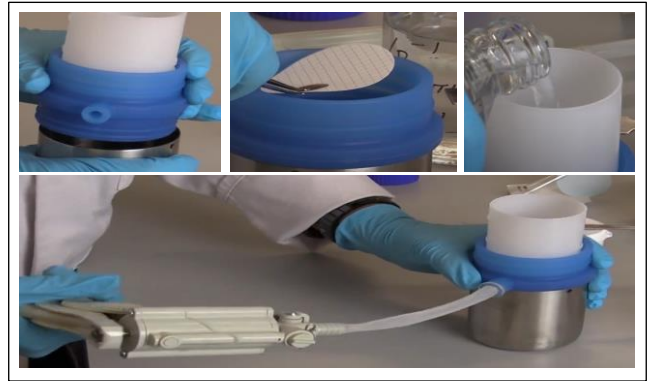


Figure 2: Filtration Assembly Setup, and Method of Using

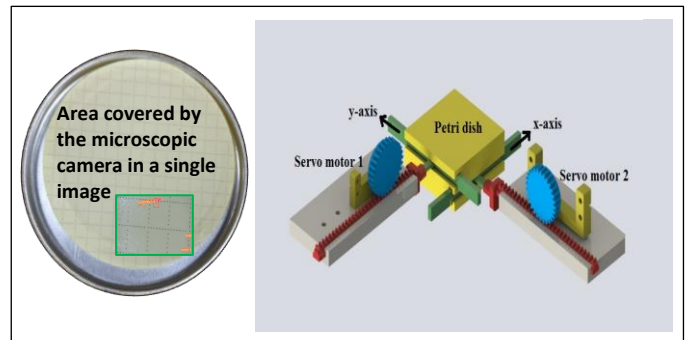


Figure 3: Automatic Moving Petri Dish System Implemented in the Automatic Kit

2) Manual System for Capturing Images

Some data is also collected manually using a 2-megapixel microscopic camera with 50-1000x optical zoom and 1920x1080P resolution shown in Figure 4. The camera comes with a stand, and we paste a custom pattern on the base, following that pattern while taking images through the microscopic camera in a mobile application resulting in covering the whole area of the petri dish in 28 images.

3) Water Samples For Data Collection

The drinking water for testing and creating the dataset has been collected from universities, colleges, homes, and restaurants in the cities of Peshawar, Nowshera, and Charsadda. The images were taken at different time stamps starting from 6H to 10H of the incubation period, which makes our dataset more generalized. Our dataset contains about 8k images with 11k instances of E. coli.

4) Dataset Labeling

As MLSB media is specific to *E. coli* and after 24 Hours of incubation at 37°C, it results in yellow color for *E. coli* and pink for other types of bacteria. In the 6H to 10H, all the colonies look transparent or very light yellow or pink which is very difficult to identify at that time whether it is *E. coli* or other bacteria or some salt particles. To make this easy we kept the 24H data as a reference and annotated *E. coli* colonies accurately and precisely in the 6-10 hours of data.

C. Model Training

YOLO (You Look Only Once) is a deep learning model mostly used for real-time object detection problems in computer vision which has addressed the issues of the traditional object detection algorithm. A recent release of Yolo is version 8. YOLO network is mainly composed of three parts i.e., backbone, neck, and head which are responsible for feature extraction, feature aggregation, and generating detection respectively. For the detection of *E. coli*, we have trained yolov5s [21] and yolov8s [22] models.

Yolo-v5 is the fifth iteration of YOLO, in which CSPDarknet53 is used in the backbone for feature extraction, Path Aggregation Network (PANet) [23] in the neck section for successful generalization on different scale objects, and head which contains detection layers to learn to detect objects of certain sizes. In general, small objects like in our case require higher resolution features and a large number of bounding boxes. YOLO-v5 automatically updates the anchor boxes for the dataset while training. For inferencing speed and accuracy tradeoff, yolo-v5 available is available in various sizes, namely YOLOV5n, YOLOV5s, YOLOv5m, etc. These networks are only different by the number of parameters, yolov5n has the lowest number of parameters and the highest inference speed followed by s and so on. YOLOv5 is trained on the COCO dataset and its various flavors performances are shown in the Figure. For the detection of *E. coli*, we did transfer learning and fine-tuning in yolov5s which has a good performance both in terms of accuracy and inferencing speed and achieved mean average precision (mAP@50) of .949 shown in Figure 5.

Yolov8 is the latest release of the YOLO family. YOLOv8 is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and tracking, instance segmentation, image classification, and pose estimation tasks. YOLOv8 has an anchor-free architecture, with an improved backbone network. Yolov8 is also available in various sizes, namely YOLOv8n, YOLOv8s, YOLOv8m, etc. For our dataset, we did transfer learning and fine-tuning in YOLOv8s and achieved mAP@50 of .950.

1) Mean Average Precision (mAP)

It is a commonly used metric to evaluate the overall performance of object detection models. It takes both precision (1) and recall (2) across different confidence thresholds. Average precession (3) calculates a precision-recall curve for a single class while mAP is the mean of the

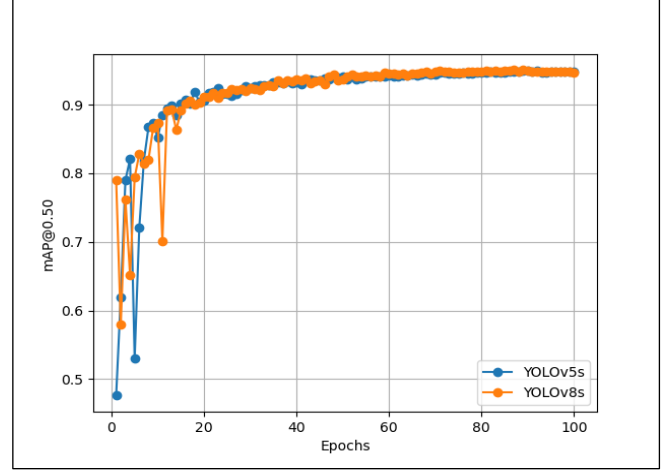


Figure 5: Comparison of YOLOv5s and YOLOv8s on current dataset

average precision calculated for all the classes. In the equations, Precision, Recall, and Average Precision are represented by P, R, and AP respectively. TP shows True Positive and FP shows False Positive.

$$R = \frac{TP}{TP + FP} \quad (1)$$

$$R = \frac{TP}{TP + FN} \quad (2)$$

$$AP = \int_0^1 p(r) dr \quad (3)$$

$$mAP = \frac{1}{k} \sum_{i=1}^k AP_i \quad (4)$$

D. Model Deployment

The trained model is deployed in an automatic independent kit shown in Figure 6 on Raspberry Pi. This kit has a tray for keeping the petri dish and servo motors for moving the petri dish in front of a microscopic camera for taking images which are then processed by Raspberry Pi. Also, the model was deployed on a local server for the remote users using our developed portable mobile application kit shown in Figure 7.



Figure 6: Automatic Imagery System Used for Making Dataset

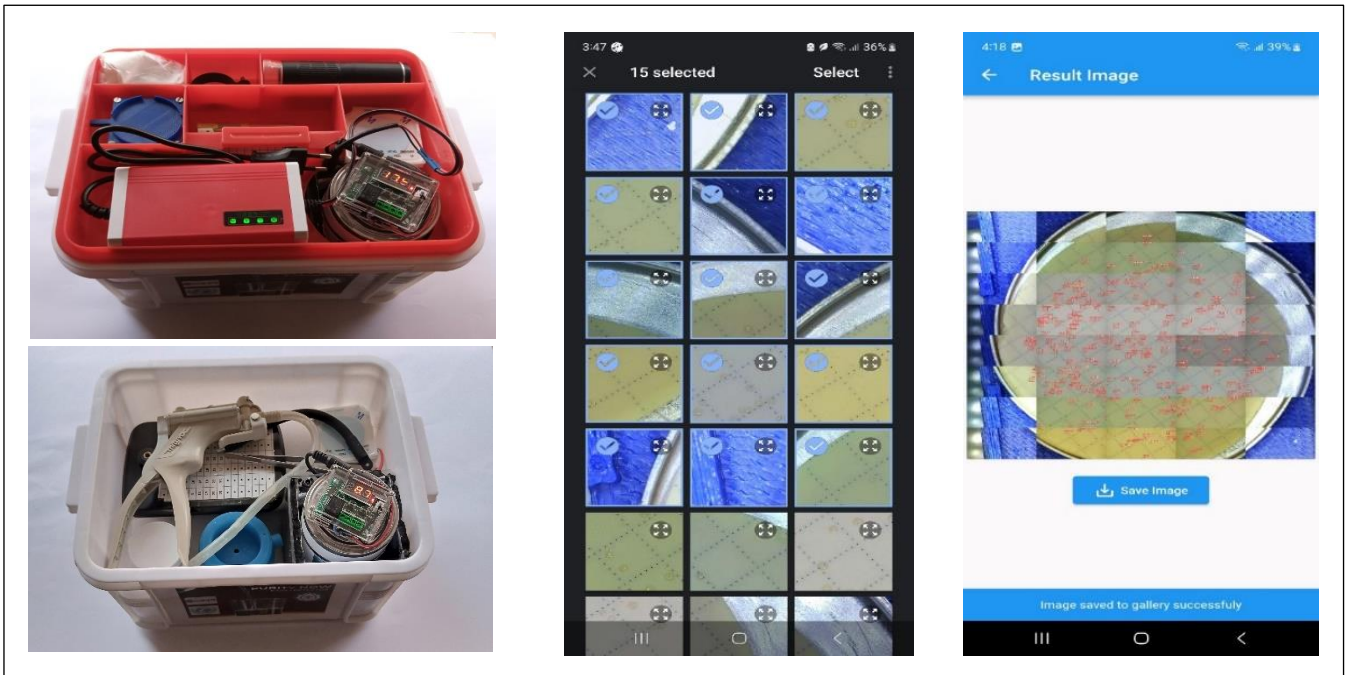


Figure 7: Manual Portable Mobile Application Kit for Remote Users

IV. RESULTS AND DISCUSSION

The final goal is to test our system on the minimum possible number of CFU field samples because samples having a large number of CFUs are easy to detect after a few hours of incubation. So far, we have tested our system for more than 200 field experiments, in which we got 36 experiments having less than 20 CFUs and 27 experiments are correctly classified. In Figure 1, the blue line shows the number of CFUs detected by our system in 7H to 8H and the orange line shows the number of CFUs after 24H. Experiments having more than 20 CFUs are all correctly classified by our system in 7H to 8H. All these testing water samples were taken from different areas in which the system shows good performance. This gap between the lines could be reduced by adding more data to our system and following the SOPs while testing and capturing images.

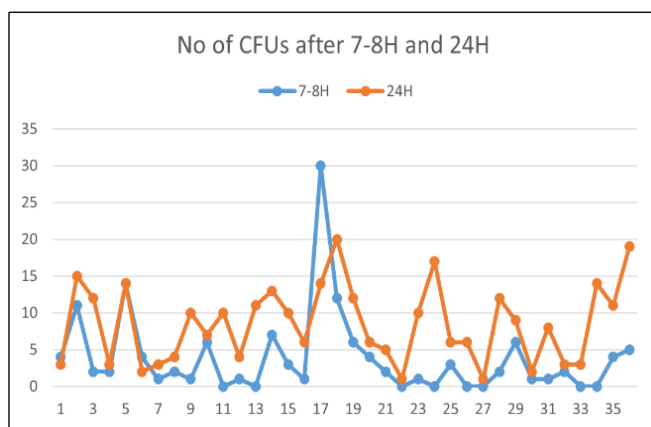


Figure 8: Performance of the Developed System in the field

V. CONCLUSION

In this paper, an easy-to-use, accurate, and portable kit is presented for the early detection of up to 1 CFU of *E. coli* bacteria in 100ml drinking water. This system not only reduces the detection time of *E. coli* bacteria but also reduces power consumption. An existing accurate methodology is used which makes our system accurate and easily integrable in the existing Delagua kit. Both the automatic and manual methods are proposed for local and remote users. The data is automatically stored on the server after processing through which the system performance improves with time.

REFERENCES

- [1] "Drinking water." Accessed: Feb. 14, 2024. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/drinking-water>
- [2] N. S. K. Gunda, S. H. Gautam, and S. K. Mitra, "Editors' Choice—Artificial Intelligence Based Mobile Application for Water Quality Monitoring," *J Electrochem Soc*, vol. 166, no. 9, pp. B3031–B3035, 2019, doi: 10.1149/2.0081909jes.
- [3] P. I. Tarr, C. A. Gordon, and W. L. Chandler, "Shiga-toxin-producing *Escherichia coli* and haemolytic uraemic syndrome," *The Lancet*, vol. 365, no. 9464, pp. 1073–1086, Mar. 2005, doi: 10.1016/S0140-6736(05)71144-2.
- [4] O. Clermont, S. Bonacorsi, and E. Bingen, "Rapid and simple determination of the *Escherichia coli* phylogenetic group," *Appl Environ Microbiol*, vol. 66, no. 10, pp. 4555–4558, 2000, doi: 10.1128/AEM.66.10.4555-4558.2000/ASSET/76C8CB9C-8C42-4A85-B407-7538CAADF53D/ASSETS/GRAPHIC/AM1000238002.JPEG.
- [5] A. Gupta and E. Ruebush, "AquaSight: Automatic Water Impurity Detection Utilizing Convolutional Neural Networks," Jul. 2019, Accessed: Feb. 14, 2024. [Online]. Available: <https://arxiv.org/abs/1907.07573v1>
- [6] T. Irani, H. Amiri, S. Azadi, M. Bayat, and H. Deyhim, "Use of a convolution neural network for the classification of *E. Coli* and *V. Cholera* bacteria in wastewater," *Environmental Research and Technology*, vol. 5, no. 1, pp. 101–110, Mar. 2022, doi: 10.35208/ERT.969400.
- [7] H. Yanik, A. Hilmi Kaloğlu, and E. Değirmenci, "Detection of *Escherichia Coli* Bacteria in Water Using Deep Learning: A Faster

- R-CNN Approach,” *Tehnički glasnik*, vol. 14, no. 3, pp. 273–280, Sep. 2020, doi: 10.31803/TG-20200524225359.
- [8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection.” pp. 779–788, 2016. Accessed: Feb. 14, 2024. [Online]. Available: <http://pjreddie.com/yolo/>
- [9] F. M. Khan, R. Gupta, and S. Sekhri, “A convolutional neural network approach for detection of E. coli bacteria in water,” *Environmental Science and Pollution Research*, vol. 28, no. 43, pp. 60778–60786, Nov. 2021, doi: 10.1007/S11356-021-14983-3/FIGURES/8.
- [10] S. Rauf *et al.*, “Digital E. coli Counter: A Microfluidics and Computer Vision-Based DNAzyme Method for the Isolation and Specific Detection of E. coli from Water Samples,” *Biosensors (Basel)*, vol. 12, no. 1, p. 34, Jan. 2022, doi: 10.3390/BIOS12010034/S1.
- [11] R. Patil, S. Levin, S. Rajkumar, and T. Ajmal, “Design of a smart system for rapid bacterial test,” *Water (Switzerland)*, vol. 12, no. 1, Jan. 2020, doi: 10.3390/w12010015.
- [12] R. Hall-Clifford *et al.*, “Toward co-design of an AI solution for detection of diarrheal pathogens in drinking water within resource-constrained contexts,” *PLOS Global Public Health*, vol. 2, no. 8, p. e0000918, 2022, doi: 10.1371/journal.pgph.0000918.
- [13] J. Carrillo-Gómez, C. Durán-Acevedo, and R. García-Rico, “Concentration Detection of the E. coli Bacteria in Drinking Water Treatment Plants through an E-Nose and a Volatiles Extraction System (VES),” *Water (Basel)*, vol. 11, no. 4, p. 774, Apr. 2019, doi: 10.3390/w11040774.
- [14] F. M. Khan, R. Gupta, and S. Sekhri, “Superposition learning-based model for prediction of E. coli in groundwater using physico-chemical water quality parameters,” *Groundw Sustain Dev*, vol. 13, no. March, p. 100580, 2021, doi: 10.1016/j.gsd.2021.100580.
- [15] Ł. Lechowicz, M. Urbaniak, W. Adamus-Białek, and W. Kaca, “The use of infrared spectroscopy and artificial neural networks for detection of uropathogenic Escherichia coli strains’ susceptibility to cephalothin,” *Acta Biochim Pol*, vol. 60, no. 4, pp. 713–718, 2013, doi: 10.18388/abp.2013_2046.
- [16] M. Khanibaseri, “Developing Artificial Neural Networks (ANN) Models for Predicting E. Coli at Lake Michigan Beaches,” no. August, 2020.
- [17] M. D. Stocker, Y. A. Pachepsky, and R. L. Hill, “Prediction of E. coli Concentrations in Agricultural Pond Waters: Application and Comparison of Machine Learning Algorithms,” *Front Artif Intell*, vol. 4, no. January, pp. 1–12, 2022, doi: 10.3389/frai.2021.768650.
- [18] S. Tinelli and I. Juran, “Artificial intelligence-based monitoring system of water quality parameters for early detection of non-specific bio-contamination in water distribution systems,” *Water Sci Technol Water Supply*, vol. 19, no. 6, pp. 1785–1792, 2019, doi: 10.2166/ws.2019.057.
- [19] S. Timms, K. O. Colquhoun, and C. R. Fricker, “Detection of Escherichia coli in potable water using indirect impedance technology,” *J Microbiol Methods*, vol. 26, no. 1–2, pp. 125–132, Jul. 1996, doi: 10.1016/0167-7012(96)00903-7.
- [20] A. Adegbite, * Adegbite, and A. Ayoade, “COMPARATIVE ASSESSMENT OF FIELD METHODS FOR MICROBIOLOGICAL WATER QUALITY TESTING IN EMERGENCIES,” 2015. [Online]. Available: <https://www.researchgate.net/publication/299461356>
- [21] “GitHub - ultralytics/yolov5: YOLOv5 🚀 in PyTorch > ONNX > CoreML > TFLite.” Accessed: Feb. 15, 2024. [Online]. Available: <https://github.com/ultralytics/yolov5>
- [22] “GitHub - ultralytics/ultralytics: NEW - YOLOv8 🚀 in PyTorch > ONNX > OpenVINO > CoreML > TFLite.” Accessed: Feb. 15, 2024. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [23] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, “PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment.” pp. 9197–9206, 2019.

Low Cost Smart Metering Using Deep Learning

Farhan Khan

Department of Electrical Engineering
University of Engineering and Technology
Peshawar, Pakistan
farhankhan@uetpeshawar.edu.pk

Sarmad Rafique

Department of Computer Systems Engineering
University of Engineering and Technology
Peshawar, Pakistan
sarmadrafiq.ncai@uetpeshawar.edu.pk

Gul Muhammad Khan

Department of Electrical Engineering
University of Engineering and Technology
Peshawar, Pakistan
gk502@uetpeshawar.edu.pk

Abstract—Utility services like electricity, water, and gas are essential for modern living, and their demand has been rising worldwide. However, traditional manual meter reading is a standard procedure for billing purposes. This is not only labor and time-intensive but also prone to mistakes, which results in incorrect billing and revenue losses. In the era of advanced AI, leveraging cutting-edge technology to automate meter readings has become increasingly viable. However, Existing AI-based meter reading systems have limitations in detecting and recognizing meters from a distance. This research addresses these problems by presenting a novel system that utilizes the YOLOv8 model to detect meter screens from a distance. In addition, the system uses a fine-tuned Paddle OCR to recognize meter readings. A Novel dataset curated for the meter screen detection, recognition, and end-to-end OCR tasks related to electricity, gas, and water utility meters has been presented, containing up to 8,044 images. The proposed system was trained and extensively tested on the proposed dataset to gauge its performance. The system achieved an exceptional mean Average Precision (mAP) of 0.995 for both analog and digital meters on the detection task; furthermore, the system achieved an accuracy of 96.92% in the recognition task, which is 70% better than the accuracy of Pre-trained Paddle OCR. Moreover, an all-encompassing evaluation that combines detection and recognition using Paddle OCR and YOLOv8, i.e., the end-to-end OCR task, achieved an accuracy of 97.8%. Lastly, the system achieved an inference speed of up to 6 frames per second, guaranteeing real-time effectiveness.

Index Terms—Yolo-v8; Paddle OCR; Meter Detection; Automatic Meter Recognition; Low cost Smart Metering

I. INTRODUCTION

Accurate meter reading is an essential component of the utility industry. However, these utility sectors still rely on traditional manual meter reading, which is time-consuming, labor-intensive, and prone to errors, resulting in huge financial losses. According to the estimation of the World Bank, electricity distribution companies lose 96\$ billion in revenue yearly due to billing errors [1]. American electric utility companies experienced an estimated 1–10\$ billion USD loss due to billing errors which is 0.5% to 3.5% of its annual GDP [2]. The only energy provider in Peninsular Malaysia, Tenaga Nasional Berhad (TNB) [3], claimed revenue losses of up to 229\$ million annually in 2004 due to billing errors. Even though photo billing has become a popular option nowadays, the manual meter reading remains the same. Each month, an employee of the service company goes to each house to take a picture of the meter and manually enter the billing data, which is time-consuming and error-prone [4].

To tackle these issues, Smart meters [5] are introduced for Automatic Meter Reading (AMR); the goal is to automatically record and invoice the reading of gas, water, and electricity. Even though smart meters have been adopted quickly, the standard procedure of manual meter reading in many places, particularly in developing countries, remains the same. Pakistan’s energy industry, WAPDA, relies on manual meter readings, which has led to several losses. For utility providers, human errors like misreading or incorrectly recording meter readings can result in improper invoicing and revenue losses. These losses make the power industry less financially viable and may limit its capacity to invest in new and improved infrastructure. Similarly, the energy sectors like Water and Gas have also faced a lot of losses due to manual meter readings. Given the difficulties associated with manual reading processes and the gradual substitution of smart meters for traditional ones[6],[7], there is an increasing demand for image-based methods for text recognition to automate the meter reading process, minimize human errors, and lessen the requirement for substantial human resources[8]. Artificial intelligence (AI) [9], a promising technology, can address the difficulties of manual meter reading in utility industries. The process of reading meters could be revolutionized by implementing AI-based metering technologies. AI technology can deliver precise and timely data, it can automate meter reading which improves billing accuracy and lowers operational costs. AI-based technology, such as object detection [10], can be used to perform meter reading detection using models like Yolo [11], SSD [12], and FAST-RCNN [13]. Additionally, Optical Character Recognition (OCR) technology like Easy OCR [14], Keras OCR [15], and Tesseract OCR [16] can be used for meter reading recognition. Electric, water, and gas utilities stand to gain significantly from using AI and Computer Vision Automatic Meter Reading (AMR) technology in photo billing, making its adoption essential for the utility sector. While AI-based meter reading solutions exist, they have limitations in detecting and recognizing meters from a distance. This study presents a novel approach based on deep learning and advanced computer vision. To address the difficulties associated with detecting and recognizing meters from a distance. The suggested system is thoroughly trained on a variety of meter images taken from a variety of meter models both analog and digital, installed on electricity, gas, and water supplies to improve its detection and recognition performance for long

distances. The intended result is a significant improvement in meter reading efficiency, accuracy, and dependability for utility companies across the globe. Specifically designed to operate in real-time from a distance, it utilizes the YOLO-v8 [17] algorithm, which has been trained on a novel custom dataset to provide the best possible detection for analog and digital meters. A novel dataset was also created just for recognition to improve the performance of Paddle OCR [18]. This two-pronged strategy achieves great results.

II. LITERATURE REVIEW

The incorporation of AI in automated meter reading (AMR) technology for utility photo billing has recently been possible because of the development of strong AI models. Utility businesses can achieve increased accuracy in meter reading and billing procedures by utilizing AI algorithms, such as deep learning and computer vision techniques, eliminating human errors. However, AMR has several challenges, like image blur, rotated digits, light reflections, and poor image quality. To overcome these drawbacks, Muhammad Waqar et al. [4] proposed an automated method for extracting and identifying numbers from electric meters that uses Faster R-CNN. Using a dataset from Pakistani electrical providers, the model training achieved a promising result, outperforming Single Shot Detector (SSD), Google Vision API, and conventional techniques. Chun-Ming Tsai et al. [6] introduced a digital region detection system for electricity meters, which achieves a higher accuracy of 99% by implementing the SSD deep learning model. Their methodology involves optimizing the SSD model through training on a dataset of 777-meter pictures. Despite this achievement, One of the limitations is that more real-world tests are required for reliable validation. Convolutional neural networks (CNNs) have also shown great potential in solving the difficult automatic meter reading (AMR) task. Chunshan Li et al. [7] proposed a lightweight spliced convolution network for smart water meter reading that substantially reduces computing load and model space while increasing running time. The system's ability to handle data in real-time when deployed on a distributed cloud platform validates its accuracy and suitability for industrial use. Rayson Laroca et al. [8] contributed a two-stage method for automatic meter reading (AMR) that uses three CNN-based algorithms (CR-NET, multitask learning, and CRNN) for recognition and Fast-YOLO for detection. With a recognition accuracy of 94.13%, the CR-NET model outperforms both multitask and CRNN models. The study also presents the UFPR-AMR dataset containing 2000 annotated images for meter screen detection. Abdullah Azeem et al. [19] proposed a Mask-RCNN (AMR) approach for Detection, Recognition, and Digit Segmentation. The proposed method was assessed on the UFPR-AMR dataset. The suggested method outperforms existing approaches in terms of F-measure and detection accuracy, achieving a prediction rate of 99.82% for counters. An efficient technique for automatic meter reading (AMR) in real-world settings is put forth by Rayson Laroca et al. [20]. Their method, including corner detection and counter

classification, achieved a 34% reduction in reading errors. They also introduced Copel-AMR, a publicly available dataset with 12,500 images of meters; their approach surpassed ten baseline models regarding precision and recognition rate. With 30.64% parameter reduction, Sichao Zhuo [21] presents DAMP-YOLO, a lightweight network for meter reading, by combining DCB, ATA, MDA, and NP with YOLOv8. The model achieves 88.82% mAP50:95, able to recognize objects in real-time on the Jetson TX1. Additionally, Wenwei Lin [22] presents a deep-learning approach for restoring blurry images and recognizing LED digital meters. Polygon-YOLOv5 was used to extract the meter region, and YOLOv5s and CRNN models were employed to recognize the meter readings, achieving 98% accuracy with a 1% missing rate. A sophisticated method for automatic water meter reading was built by Mith Lewis W. Concio et al. [23] using deep learning in a cloud database and mobile app with U-Net binary segmentation for counter detection and Faster RCNN for counter recognition; the pipeline achieves 91.5% accuracy on foreign meters but struggles with 75% accuracy on local meters in the Philippines. Rafaela Carvalho et al. [24] presented a deep-learning model for flow meters and universal controllers as a means of automating manual meter readings. The method consists of screen detection, perspective correction, text detection, template matching, and text recognition. The full pipeline on a taken image takes approximately 1500 milliseconds to complete, whereas screen detection usually takes less than 250 milliseconds. Using YOLO v3 for text extraction and recognition, Muhammad Imran et al. [25] created an automated system for reading electrical energy meters that achieved a 77% precision and 98% recall on a dataset of 10,000-metre images. A lightweight DNN solution for automatic meter reading was introduced by Akshay Kumar Sharma et al. [26], and it outperformed traditional CNN models with 96% accuracy. Although the system contains an Android application for real-time storage and extracts the region of interest, it lacks advanced analysis capabilities and relies on OpenCV for identification. Deyuan Liu [27] combines YOLOv5s with an enhanced k-means algorithm to detect reflecting places in pointer meters for inspection robots. The solution contains a novel robot pose control mechanism for effectively eliminating reflective surfaces, and it shows applicability in complicated situations with a remarkable accuracy of 80.9%. To automate the collection of water meter data in Morocco Ayman Naim et al. [28] developed an AI system that included a Recognition System built on a Convolutional Neural Network (CNN) model. With 140,000 high-quality digital meter photographs as its training dataset, the CNN model scored an astounding 98.70% accuracy during training.

In this work, we adopt a structured approach to address the problem at hand. Section III outlines our methodology, including details on the employed dataset, the Models used, and our proposed framework. Section IV expounded the experimental setup, covering data pre-processing, network

training, and evaluation metrics. Section V delves into the results and discussions regarding the performance of our proposed system. Finally, conclusions are drawn in Section V, encapsulating the findings, contributions, and future directions of this work.

III. METHODOLOGY

A. Employed Dataset

Our process began with the acquisition of a diversified utility meter image dataset to successfully address the detection, recognition, and end-to-end OCR tasks. Therefore, a comprehensive training dataset including 3,905 pictures was produced by incorporating datasets from reliable sources, including the UFPR AMR dataset [8], Water Meters dataset [29], YUVA EB dataset [30], and Gas Meter dataset [31] and around 241 new images were added. The data set’s high quality and diversity make it easier to create more sophisticated algorithms and models for detecting and recognizing meter readings. Random samples from the data set are shown in Figure 1 and Figure 2. Furthermore, a separate novel dataset of 3,154 images was gathered and labeled appropriately for optical character recognition by cropping the meter screen regions from the detection dataset. This dataset is unique, as such, a comprehensive dataset is not available elsewhere. Lastly, the end-to-end dataset contained 985 images and was taken as a subset of the detection dataset.



Figure 1: Detection Dataset Figure 2: Recognition Dataset

B. YOLO-v8

The YOLOv8 [17] model is a real-time, one-stage detection system built on Convolutional Neural Networks (CNN), and it is an improvement over the YOLO (You Only Look Once) series. Acknowledged for its effectiveness in fusing features and providing accurate detection outcomes in a lightweight design, YOLOv8 brings new features and enhancements over its predecessors. YOLOv8’s anchor-free design, which deviates from conventional anchor-based methods, speeds up non-maxima suppression and improves overall detection efficiency. Designed to meet a variety of research requirements, YOLOv8 offers five different scale models (n, s, m, l, x). Three essential modules make up the network architecture, as shown in Figure 3. The Head, Neck, and Backbone modules handle prediction output, multi-feature fusion, and feature extraction, respectively. The Backbone module includes the C2F structure and

uses the Spatial Pyramid Pooling Fusion (SPPF) to improve gradient flow information while keeping a lightweight profile. To improve model generalization and resilience, the Head module provides a Decoupled Head structure, which extracts target location and category information independently. The Neck module uses a PAN (Path Aggregation Network) and FPN (Feature Pyramid Network) technique for feature fusion. Thus, YoloV8 is the current state-of-the-art in object detection.

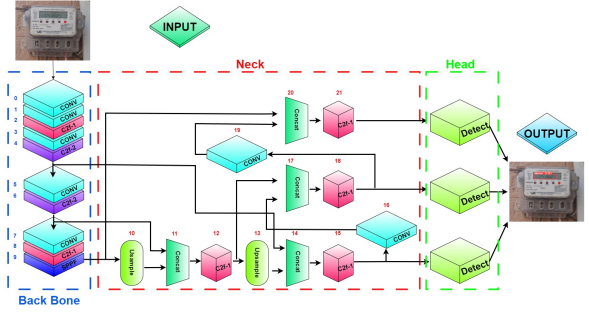


Figure 3: Yolo-v8 Object Detection Architecture

C. Paddle Ocr

Baidu’s PaddleOCR [18] is a powerful OCR model that works with over 25 languages, has pre-trained models, and is very good at recognizing text that is lengthy, vertical, and has digits. It was created by PaddlePaddle and uses deep learning to extract text quickly and accurately. PP-OCRv3, the most recent release, offers independent usability for recognition, classification, and detection. In PP-OCRv3, several optimization techniques are added to increase the recognition model’s effectiveness and precision. To achieve improved performance, Transformer-based SVTR and CNN-based PP-LCNet are combined in the lightweight text recognition network known as SVTR_LCNet, improving prediction speed by 20% without appreciably sacrificing accuracy. The Attention module is used in the GTC method to provide guided CTC training, which enhances accuracy. TextConAug is a data augmentation approach used to improve contextual information variety for better model performance. Thus, PaddleOCR is a very diverse model for character recognition.

D. Proposed Framework

The detection dataset was used to train various versions of the YOLO models for better bench-marking among which the YOLO V8 model produced the best results for detection. Additionally, PaddleOCR was trained and fine-tuned specifically for Recognition. The validation set was used to thoroughly validate the model’s performance.

After training, the detection and recognition model is incorporated into a unified workflow. The first step in the method is to take a picture of the meter display or screen, which will be used as input into the model. Preprocessing for detection is then carried out. The model evaluates the confidence level after determining the location of the meter screen. The image is sent back to the detection pre-processing stage if the confidence value is less than 0.5. On the other hand, the Detected region

is cropped and sent for character recognition preprocessing if the confidence level is higher than 0.5. The intended outcome is attained if the cropped region is Recognised and the model's confidence level is higher than 0.5. If the model's confidence level is less than 0.5, the picture is returned to the detection phase until the model's confidence level rises over 0.5.

The Block diagram of the overall working of the system is shown in Figure 4.

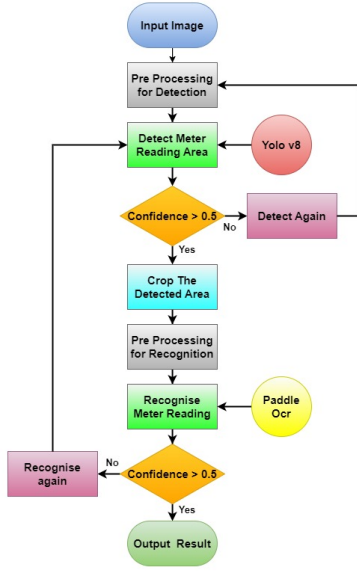


Figure 4: Block diagram of Proposed Framework

IV. EXPERIMENTAL SETUP

A. Data Pre-processing

The Detection dataset was further split into training and validation sets using a split ratio of 70% for training and 30% for validation. Thorough pre-processing procedures, including the elimination of redundant, superfluous, and noisy images, were carried out before the model was trained, and bounding box annotation was used in the training of the detection model on the dataset. To improve the dataset balance and diversity, methods such as image scaling, standardization, and augmentation were also used. The recognition dataset was divided into training and validation sets using 80% and 20% split ratios, respectively. The end-to-end dataset was used to evaluate the overall performance of the proposed system and thus was not split into train and validation ratios. The Bounding box annotation was performed on the detection dataset using the Labelimg tool, and the annotation files were saved in txt format.

B. Networks training

The proposed system in this study was trained and evaluated on a computer running on the Windows 10 OS with Intel Core i7-10700 CPU @ 2.90 GHz, NVIDIA GeForce RTX 3060 12 GB, 16 GB Ram, and the programming language used was Python 3.7 with the PyTorch framework.

The training parameters for the detection and recognition are summarized in Table 1.

TABLE I: Training Parameters

No.	Parameters for Detection and Recognition Models			
	Detection Parameters	Details	Recognition Parameters	Details
1	Picture size	640 x 640	Picture size	48 X 320
2	Epochs	300	Epochs	500
3	Batch size	16	Batch size	128
4	Optimizer	SGD	Optimizer	Adam
5	Learning rate	0.01	Learning rate	0.001
6	Workers	8	Workers	4
7	Patience	40		

C. Evaluation Metrics

To provide a thorough assessment of the proposed system, relevant evaluation metrics were used for each task. F1 score, precision, recall, mAP50, and mAP50-90 were used to measure detection performance; these metrics provide an extensive assessment of the system's efficiency in object detection. For the end-to-end OCR and recognition tasks, metrics like Character Error Rate, Recognition accuracy, and Character Accuracy were used to gauge their performance. CER is more suited for single-word recognition tasks, such as meter readings than Word Error Rate, which is used for sentences. Because the recognition task is unpredictable and the recognized output may contain extra or missing data, we refrained from utilizing precision, recall, or F1 score.

i. F1-Score:

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

ii. Precision:

$$P = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

iii. Recall:

$$R = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (3)$$

iv. mAP50:

$$\text{MAP50} = \frac{1}{|X|} \sum_{i=1}^{|X|} \text{AvgP}(z_i) \quad (4)$$

Where:

- $|X|$: Number of queries in the dataset.
- z_i : i -th query in the dataset.
- $\text{AvgP}(z_i)$: Average precision for the i -th query, using the first 50 items in the ranked list.

v. mAP50-90:

$$\text{MAP50-90} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \text{AvgP}(q_i) \quad (5)$$

Where:

- $|Q|$: is the total number of queries in the dataset.
- q_i : represents the i -th query in the dataset.
- $\text{AvgP}(q_i)$: Average precision for the i -th query, using only the top 50 to 90 ranked items.

vi. Character Error Rate:

$$\text{CER} = \frac{S + D + I}{C} \times 100 \quad (6)$$

Where:

- S : Substitutions
- D : Deletions
- I : Insertions
- C : Total characters in reference transcription

vii. Recognition Accuracy:

$$A = \left(\frac{N}{T} \right) \times 100 \quad (7)$$

Where:

- N : Correctly recognized meter readings
- T : Total meter display readings

viii. Character Accuracy:

$$CA = \left(\frac{CC}{TC} \right) \times 100 \quad (8)$$

Where:

- CC : Number of Correct Character
- TC : Total Characters

V. RESULTS AND DISCUSSION

Using the proposed dataset, the training results of several detection models produced impressive results, as shown in Table II. Impressively, our system obtained an F1-Score of 99.3%. While every model performed well, YOLOv8 Nano was particularly noteworthy as it produced the greatest results in terms of mAP@50 achieving 0.995, and mAP@50-90 achieving 0.826, mAP50 and mAP50-90 plots of all training models are shown in Figure 5 and Figure 6. At 99.4% and 99.3%, respectively, the meter detection precision and recall hit their peak, and no further optimization was possible. Turning to recognition models, we carefully examined the most famous OCR models like Karas OCR and Paddle OCR. By testing these two OCR models, using a subset of 976 cropped meter screen images from the end-to-end dataset. Paddle OCR was the clear winner, with better performance than its competitor, as shown in Table III. Therefore, Paddle OCR was fine-tuned on the recognition dataset, and an accuracy of 99.21% was achieved. When this fine-tuned model was tested using the previous subset dataset, an accuracy of 96.92% with a CER of 0.0054 was achieved.

A. Proposed system Performance

The fine-tuned YOLOv8 Nano was chosen due to its high inference speed and better results and was combined with Paddle OCR to form the proposed framework which was tested on the end-to-end dataset comprising 987 images, an overall accuracy of 97.8% was achieved encapsulating both detection and recognition performance. The results are visualized in Figure 7. The inference speed of the proposed framework was around 6 frames per second, showcasing real-time performance. This amalgamation of detection and recognition models showcases a promising avenue for bolstering the accuracy and efficiency of meter detection and recognition tasks.

TABLE II: Training results of the Detection Models

Model	Performance Evaluation metrics for Detection Task				
	F1 score	Precision	Recall	mAP-50	mAP50-90
YOLOv5-n	99.0%	99.2%	98.9%	0.995	0.801
YOLOv5-s	99.2%	99.2%	99.3%	0.995	0.805
YOLOv7	99.3%	99.5%	99.3%	0.994	0.817
YOLOv8-n	99.3%	99.4%	99.3%	0.995	0.826
YOLOv8-s	99.3%	99.6%	99.2%	0.994	0.825

TABLE III: Bench-marking of Recognition Models

Model	Performance Evaluation Metrics for Recognition Task			
	Accuracy	CA	CER	Training Accuracy
Keras OCR Pre-trained	1.536%	6.663%	0.8759	Nil
Paddle OCR Pre-trained	19.87%	39.79%	0.5042	Nil
Paddle OCR Fine-tuned	96.92%	99.11%	0.0054	99.21%

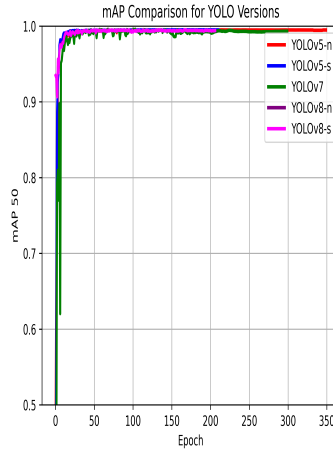


Figure 5: mAP0.5 Training curves

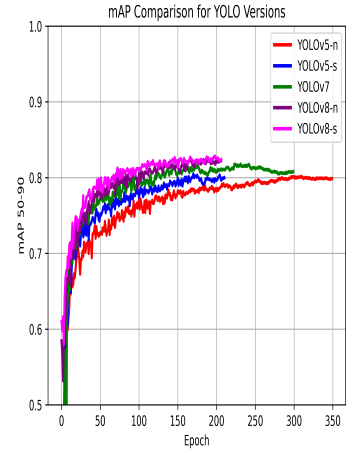


Figure 6: mAP0.5-0.9 Training curves

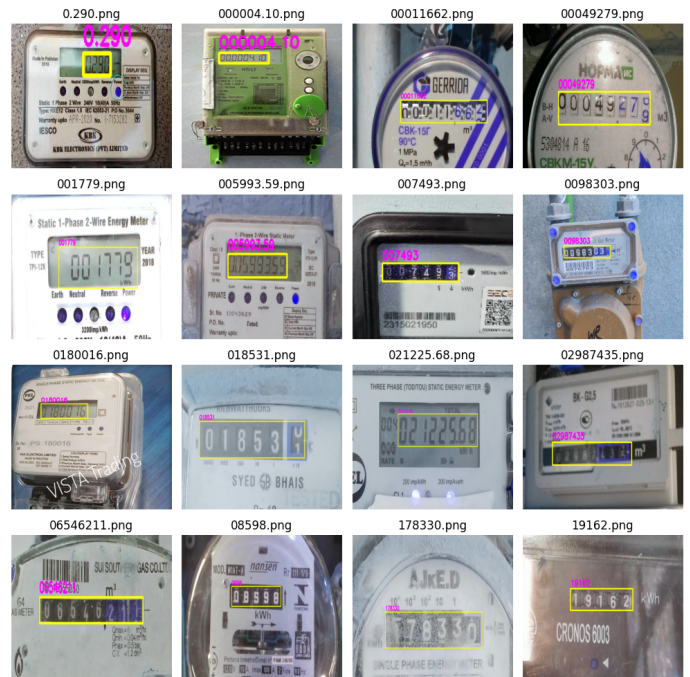


Figure 7: End to End OCR Model Results Visualization

VI. CONCLUSION

Our primary objective in this research has been to tackle the complex problem of detecting and recognizing both digital and analog meters from a distance. We observed the core limits of current AI systems and presented a groundbreaking solution by utilizing and combining the advanced features of YOLOv8 for meter screen detection and Paddle OCR for digit recognition for an end-to-end OCR system; our study attempted to close this gap and produced an excellent mean Average Precision (mAP) of 0.995 and an F1 score of 99.3%. Furthermore, the recognition performance of the system, powered by 99.21% accuracy Paddle OCR, highlights how effective our suggested method is in addressing the drawbacks of existing systems. We hope to further the progress of meter reading technology by releasing publicly accessible datasets that are expressly intended for detection, recognition, and end-to-end AMR tasks. We aimed to contribute not only to the advancements in meter reading technology but also to provide a benchmark for the research community to evaluate and build upon. Our system's dependability and efficiency are confirmed by the extensive testing on the proposed datasets, which includes a sizable dataset of 8044. There is room for our system to be expanded in the future. Using forecasting models in conjunction with past consumption data is one approach that is worth investigating. This calculated addition might improve meter reading validation, providing a more thorough and precise result.

ACKNOWLEDGMENT

We express our sincere gratitude to the National Center of Artificial Intelligence at the University of Engineering and Technology, Peshawar, for providing the computational resources for this research.

REFERENCES

- [1] India's power sector. (2012, August 9). World Bank. <https://www.worldbank.org/en/news/feature/2010/04/19/india-power-sector>
- [2] Smith, T. B. (2004). Electricity theft: a comparative analysis. *Energy policy*, 32(18), 2067-2076.
- [3] Annual Report Tenaga Nasional Berhad 2004 TNB, 2004
- [4] Waqar, M., Waris, M. A., Rashid, E., Nida, N., Nawaz, S., Yousaf, M. H. (2019, October). Meter digit recognition via Faster R-CNN. In 2019 International Conference on Robotics and Automation in Industry (ICRAI) (pp. 1-5). IEEE.
- [5] Cooper, A., Shuster, M., Lash, J. (2021). Electric company smart meter deployments: foundation for a smart grid (2021 update). Institute for Electric Innovation: Washington, DC, USA.
- [6] Tsai, C. M., Shou, T. D., Chen, S. C., Hsieh, J. W. (2019, July). Use SSD to detect the digital region in electricity meter. In 2019 International Conference on Machine Learning and Cybernetics (ICMLC) (pp. 1-7). IEEE.
- [7] Li, C., Su, Y., Yuan, R., Chu, D., Zhu, J. (2019). Light-weight spliced convolution network-based automatic water meter reading in smart city. *IEEE Access*, 7, 174359-174367.
- [8] Laroca, R., Barroso, V., Diniz, M. A., Gonçalves, G. R., Schwartz, W. R., Menotti, D. (2019). Convolutional neural networks for automatic meter reading. *Journal of Electronic Imaging*, 28(1), 013023-013023.
- [9] Zhang, C., Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration*, 23, 100224.

- [10] Zhao, Z. Q., Zheng, P., Xu, S. T., Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), 3212-3232.
- [11] Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B. (2022). A Review of Yolo algorithm developments. *Procedia Computer Science*, 199, 1066-1073.
- [12] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., Berg, A. C. (2016). Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14* (pp. 21-37). Springer International Publishing.
- [13] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Conference on computer vision* (pp. 1440-1448).
- [14] JaidevAI. (n.d.). GitHub - JaidevAI/EasyOCR: Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, Cyrillic and etc. [GitHub](https://github.com/JaidevAI/EasyOCR).
- [15] keras-ocr — keras_ocr documentation. (n.d.). <https://keras-ocr.readthedocs.io/en/latest/>
- [16] tesseract-ocr. (2019, October 20). tesseract-ocr/tesseract. [GitHub](https://github.com/tesseract-ocr/tesseract).
- [17] Jocher, G., Chaurasia, A., Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. <https://github.com/ultralytics/ultralytics>
- [18] PaddlePaddle/PaddleOCR. (2022, April 19). [GitHub](https://github.com/PaddlePaddle/PaddleOCR).
- [19] Azeem, A., Riaz, W., Siddique, A., Saifullah, U. A. K. (2020, August). A robust automatic meter reading system based on mask-rcnn. In 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA) (pp. 209-213). IEEE.
- [20] Laroca, R., Araujo, A. B., Zanlorensi, L. A., De Almeida, E. C., Menotti, D. (2021). Towards image-based automatic meter reading in unconstrained scenarios: A robust and efficient approach. *IEEE Access*, 9, 67569-67584.
- [21] Zhuo, S., Zhang, X., Chen, Z., Wei, W., Wang, F., Li, Q., Guan, Y. (2023). DAMP-YOLO: A Lightweight Network Based on Deformable Features and Aggregation for Meter Reading Recognition. *Applied Sciences*, 13(20), 11493.
- [22] Lin, W., Zhao, Z., Tao, J., Lian, C., Zhang, C. (2023). Research on Digital Meter Reading Method of Inspection Robot Based on Deep Learning. *Applied Sciences*, 13(12), 7146.
- [23] Concio, M. L. W., Bernardo, F. S., Opulencia, J. M., Ortiz, G. L., Pedrasa, J. R. I. (2022, November). Automated Water Meter Reading Through Image Recognition. In *TENCON 2022-2022 IEEE Region 10 Conference (TENCON)* (pp. 1-6). IEEE.
- [24] Carvalho, R., Melo, J., Graça, R., Santos, G., Vasconcelos, M. J. M. (2023). Deep Learning-Powered System for Real-Time Digital Meter Reading on Edge Devices. *Applied Sciences*, 13(4), 2315.
- [25] Imran, M., Anwar, H., Tufail, M., Khan, A., Khan, M., Ramli, D. A. (2023). Image-Based Automatic Energy Meter Reading Using Deep Learning. *CMC-COMPUTERS MATERIALS CONTINUA*, 74(1), 203-216
- [26] Sharma, A. K., Kim, K. K. (2021). Lightweight CNN based Meter Digit Recognition. *Journal of Sensor Science and Technology*, 30(1), 15-19.
- [27] Liu, D., Deng, C., Zhang, H., Li, J., Shi, B. (2023). Adaptive Reflection Detection and Control Strategy of Pointer Meters Based on YOLOv5s. *Sensors*, 23(5), 2562.
- [28] Naim, A., Aaroud, A., Akodadi, K., El Hachimi, C. (2021). A fully AI-based system to automate water meter data collection in Morocco country. *Array*, 10, 100056.
- [29] Water Meters Dataset, 1244 photos masks. (2020, February 29). Kaggle. <https://www.kaggle.com/datasets/tapakah68/yandextoloka-water-meters-dataset>
- [30] Kanagarathinam, K., Sekar, K. (2019). Text detection and recognition in raw image dataset of seven segment digital energy meter display. *Energy Reports*, 5, 842-852. https://drive.google.com/drive/folders/1J9TYUilKdJKfSeotL_EIyvXQ-3pE282?usp=sharing
- [31] Iqbal, A., Basit, A., Ali, I., Babar, J., Ullah, I. (2021). Automated Meter Reading Detection Using Inception with Single Shot Multi-Box Detector. *Intelligent Automation Soft Computing*, 27(2). Available: <https://github.com/arifqbal2018/meter>.

Visually: Assisting the Visually Impaired People Through AI-Assisted Mobility

Muhammad Arsalan
Kamran
Dept. of Computer
Systems Engineering
University of Engineering
and Technology
Peshawar, Pakistan
20pwcse1957@uetpeshaw
ar.edu.pk

Alishba Orakzai
Dept. of Computer
Systems Engineering
University of
Engineering and
Technology
Peshawar, Pakistan
20pwcse1953@uetpesha
war.edu.pk

Umama Noor
Dept. of Computer
Systems Engineering
University of
Engineering and
Technology
Peshawar, Pakistan
20pwcse1967@uetpesha
war.edu.pk

Yasir Saleem Afridi
Dept. of Computer
Systems Engineering
University of
Engineering and
Technology
Peshawar, Pakistan
yasirsaleem@uetpesha
war.edu.pk

Madiha Sher
Dept. of Computer
Systems Engineering
University of
Engineering and
Technology
Peshawar, Pakistan
madihasher@uetpesha
war.edu.pk

Abstract—This research introduces “VisuAlly”, a revolutionary mobile application which aims to address the complications that visually impaired people come across in their daily lives. By deploying advanced deep learning models for real-time object detection, facial recognition, and currency identification with voice outputs for each feature, the “Visually” app strives to enhance the autonomy, independence and mobility of visually impaired people. The system undergoes thorough training on a diverse dataset, incorporating augmentation techniques to enhance the robustness of the models. The project's multifaceted objectives include a user-friendly interface, real-time object detection, multi-modal recognition, Text-to-Speech audio output, and an overarching aim of enriching the lives of visually impaired individuals. Driven by the global prevalence of the visual impairment and the demand for cost-effective solutions, “Visually” is aligned with the international efforts for accessibility and inclusivity. For cross-platform compatibility the machine learning models have been integrated whilst being deployed with TensorFlow Lite. With Offline availability, the app ensures accessibility even in rural areas with limited network connectivity. To make a substantial societal impact “VisuAlly” aims to contribute to a more inclusive and equitable society, by transforming the way visually impaired individuals navigate around the environment. Positioned at the intersection of technology, accessibility, and empowerment, the “Visually” project is poised to bring about positive change for a community that frequently encounters unique challenges in their daily lives.

Keywords: VisuAlly; Mobile application; visual impairments; Real-time object detection; facial recognition; currency identification; societal impact; technology; positive change; Visually impaired individuals.

I. INTRODUCTION

In the realm of assistive technology, individuals with visual impairment face challenges everyday that significantly impact their self-sufficiency, independence and mobility. Despite the various existing technological aids, the challenge of navigating independently in unacquainted environments remains a pervasive issue, often requiring reliance on sighted assistance. This paper aims to address such challenges through the introduction of “VisuAlly,” an innovative mobile application developed to empower the visually impaired community. The primary goal of the project is to offer a real-time vision system, incorporating object detection, face recognition, and currency recognition features. These functionalities, integrated into the “VisuAlly” app, incorporating deep learning models trained on a diverse dataset carefully curated for the unique needs of the visually impaired. This introduction outlines the background, research

problem, and significance of the “VisuAlly” project, which

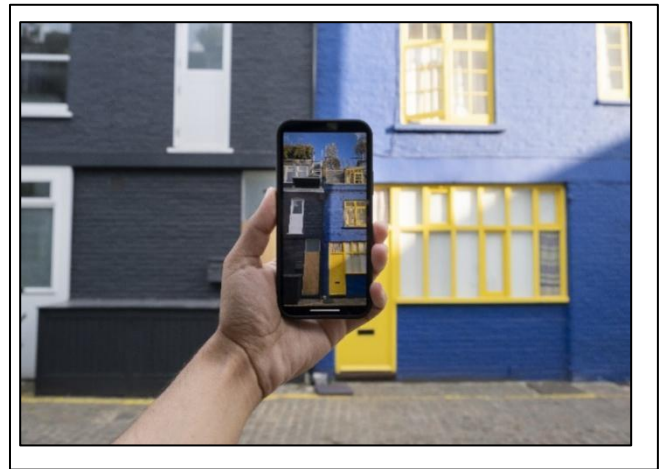


Figure 1: Navigation support using mobile app [16]

aims to ameliorate the daily experiences of visually impaired individuals by enhancing their independence, accessibility and mobility. The main contributions of this research include addressing the challenges faced by visually impaired individuals, the global prevalence of visual impairment, and the development of an affordable and accessible solution with the potential to have a positive societal impact. The following sections will delve into the technical details of the model architectures, training methods, and the expected impact of the “VisuAlly” app on its users.

II. RELATED WORK

A. Microsoft's “Seeing AI”:

The landscape for assistive technologies for visually impaired individuals has witnessed significant contributions from notable projects and research endeavors. One of the major initiative is Microsoft's “Seeing AI,” a cloud-powered application that integrates Machine Learning (ML) to describe the user's surroundings and converting it into voice output, including features like text recognition, scenic descriptions and more. “Seeing AI” can be considered as a valuable benchmark for understanding of the widespread usage beneficial for visually impaired users, influencing the development decisions and design considerations for the “VisuAlly” app [1].

B. Noteify (Currency Recognition):

Furthermore, “Noteify”, an app made in India for currency detection focuses on verifying accurate financial handling for the visually impaired users, making it a worthy solution for the requirement [2].

C. Cutting-edge Object Detection Techniques:

In the field of advanced object detection, Mahendran and his team explore cutting-edge solutions based on real-time vision systems using deep learning and point cloud processing. By investigating these techniques, the team helps the app recognize objects quickly and accurately. Their research is crucial for refining the goals of the "VisuAlly" app, especially in computer vision and object recognition [3].

D. DEEP-SEE FACE Framework:

The DEEP-SEE FACE framework introduces an intriguing approach using convolutional neural networks (CNNs) for real-time facial recognition. This technology provides valuable insights for developing facial recognition models, focusing on features like hard negative mining and acoustic communication. These aspects align with "VisuAlly's" goals of improving user understanding and communication, making the app more user-friendly and effective for visually impaired individuals [4].

E. Currency Recognition System:

A notable innovation is a currency recognition system designed for the visually impaired, enabling real-time identification of currency notes and obstacle-aware navigation. This system offers essential considerations for integrating an efficient and accurate currency recognition feature into the "VisuAlly" app, ensuring that users can manage their finances independently and safely [5].

F. Summary

In summary, the insights from these different sources serve as guiding principles for the development of the "VisuAlly" app. Each project and research paper provides unique perspectives and considerations, collectively shaping the innovative, user-centric solutions aimed at transforming the lives of visually impaired individuals. Through the integration of these technologies, "VisuAlly" aims to empower users, enhancing their independence and quality of life.

III. METHODOLOGY

The project’s research design was meticulously crafted to ensure the effective implementation of its goals. The development of the “VisuAlly” project follows a thorough and systematic methodology, covering all stages from data collection to model deployment. This approach is detailed in the following sections, providing a glimpse into the research design, data acquisition process, model training, and deployment strategies. Fig 2 demonstrates the development stages for the VisuAlly app.

A. Data Collection:

The “VisuAlly” Project is built upon a rich and extensive dataset, specifically tailored and designed to meet the unique requirements of the visually impaired people. The dataset

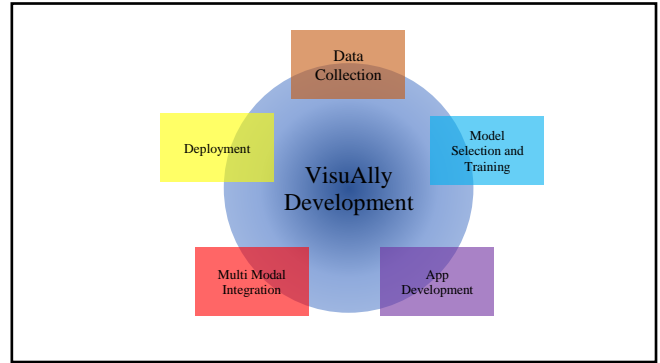


Figure 2: Development Stages for VisuAlly App

includes a wide variety of indoor and outdoor objects, with a focus on objects essential for daily navigation. To ensure the model, that is to be trained on this dataset, adaptability to various real-world scenarios, the dataset undergoes careful augmentation and labeling. Techniques such as rotation, scaling, and flipping the images are employed to improve the model’s ability to learn different features of the images and enhance its accuracy in identifying objects from different angles and lightning conditions.

B. Model Architecture and Training:

The core of the “VisuAlly” app is its deep learning models, which utilizes the “You Only Look Once (YOLO)” architecture known for its ability to detect objects in real-time videos with high accuracies. The YOLO model is pretrained on a diverse dataset but can be self-trained on specific dataset to detect specific objects. This model is trained on the augmented dataset using advanced optimization methods, such as stochastic gradient descent. During training, the focus is on minimizing detection loss while enhancing the model’s accuracy in recognizing and pinpointing objects in the user’s environment.

Table 1. Differences between Yolo Architecture and MobileNet SSD

Features	YOLOv5	MobileNet SSD
Number of Layers	Backbone: 53 layers Additional: depends on variant	Backbone: 13 layers Additional: 6 layers
Optimizer Used	Adam	Adam
Number of Epochs	Around 300	Around 200
Dataset	COCO dataset	VOC dataset
Images Used	Around 118K images	7,000 images
Average Accuracy	mAP of 50.1 on COCO test set	mAP of 72.7 on VOC test set
Frames per Second	60+ FPS on a GPU	Over 100 FPS on modern hardware
Model Size	27 MB	23 MB

Table 1 compares two popular object detection models, YOLOv5 and MobileNet SSD, across various metrics. YOLOv5 is a state-of-the-art object detection model known for its efficiency and accuracy. It features a highly optimized architecture, enabling it to achieve real-time inference speeds

on GPU hardware. YOLOv5 has gained popularity for its simplicity, ease of use, and impressive performance on a variety of datasets. It offers a range of pre-trained models of different sizes (e.g., YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x), allowing users to choose the right balance between speed and accuracy for their specific application. Additionally, YOLOv5 is actively maintained and updated, ensuring compatibility with the latest advancements in object detection research. Overall, YOLOv5 is a compelling choice for developers seeking a powerful and efficient object detection solution.

C. Multi-Modal Recognition:

The "VisuAlly" app offers a comprehensive user experience by integrating various recognition features such as object, face, and currency recognition. Face recognition utilizes the weights of the pretrained models and performs personalized training to achieve precise identification. Currency recognition involves the development of a specialized dataset and model, providing users with information about the currency notes they come across [6].

D. Text-to-Speech (TTS) Integration:

To enhance communication with visually impaired people, the "VisuAlly" app provides an audio output feature powered by Text-to-Speech (TTS) technology. This feature not only describes the objects but also provides contextual information, such as the proximity to the obstacles, identifying which feature the user is currently using and giving audio output of the recognized faces etc.

E. Deployment:

To achieve compatibility across Android and IOS platforms, all machine/deep learning models in the app are converted into TensorFlow Lite (tflite) versions. The app is developed using the Flutter framework, which utilizes the tflite_flutter_helper package for efficient image processing and input preprocessing.

F. Offline Functionality:

Recognizing the importance of accessibility in areas with limited network coverage, "VisuAlly" emphasizes offline functionality. By using tflite files, all the models will be stored on the device which will allow the user to make use of the features even without internet services. Face embeddings and emergency contacts are stored locally, reducing the reliance on a constant internet connection.

This approach aligns with the project's goal of creating a user-friendly, adaptive, and accessible solution for visually impaired individuals, ensuring that the app effectively addresses their unique challenges and contributes to positive societal impact [7].

IV. RESULTS

By utilizing the YOLO-based object detection model, "VisuAlly" can swiftly and accurately identify a wide range of objects in real-time, such as obstacles, furniture, and other navigation aids. This feature offers users instant information

about their environment, improving their spatial awareness and safety.

Our thorough assessment of "VisuAlly's" object detection abilities revealed exceptional performance, demonstrating the reliability and effectiveness of the integrated technologies.

A. Laptops Detection:

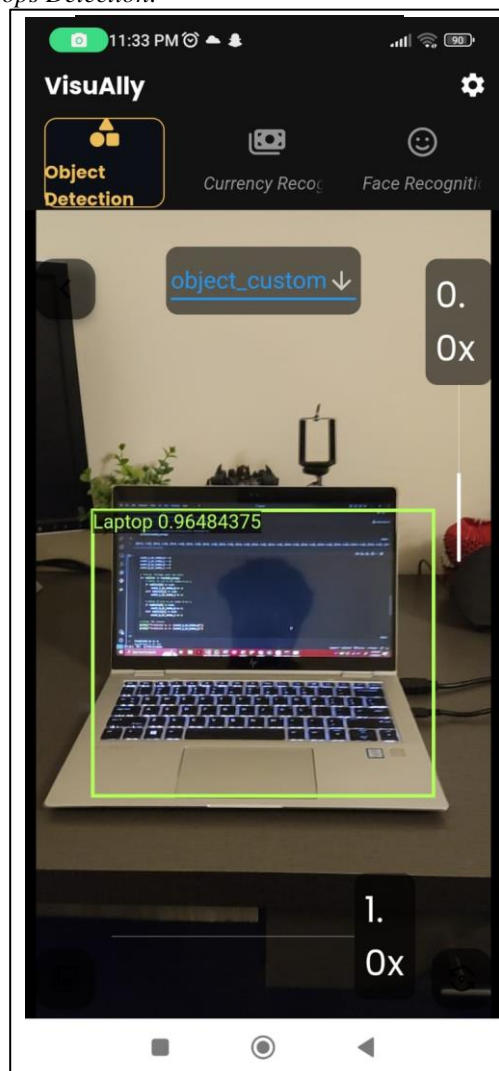


Figure 3: Laptop Detected

Fig 3 shows the snapshot of the app detecting a laptop. The model demonstrated exceptional proficiency in identifying laptops, achieving an impressive accuracy of 96%. This high level of accuracy is essential for users who rely on the app to navigate environments where laptops are prevalent.

B. Bottle Detection:



Figure 4: Bottle Detected

The model detected everyday objects, like water bottles, with exceptional accuracy, reaching 96%, as demonstrated in Fig 4. This ensures that the app can help a user in assisting with daily tasks such as locating personal items or identifying objects in their immediate vicinity.

C. Person Detection:

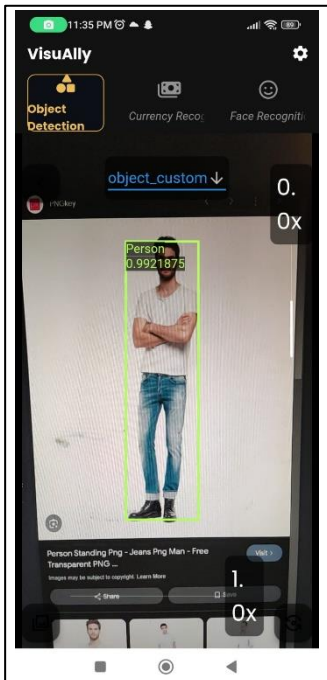


Figure 5: Person Detected

Recognizing individuals in front of the user is a critical aspect of the app's functionality. In Fig 5, the detection of a person can be seen. Our model achieved an impressive accuracy of

99% in person detection, ensuring users are aware of the presence of others in their surroundings.

D. Car Detection:

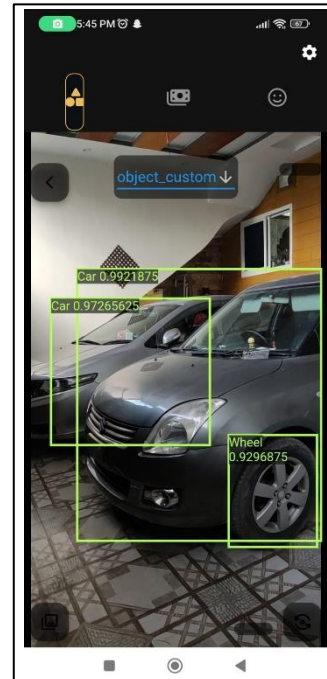


Figure 6: Car Detected

The most important objects to be detected in daily navigation are vehicles. Our model successfully detects cars in front of the user with a remarkable accuracy of 99%, making daily navigation easy for visually impaired people. Fig 6 demonstrates the snapshot of our app where it is detecting vehicles.

E. Multi-Object Detection:

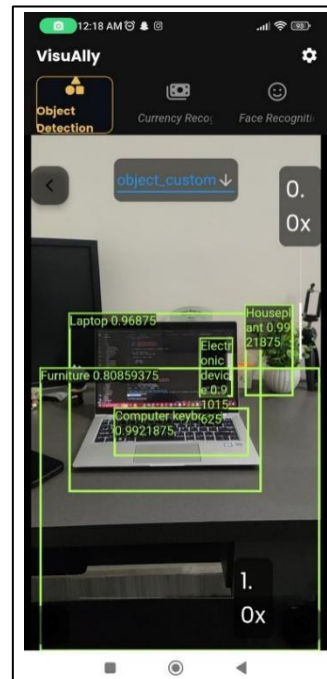


Figure 7: Multi Objects Detected

The app's ability to detect multiple objects simultaneously further enhances its utility. With a focus on accuracy, the app

provides users with detailed information about their environment, allowing for more informed navigation decisions. Fig 7 shows the multi object detection capabilities of VisuAlly.

Moreover, TTS API is utilized to announce each object to the user that is detected, providing detailed information about their environment and further enhancing the app's usability.

V. DISCUSSION

The "VisuAlly" app enhances the mobility of visually impaired individuals through the integration of YOLO v5 for real-time object detection and Google's ML Kit for text-to-speech conversion. These technologies offer several key benefits:

A. Immediate Environmental Feedback:

By incorporating real-time object recognition technology, visually impaired people may navigate safely and autonomously by being promptly informed about their surroundings. Their improved spatial awareness makes them more competent navigators.

B. High Precision Object Identification:

The remarkable accuracy of YOLO v5 ensures accurate information about the surroundings is provided and informs the user about the items in their immediate environment, hence enhancing their spatial comprehension.

C. Efficient Navigation:

The app vocalizes the names of detected items, especially in unfamiliar locations, reducing users' cognitive load and allowing them to focus more on their surroundings.

D. Enhanced Safety:

The safety of the visually impaired people is improved by accurate object detection and TTS announcements, which notify them of potential hazards or obstructions in their route. Because of the trustworthy information given to the visually impaired people, they may navigate with more confidence thanks to this proactive approach to safety.

E. User-Friendly Interface:

The user-friendly interface of the app makes it easier even for consumers with little technological background. The incorporation of these technologies into a Flutter app result in an intuitive user interface that is simple to use. This accessibility is essential to guarantee that a broad spectrum of users, regardless of their level of technical expertise, can use the app efficiently.

In conclusion, the integration of YOLO v5 and Google's ML Kit in the "VisuAlly" app greatly enhances the mobility and independence of individuals with visual impairments. This integration allows them to confidently explore their surroundings, ensuring they have the necessary information for safe and effective navigation through the combination of real-time object detection and text-to-speech conversion.

VI. COMPETITIVE ANALYSIS

"VisuAlly" aims at providing an extensive solution to enhance the independence and mobility of visually impaired

and blind people, exhibiting several significant advantages in comparison to the already available apps designed on the same objective. This discussion concentrates on elucidating the project's results and implications, including comparison of results with the available solutions, whilst highlighting the significant contribution of "VisuAlly" to the field.

A. Comparison with Existing Apps:

"VisuAlly" discerns itself from the multiple existing apps for the visually impaired people through its encompassing approach and multi-modal features. Current apps mostly focus on singular functionalities such as object detection or navigation, whereas "VisuAlly" incorporates real time object detection with face recognition and currency recognition, making it an extensive solution catering to meet the multiple needs of users. Additionally, features like offline mode, user-friendly design, and continuous user feedback integration make "VisuAlly" unique.

Other popular apps like Be My Eyes and Seeing AI offer useful features such as remote assistance and text recognition. However, they may not offer the same depth of functionalities and offline capabilities as "VisuAlly." The use of YOLO architecture for object detection in "VisuAlly" enhances its real-time processing capabilities, setting it apart from other solutions.

B. Significance of Results and Contribution to the Field

The "VisuAlly" project represents a significant advancement in assistive technology, specifically tailored to address the complex challenges faced by visually impaired individuals in their daily lives. Through its versatile and integrated approach, the app not only enhances users' independence but also reduces their reliance on others, fostering a sense of empowerment.

One of the key strengths of "VisuAlly" lies in its offline functionality, which ensures accessibility in a variety of environments, thereby filling a crucial gap in existing solutions. Moreover, the project's significance goes beyond its technical capabilities; it embodies a commitment to inclusivity, affordability, and user-centered design principles. By aligning with global initiatives for accessibility, "VisuAlly" acknowledges the widespread impact of visual impairment and strives to make a meaningful difference in the lives of affected individuals. The integration of user feedback further enhances the project's value, ensuring that the app evolves in response to the evolving needs of its users.

VII. CONCLUSION

"VisuAlly" emerges as a groundbreaking assistive technology, designed to address daily challenges for visually impaired individuals. Utilizing advanced deep learning models, including real-time object detection, face recognition, and currency recognition, the app aims to transform the landscape of independence and mobility.

One of the key features of "VisuAlly" is its commitment to a user-friendly interface and multi-modal recognition, including Text-to-Speech audio. This comprehensive approach is further enhanced by rigorous training on a diverse

dataset, ensuring the app's adaptability to various real-world scenarios and reflecting its focus on meeting user needs.

In line with global accessibility initiatives, "VisuAlly" places a strong emphasis on affordability and offline functionality, which are essential for users in diverse environments. By combining the YOLO architecture for real-time processing with a holistic approach, the app surpasses existing solutions in its field.

What sets "VisuAlly" apart from other apps is its offline capabilities, user-friendly design, and the integration of continuous user feedback. Beyond its technical capabilities, the app embodies inclusivity and user-centric design principles, contributing to a more equitable society.

In conclusion, "VisuAlly" represents a significant step towards empowerment for visually impaired individuals, redefining how they navigate the world and fostering inclusivity. With its responsive development cycle, the app is poised to have a lasting impact, evolving to meet the changing needs of its users and contributing to a more accessible world.

ACKNOWLEDGEMENT

We acknowledge the usage of the GenAI platform for refining and enhancing the writing quality of this manuscript intended for the submission at the ICTIS'24 conference.

REFERENCES

- [1] "Seeing AI" [Online]. Available: <https://www.microsoft.com/en-us/ai/seeing-ai>.
- [2] "Noteify: Indian Currency Recognition App" [Online]. Available: <https://github.com/chandran-jr/Noteify>
- [3] Jagadish K. Mahendran, Daniel T. Barry, Anita K. Nivedha, Suchendra M. Bhandarkar; "Computer Vision-based Assistance System for the Visually Impaired Using Mobile Edge Artificial Intelligence" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2021, pp. 2418-2427
- [4] B. Mocanu, R. Tapu and T. Zaharia, "DEEP-SEE FACE: A Mobile Face Recognition System Dedicated to Visually Impaired People," in IEEE Access, vol. 6, pp. 51975-51985, 2018, doi: 10.1109/ACCESS.2018.2870334.
- [5] Asmaeil, Ibthaj & Albagul, Abdulgani & Ali Ahmed, Abdussalam. (2022). Currency Recognition System Using Image Processing: Libyan Banknote as a Case Study. 10. 10.30534/ijeter/2022/171022022.
- [6] Joshi, R.C.; Yadav, S.; Dutta, M.K.; Travieso-Gonzalez, C.M. "Efficient Multi-Object Detection and Smart Navigation Using Artificial Intelligence for Visually Impaired People." Entropy 2020, 22, 941. <https://doi.org/10.3390/e22090941>
- [7] J. Madake, S. Bhatlawande, A. Solanke and S. Shilaskar, "A Qualitative and Quantitative Analysis of Research in Mobility Technologies for Visually Impaired People," in IEEE Access, vol. 11, pp. 82496-82520, 2023, doi: 10.1109/ACCESS.2023.3291074.
- [8] Khokare, Ganesh & Solanki, Kalpeshkumar. (2022). REAL TIME OBJECT DETECTION WITH SPEECH RECOGNITION USING TENSORFLOW LITE. 9. 552-559.
- [9] Alsharabi, Naif. (2023). Real-Time Object Detection Overview: Advancements, Challenges, and Applications. مجلة جامعة عمران. 3. 12. 10.59145/jaust.v3i6.73.
- [10] Vaishnavi, K. & Reddy, G. & Reddy, T. & Iyengar, N. & Shaik, Subhani. (2023). Real-time Object Detection Using Deep Learning. Journal of Advances in Mathematics and Computer Science. 38. 24-32. 10.9734/james/2023/v38i81787.
- [11] https://pub.dev/packages/google_mlkit_object_detection
- [12] <https://blog.codemagic.io/live-object-detection-on-image-stream-in-flutter/>
- [13] <https://www.ijitee.org/wp-content/uploads/papers/v8i12S/L110310812S19.pdf>
- [14] "Why use Python for AI and ML". [Online]. Available: <https://steelkiwi.com/blog/python-for-ai-and-machine-learning>
- [15] "Python Community". [Online]. Available: <https://www.python.org/community>
- [16] https://www.freepik.com/free-photo/hand-taking-photos-with-smartphone-city_29609123.htm

An aggregated approach towards NILM on ACS-F2 using Machine Learning

Arsalan Ali Mujtaba

Department of Electrical Engineering
University of Engineering and Technology
Peshawar, Pakistan
arsalanali@uetpeshawar.edu.pk

Sarmad Rafique

Department of Computer Systems Engineering
University of Engineering and Technology
Peshawar, Pakistan
sarmadrafique@uetpeshawar.edu.pk

Gul Muhammad Khan

Department of Electrical Engineering
University of Engineering and Technology
Peshawar, Pakistan
gk502@uetpeshawar.edu.pk

Abstract—The Energy Sector across the globe is experiencing rapid growth, driven by Internet of Things (IoT) integration technologies and advanced algorithms. This evolution is particularly evident in the ongoing competition among tech companies in the development of smart metering solutions. Despite these advancements, a critical challenge persists—the lack of definitive technical protocols for monitoring the total usage or power signatures of individual appliances, referred to as non-intrusive load monitoring (NILM) in aggregate. While intrusive load monitoring (ILM) provides very accurate and thorough insights, non-intrusive methods are essential to address losses specially in residential areas. In this research a groundbreaking approach is proposed towards handling NILM problems by analyzing and aggregating the load patterns of four key appliances of daily use, namely the Coffee Machine, Fridge, Kettle, and Laptop from the ACS-F2 dataset. The generated aggregated dataset, is systematically combined using electrical formulations to yield the desired data which reflects the simultaneous operation of multiple appliances, this has been explored for the first time in the known literature. The proposed dataset contains around 6750 aggregated appliance load patterns for both training and testing. Furthermore, multiple Time Series Classifiers (TSC) were gauged using a suite of evaluation metrics, on the proposed dataset and an accuracy of 92.1% was achieved by the CATCH22 classifier.

Keywords: Non-Intrusive Load Monitoring (NILM); Intrusive Load Monitoring (ILM); Appliance Identification; Load Patterns; AEON toolkit; Energy Disaggregation.

I. INTRODUCTION

Complex algorithms used in digital devices are becoming popular in the context of global digital advancements. Smart energy metering, these days, is an important topic worldwide owing to its growing demand. According to the Federal Energy Regulatory Commission US, the annual smart metering and demand response study shows an yearly growth of approximately 8 million additional smart meters. The annual report, mandated by regulations, includes 2021 smart meter data sourced from the EIA and the Institute for Electric Innovation. The figures from these sources indicate 111.2 million and 115.3 million smart meters, reflecting penetrations of 68.3% and 70.8%, respectively, among a total of 162.8 million metering endpoints in the United States. Additionally, it's noteworthy that the year 2021 marked the sixth consecutive year with an increase of nearly 8 million advanced meters [1]. IoT has seen a surge in its popularity which has increased the interest of many developers to quickly incorporate it in energy metering [2]. The commercial energy metering is advancing relatively slowly as compared to other areas of digital innovation. The absence of comprehensive data on individual appliance energy usage is a significant problem with commercial energy metering nowadays [3]. To prevent energy theft or loss, one must be

aware of the energy consumption of their appliances. George Hart came up with the concept in the beginning and had a patent for it in the 1980s [4]. His research demonstrates the analysis of both single-state and multistate appliances, as shown in Figure 1, including freezers and many more [5].

With technological advancements, the nature of load has evolved significantly. It's a challenge not only for the energy disaggregation problem but also for a low maximum demand. In the past, energy disaggregation was simpler because of the less complex energy patterns unlike the modern appliances. Fortunately, sophisticated machine learning models simplify the energy disaggregation problem [6]. Researchers' work in feature engineering equips us with powerful tools to enable the available models to classify more complex patterns. As stated by the famous No Free Lunch Theorem, there is no single exclusive ML strategy that can handle all possible problems better than any other machine learning algorithm [7]. In ML, the most time-consuming process is algorithm and model selection. Machine learning has seen a growth in recent years, resulting in the creation of very effective categorization techniques that help in the selection of models and algorithms.

The ACS-F2 dataset was selected for this work because of its low sampling rate which is similar to the commercial energy meters and could be easily deployed after the parameters are finalized [8]. This dataset, an improved version of the previously developed ACS-F1 dataset [9] comprises 15 classes, each representing a different appliance, and 15 subclasses for each class representing different manufacturing brands, totaling approximately 225 load signatures of appliances. This dataset consists of two distinct instances: A1 and A2 used for training and testing respectively. The subclasses represent the load signatures of appliances from various manufacturers, allowing for comparisons. The ACS-F2 dataset's appliance signatures are pre-segregated, and the pre-processed data is then fed into the proposed algorithm, which combines the data into an aggregated dataset, for this study to be implementable [10]. AEON is a toolkit for learning tasks that is specially being used here for TSC [11]. It is compatible with scikit-learn and offers the ability to access the latest algorithms for time series machine learning, and additionally offers an abundance of conventional methods for learning tasks including classification and forecasting [12]. This research focuses on utilizing the models included in AEON toolkit.

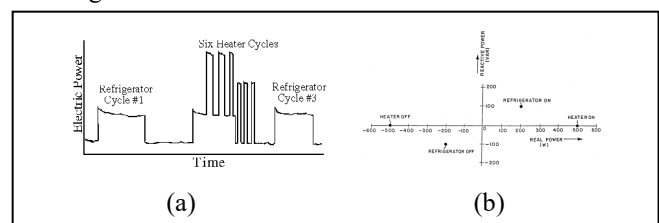


Figure 1: Classification of Appliances (a) Total Load vs Time graph approach by G. Hart for NILM [5] (b) Segregate appliances by mapping them on Real vs Reactive Plot. Source: G. Hart Patent [4]

II. RELATED WORK

Researchers have recently concentrated on categorizing appliances according to their electrical properties, using two main approaches: ILM and NILM. NILM uses larger electrical patterns for categorization, whereas intrusive load monitoring requires direct access to specific appliances for in-depth study [5]. A pivotal figure in the evolution of intelligent load classification is George Hart, who introduced the concept in the 1980s and patented his work in 1989. Hart's approach incorporated a cluster analysis unit to differentiate between various appliances operating simultaneously as illustrated in Figure 2. This marked a significant advancement in load monitoring methodologies, particularly for intrusive approaches [4].

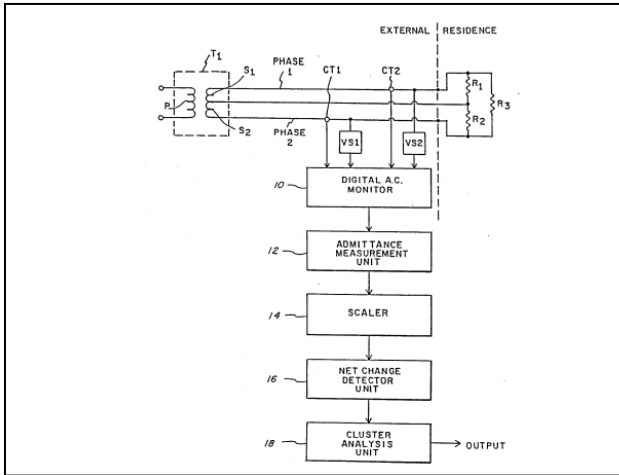


Figure 2: Patented NILM hardware design by George Hart [4]

It's noteworthy that non-intrusive load monitoring technique predates its intrusive counterpart. This method typically involves installing an energy meter at a household's primary supplying point and disaggregating energy consumption through a thorough examination of electrical patterns. The advantages of NILM lie in its holistic approach to understanding energy usage without direct appliance access. NILM has undergone a radical change in the modern environment due to the widespread implementation of smart energy metering and IoT. By leveraging these technologies, aggregated data can be seamlessly transmitted to online servers, where sophisticated algorithms can be applied with minimal difficulty as Ruano, A., et al. [13] presented in their research. This integration enhances the efficiency and accessibility of energy analysis, opening new possibilities for optimizing consumption patterns and enhancing overall energy management strategies.

According to Yu, L., et al. [14], A better knowledge of electricity consumption and appliance safety is anticipated from smart home systems, which are anticipated to meet the increasing need for intelligent and energy-saving services. Home appliance power consumption must be measured and tracked in the context of a smart grid, since comprehensive data on power loads becomes essential for demand management and optimization.

NILM's optimization is difficult because of the multiple states of each appliance, erratic consumption patterns, and the difficulties in detecting concurrent appliance consumption. Furthermore, appliances frequently generate distinct waveforms. This complexity and the management of

aggregated information create exponential hurdles, as Kelly describes [15]. Furthermore, NILM could be categorized as a pattern recognition problem, employing techniques based on event detection to classify and recognize appliance properties, as shown in Figure 1(b). Algorithmic strategies centered around event detection aim to determine the distinguishing characteristics of particular appliances after identifying important events, hence aiding differentiation and identification [13]. Salihagiü, Kevric, et al. [10] studied the use of sophisticated feature extraction approaches to improve the comprehension of household energy consumption patterns in the context of load categorization using the ACS-F2 dataset of Appliance Consumption Signatures. The study addresses the challenge of predicting appliance states based on historical data and suggests utilizing machine learning algorithms implemented in the WEKA software for effective categorization.

Puente, C., et al. [16] provide a comprehensive overview of techniques for NILM, showcasing their growing importance in energy disaggregation amid the increasing deployment of smart meters employing the well-known UK-DALE dataset [17] and a public dataset of a single house from France [18]. It emphasizes the uses of NILM in Ambient Assisted Living (AAL) and Home Energy Management Systems (HEMS), with a focus on new approaches, emphasizing the significance of determining appliance status for informed decision-making. Additionally, it offers insights into future research directions in these areas, complementing existing NILM reviews. Angelis, G.-F., et al. [19], discussed the distinct energy consumption behavior of electrical appliances, referred to as "load signatures," allowing for four classifications. Lamps and toasters are examples of Type I appliances, which have binary ON/OFF states. Type II equipment, which includes washing machines, has many finite states with distinct patterns. Type-III equipment, such as dimmer lights, have continually fluctuating usage, which makes disaggregation challenging. Finally, Type-IV equipment, such as television receivers, is constantly operating, resulting in specific consumption characteristics. These categories help to understand energy use without requiring further equipment deployment.

Moreover, Kelly, J et.al. [17, 20], focused on using deep neural networks to energy disaggregation using the UK-DALE dataset. Salerno, V.M. et.al, employed LSTM, regression networks and denoising autoencoders in their study, surpassing previous methods like combinatorial optimization and factorial hidden Markov models on real aggregate power data from five appliances. Their models demonstrated notable performance on unseen homes. In their work they address the challenges in power disaggregation, highlighting issues with Artificial Neural Networks (ANNs) and Factorial Hidden Markov Models (FHMMs). ANNs demand extensive training and large datasets for optimal performance, while FHMMs face computational burdens and scalability issues [21]. To overcome these, the study suggests employing extreme learning machine technique, utilizing randomly selected hidden units and analytically computed output weights. Results demonstrate its superiority over FHMMs and ANNs on the UK-DALE dataset [17], showing better generalization to new homes and reducing the need for extensive training data.

In Section III the methodology to this is explained which includes the dataset preprocessing steps and NILM dataset

generation based on the provided combinational formulations. The architecture of the classifier selected is also explained including its feature extraction methods. In Section IV, results and discussion section the results generated from the classifier used and the other available classifiers of the AEON toolkit are listed.

III. METHODOLOGY

The choice of a high-quality dataset is crucial for successful AI project which influences the subsequent model selection and preprocessing methods. Typically, iterative trial-and-error processes determine the most suitable models and techniques. For this study, the ACS-F2 dataset, which has signatures for the most common daily use appliances and has a low sampling frequency which is similar to that of the commercial energy meters, was selected based on these factors.

A. DATASET

For this study, the icosys Institute-owned ACS-F2 opensource dataset [8] was selected. The measurements in this dataset are 2 hours long and are kept in separate files with the labels A1 and A2. There are 15 different appliance kinds and each appliance kind has been sampled from 15 different manufacturers, thus the dataset contains 225 samples for training and an additional 225 samples for testing.

Among the electrical variables for classification in the dataset are real power (W), reactive power (var), RMS current (A), frequency (Hz), RMS voltage (V), and phase of voltage relative to current (ϕ). Among them, real power, reactive power, perceived power, power factor, and RMS current are considered for classification purposes.

The ACS-F1 dataset comprised the following items: microwaves, coffee machines, mobile phones (with battery chargers), LCD TVs, Hi-Fi systems with CD players, laptops, computer stations with monitors, refrigerators and freezers, and printers.

More appliances, including fans, kettles, incandescent lights, shavers, and monitors, were added in the second version. There are originally around 360 time steps in each sample for every device in the collection. The total number of samples per appliance is changed after preprocessing, which handles NaN values and performs zero padding to match the sample lengths. Figure 3 shows the algorithm flow, research workflow, and the procedure from dataset preprocessing to the result generation of the trained model.

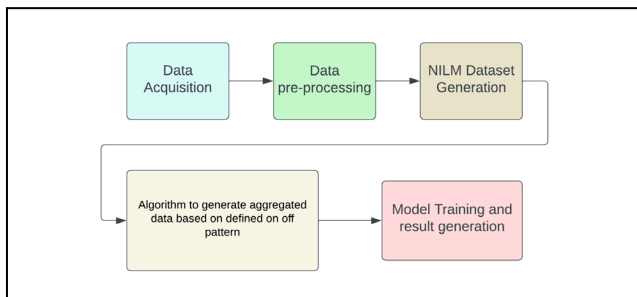


Figure 3: NILM algorithm workflow

1) *Preprocessing*: The dataset is preprocessed to address any discrepancies or missing values before model training and assessment. Among the preprocessing actions are:

a) *Extracting data from dataset files*: The dataset is made up of files in different formats. Data extraction from these files is needed to make additional processing and model training easier.

b) *Handling Outliers*: Removal of Nan values and other outliers is essential to ensure the authenticity of training results.

c) *Padding Values and Interpolation*: The sample size might decrease for certain subclasses due to outlier removal. If deleted values lie between two adjacent ones, they can be substituted with an interpolated value. Also, if the sample counts across subclasses aren't balanced, padding should be applied to minimize impact on the original dataset.

d) *Normalization*: To promote model convergence and ensure consistency across features, the dataset is standardized to a uniform scale.

e) *Storing Preprocessed dataset*: The preprocessed dataset is stored separately from the original files. This facilitates efficient model training by reducing the computational overhead.

2) *NILM Dataset Generation*: To train a model for NILM, we combine data from individual appliances using various combinations of appliances for the training and testing data. In this study, four appliances of daily use are selected and controlled by their on/off states using software filters for all relevant electrical parameters during the training session to generate the desired NILM dataset. Amongst the selected four appliances, all of their 15 appliance model categories are included. For this, we apply binary logic to turn on and off all four appliances sequentially and one appliance model from each appliance at a time so in total 16 aggregated classes are generated in which since, each appliance has 15 different models, turning on and off all 15 models of the same appliance at a time makes the total count of unique aggregated appliance data to be 3375 for classification. Making use of both A1 and A2 instances of data provided the total data instances will be 6750. By applying this logic and generating all aggregated patterns, all possible combinations for anyone using any model of these four appliances are covered. As the dataset lacks apparent power, we calculate it using equation (3,4). This aids in better training of the model by incorporating more relevant features, given the large number of overall classes in this approach.

Real Power is combined using equation (1), where the summation of all powers from different appliances is used in the aggregation of data. Reactive Power using equation (2) reactive power is aggregated directly using the summation of several appliances. Since power factor couldn't be added directly, its obtained using the relational formula of power which is obtained from equations (1,2,3,4). When all the powers are obtained, the aggregated powerfactor is calculated using the general formula from equation (5)

$$P_{total} = \sum_{k=0}^n P_k \quad (1)$$

$$Q_{total} = \sum_{k=0}^n Q_k \quad (2)$$

$$S_{total} = \sqrt{P_{total} + Q_{total}} \quad (3)$$

$$S_{total} = \sqrt{(\sum_{k=0}^n P_k)^2 + (\sum_{k=0}^n Q_k)^2} \quad (4)$$

$$\phi_{total} = \frac{\sum_{k=0}^n P_k}{\sum_{k=0}^n S_k} \quad (5)$$

In Table I below, the samples of raw data collected from listed appliances are shown. Each row represents a single timestamp, with measurements recorded for different appliances. One notable aspect observed in the table is the variation in the load characteristics, ranging from predominantly inductive loads to those exhibiting slight capacitive behavior. Accurate load disaggregation and appliance identification in NILM systems depend on an understanding of these distinctions.

The NILM measurement data for the identical group of appliances shown in Table I are reported in Table II. It is crucial to remember that Table II showcases just one of the numerous combinations possible in NILM analysis. In this particular combination, highlighted in green, the characteristics of the two appliances are aggregated, while the other two appliances are turned off, as indicated in gray. Such combinations allow for the examination of various scenarios to better understand the behavior of individual appliances and their collective impact on the overall power consumption profile.

TABLE I. SAMPLES OF MENTIONED APPLIANCES

Appliance	Single Sample from Dataset		
	Real Power	Reactive Power	Apparent Power
Coffee Machine	0.104	-3.226	3.227675944
Fridge	48.551	18.53	51.96691737
Kettles	955.803	-1.035	955.8035604
Laptops	36.115	-13.426	38.52986765

^a Note: The raw data samples in the table represent measurements recorded at single timestamps for various appliance types. The data for each appliance type is from a certain model of that appliance and doesn't represent the total number of appliance models in that appliance type category. Different load characteristics, such as inductive or capacitive, might affect the overall profile of power consumption. Data collected from [8].

TABLE II. NEW DATA GENERATED BASED ON NILM ALGORITHM

Appliance	Data Generated For NILM		
	Real Power	Reactive Power	Apparent Power
Coffee Machine	0.104	-3.226	3.227675944
Fridge	0	0	0
Kettles	0	0	0
Laptops	36.115	-13.426	38.52986765
Total	36.219	-16.652	41.75754359

^a Note: Table II displays measurement values for NILM analysis, representing one of several possible combinations of appliance states. Green highlights denote active appliances, while gray indicates inactive ones. The data presented offers insight into the aggregated characteristics of appliances under different operational scenarios.

The continuous graphical representation of Table I and II for the Coffee Machine and Laptop including their loads separately is represented in Figure 4.

This data represents just one instance among many. Among the listed appliances only one of their models is taken into account, as demonstrated in Table II. To streamline the process, for this research, an algorithm is developed that's

capable of directly aggregating data according to the proposed algorithm and feeding it to the model without additional processing or storage requirements as shown in Figure 5. Through this approach, we anticipate generating 6750 unique combinations of aggregated appliances.

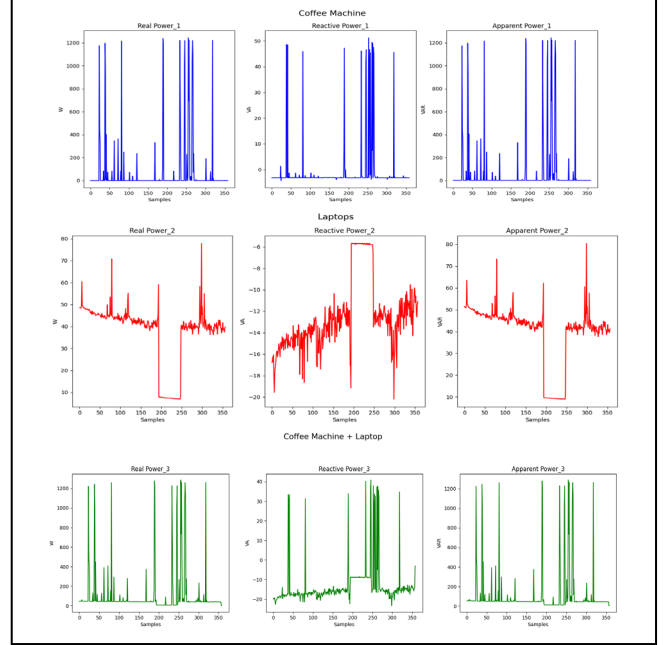


Figure 4: Appliance Load Combinational Graphs

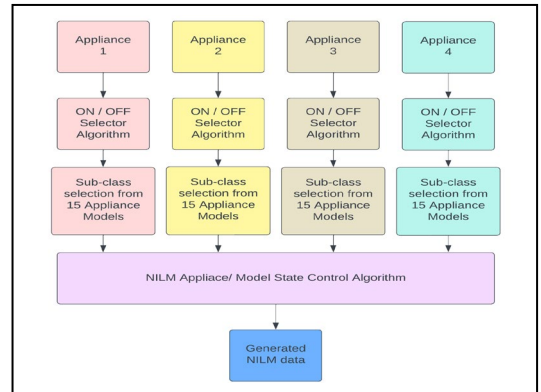


Figure 5: Auto appliance on/off algorithm for training of NILM system

B. Models From AEON Toolkit

Designed to make machine learning research and testing easier, the AEON toolkit [11] is a flexible python package. Time series analysis, picture classification, and other techniques are supported by AEON, which provides a large selection of models and algorithms. Both practitioners and researchers may easily investigate and use cutting-edge machine learning techniques. Research in a variety of areas, including data analysis, pattern recognition, and beyond, benefits greatly from the use of AEON because of its versatile and strong capabilities.

The models utilized in this research comprise those found in the AEON toolkit [11], which offers a variety of models for different approaches. However, for this research, only the time series classification models are considered. The types of classification techniques used by models in the AEON toolkit are listed below:

- Feature Based
- Convolution Based
- Distance Based
- Interval Based
- Shapelet Based

Carl H. Lubba et al. [22] utilized 7700 features from the HCTSA feature extractor. After filtering the raw HCTSA features and applying a threshold to their calculated p-values through classification problems, the feature count was reduced to 4791. These filtered features underwent additional classification tasks, and by arranging them according to their accuracies and applying a threshold value, the top-performing 22 features were identified, resulting in the creation of the CATCH22 Classifier. The feature extraction process for CATCH22 is summarized in Figure 6.

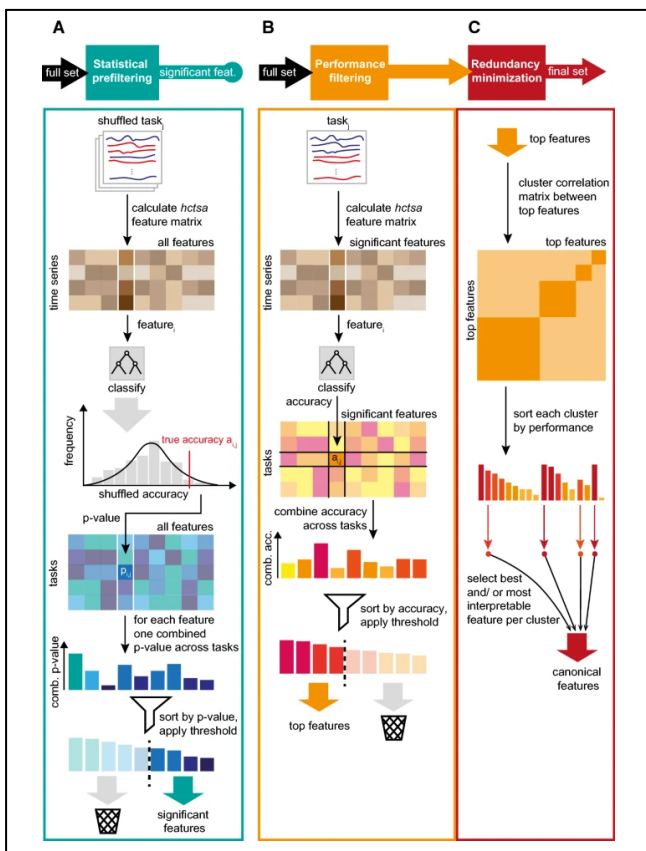


Figure 6: Process of CATCH22 feature selection [22]

Utilizing these techniques researchers have developed and modified numerous models some of them tested for this research are; the CATCH22 Classifier (Canonical Time-series Characteristics) [22], ROCKET Classifier (Random Convolutional Kernel Transform) [23], Signature Classifier, Arsenal Classifier [24], Supervised Time Series Forest Classifier, Time Series Forest Classifier, Canonical Interval Forest Classifier [25], K Neighbor Time Series Classifier [26], RDST Classifier (Random Dilated Shapelet Transform) [27], CATCH22 [22], an acronym for CAnonical Time-series CHaracteristics is a very efficient classifier for efficiently classifying time series data. The working principle of CATCH22 mostly revolves around selection of best

performing features from the Highly comparative time-series analysis (HCTSA) [28] MATLAB toolbox. This toolbox has the capability of extracting massive number of features. Its total capacity constitutes of over 7700 features in which each feature has a unique scientific formulation.

These finalized 22 features are computationally less expensive than the total 7700 features of HCTSA. The CATCH22 features computationally take around 0.5sec for a total of 10,000 samples which is comparably 1000 times faster than the total computation of HCTSA from MATLAB.

These extracted features are then fed into a chosen classifier. CATCH22 itself is not a model but rather it's a feature extraction method. Its 22 features are extracted from the input time series data and after the transformation is applied to the data the CATCH22 Classifier, which is a wrapper provided with the Canonical feature extraction method, uses the random forest classifier from scikit-learn.

IV. RESULTS AND DISCUSSION

The ACS-F2 dataset was utilized to create a NILM dataset specifically for this study, focusing on four selected appliances of our daily use, mimicking real-world scenarios where multiple appliances may be in use concurrently. Each appliance in the ACS-F2 dataset encompasses data from 15 different models from various manufacturers, providing a diverse range of appliance characteristics. The selection of the ACS-F2 dataset was driven by its low-frequency sampling, which aligns with the sampling frequency of many commercial energy meters. This compatibility ensures that the generated NILM dataset closely resembles real-world energy consumption patterns. The confusion matrix for CATCH22 on the test aggregated dataset could be seen in Figure 7.

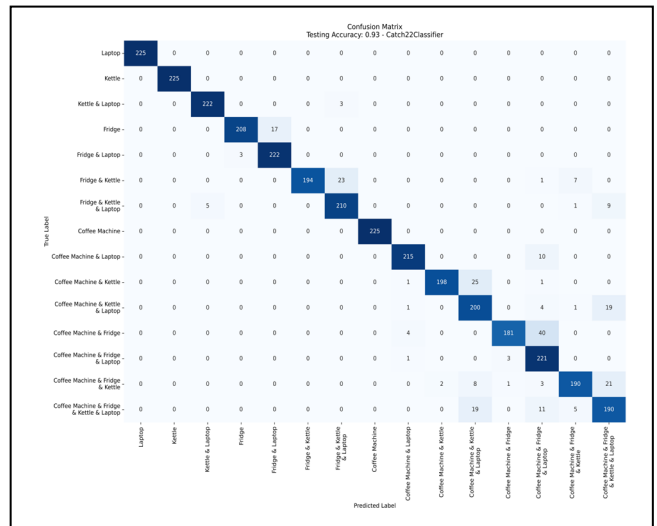


Figure 7: Confusion Matrix for CATCH22 Classifier

A set of machine learning classifiers were utilized in this study and making use of best performing time series classifier. Amongst all listed Classifiers the CATCH22 classifier outperformed all of the other ones. That's because of its operational methodology of extracting the best performing 22 features which results in its exceptional performance. The accuracy for the total of 3375 test samples of aggregated data comes out to be 92.1%. The precision, recall and F1 score being 93%, 92% and 92% respectively. The performance comparison for the different models

benchmarked on the proposed NILM dataset are included in Table III.

TABLE III. PERFORMANCE COMPARISON TABLE

Classifier	Computed Results			
	Accuracy	F1 Score	Precision	Recall
ROCKET	0.85	0.86	0.87	0.86
Arsenal	0.86	0.85	0.86	0.85
CATCH22	0.92	0.92	0.93	0.92
Signature	0.73	0.73	0.76	0.73
KNeighbours Time Series	0.58	0.57	0.62	0.58
Canonical Interval Forest	0.84	0.85	0.86	0.84
Supervised Time Series	0.57	0.5	0.53	0.57
Time Forest Classifier	0.73	0.73	0.77	0.73

^b. Note: These values were obtained from the test set

CONCLUSION

To summarize, this study on Energy Disaggregation has significantly addressed the technology gap between energy generation and consumption. It not only gauged state of the art algorithms essential for implementing NILM but also pioneered a unique framework for generating aggregated appliance electrical patterns using disaggregated appliance data from the ACS-F2 dataset. The CATCH22 Classifier, has been effectively implemented for Energy Disaggregation using the proposed dataset, achieving a staggering accuracy of 92% on the test set. This approach is the first of its kind on the ACS-F2 dataset as per our knowledge. The significance of this research lies in the compatibility of the dataset's sampling frequency with that of commercially used energy meters, ranging from 1 Hz to 10 Hz. Moreover, by utilizing disaggregated signals from appliances, it could become feasible to analyze their electrical or mechanical health in the future to enable predictive maintenance. This study concludes by demonstrating how successfully the proposed data generation methodology and the CATCH22 classifier can be used together for NILM. The study emphasizes the usefulness of NILM in situations found in the real world, highlighting their effectiveness and possible influence on energy management strategies.

REFERENCES

- Jones, J.S. *Smart metering and demand response continue to increase in US*. Dec 22, 2023; Available from: <https://www.smart-energy.com/industry-sectors/smart-meters/smart-metering-and-demand-response-continue-to-increase-in-us/>. Accessed 3 Apr. 2024.
- Lin, Y.-H. *Design and Implementation of an IoT-Oriented Energy Management System Based on Non-Intrusive and Self-Organizing Neuro-Fuzzy Classification as an Electrical Energy Audit in Smart Homes*. Applied Sciences, 2018. **8**, DOI: 10.3390/app8122337.
- Guinard, D., M. Weiss, and V. Trifa. *Are you energy-efficient? sense it on the web*. in *Adjunct Proceedings of Pervasive*. 2009.
- Hart, G. W., Kern Jr, E. C., & Schwegge, F. C. (1989). *U.S. Patent No. 4,858,141*. Washington, DC: U.S. Patent and Trademark Office.
- Hart, G.W., *Nonintrusive appliance load monitoring*. Proceedings of the IEEE, 1992. **80**(12): p. 1870-1891.
- Liang, J., et al., *Load Signature Study—Part II: Disaggregation Framework, Simulation, and Applications*. IEEE Transactions on Power Delivery, 2010. **25**(2): p. 561-569.
- Wolpert, D.H. and W.G. Macready, *No free lunch theorems for optimization*. IEEE Transactions on Evolutionary Computation, 1997. **1**(1): p. 67-82.
- Ridi, A., C. Gisler, and J. Hennebert. *ACS-F2 — A new database of appliance consumption signatures*. in *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPar)*. 2014.

- Gisler, C., et al. *Appliance consumption signature database and recognition test protocols*. in *2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)*. 2013.
- Salihagić, E., J. Kevric, and N. Dođru. *Classification of ON-OFF states of appliance consumption signatures*. in *2016 XI International Symposium on Telecommunications (BIHTEL)*. 2016.
- aeon. *aeon-toolkit*. 2018; Available from: www.aeon-toolkit.org. Website accessed on: 3/Apr/2024
- Middlehurst, M., Schäfer, P., & Bagnall, A. (2023). *Bake off redux: A review and experimental evaluation of recent time series classification algorithms*. *ArXiv*. /abs/2304.13029
- Ruano, A., et al. *NILM Techniques for Intelligent Home Energy Management and Ambient Assisted Living: A Review*. Energies, 2019. **12**, DOI: 10.3390/en12112203.
- Yu, L., et al., *Nonintrusive appliance load monitoring for smart homes: recent advances and future issues*. IEEE Instrumentation & Measurement Magazine, 2016. **19**(3): p. 56-62.
- Kelly, Daniel. *Disaggregation of Domestic Smart Meter Energy Data*. 2016, <https://api.semanticscholar.org/CorpusID:57980639>. Accessed 3 Apr. 2024.
- Puente, C., et al. *Non-Intrusive Load Monitoring (NILM) for Energy Disaggregation Using Soft Computing Techniques*. Energies, 2020. **13**, DOI: 10.3390/en13123117.
- Kelly, J. and W. Knottenbelt, *The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes*. Scientific Data, 2015. **2**(1): p. 150007.
- Hebrail, Georges and Berard, Alice. (2012). *Individual Household Electric Power Consumption*. UCI Machine Learning Repository. <https://doi.org/10.24432/C58K54>.
- Angelis, G.-F., et al., *NILM applications: Literature review of learning approaches, recent developments and challenges*. Energy and Buildings, 2022. **261**: p. 111951.
- Kelly, J. and W. Knottenbelt, *Neural NILM: Deep Neural Networks Applied to Energy Disaggregation*, in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. 2015, Association for Computing Machinery: Seoul, South Korea. p. 55–64.
- Salerno, V.M. and G. Rabbeni *An Extreme Learning Machine Approach to Effective Energy Disaggregation*. Electronics, 2018. **7**, DOI: 10.3390/electronics7100235.
- Lubba, C.H., et al., *CATCH22: CAnonical Time-series Characteristics*. Data Mining and Knowledge Discovery, 2019. **33**(6): p. 1821-1852.
- Dempster, A., F. Petitjean, and G.I. Webb, *ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels*. Data Mining and Knowledge Discovery, 2020. **34**(5): p. 1454-1495.
- Middlehurst, Matthew, et al. *Bake off Redux: A Review and Experimental Evaluation of Recent Time Series Classification Algorithms*. *ArXiv*, 2023, /abs/2304.13029. Accessed 3 Apr. 2024.
- M. Middlehurst, J. Large and A. Bagnall, "The Canonical Interval Forest (CIF) Classifier for Time Series Classification," *2020 IEEE International Conference on Big Data (Big Data)*, Atlanta, GA, USA, 2020, pp. 188-195, doi: 10.1109/BigData50022.2020.9378424.
- Cunningham, Padraig, and Sarah J. Delany. *K-Nearest Neighbour Classifiers: 2nd Edition (with Python Examples)*. *ArXiv*, 2020, <https://doi.org/10.1145/3459665>. Accessed 3 Apr. 2024.
- Guillaume, A., Vrain, C., & Wael, E., *Random Dilated Shapelet Transform: A New Approach for Time Series Shapelets*. *ArXiv*, 2021.
- Fulcher, B.D. and N.S. Jones, *hctsa: A Computational Framework for Automated Time-Series Phenotyping Using Massive Feature Extraction*. Cell Syst, 2017. **5**(5): p. 527-531 e3.

Relevance Classification of Flood-Related Tweets using XLNET Deep Learning Model

Aneela Habib

Department of Computer
Systems Engineering
University of Engineering and
Technology Peshawar
aneelahabib2@gmail.com

Yasir Saleem Afridi

Department of Computer
Systems Engineering
University of Engineering and
Technology Peshawar
yasirsaleem@uetpeshawar.edu.pk

Madiha Sher

Department of Computer
Systems Engineering
University of Engineering and
Technology Peshawar
madiha@uetpeshawar.edu.pk

Tiham Khan

Department of Computer
Systems Engineering
University of Engineering and
Technology Peshawar
tihamkhan3b@gmail.com

Abstract:

Floods, being among nature's most significant and recurring phenomena, profoundly impact the lives and properties of tens of millions of people worldwide. As a result of such events, social media structures like Twitter often emerge as the most essential channels for real-time information sharing. However, the total volume of tweets makes it hard to manually distinguish between those relating to floods and those that are not. This poses a large obstacle for responsible government officials who need to make timely and well-knowledgeable decisions. This study attempts to overcome this challenge by utilizing advanced techniques in natural language processing to effectively sort through the extensive volume of tweets. The outcome we obtained from this process is promising, as the XLNET model achieved an extraordinary F1 rating of 0.96. This high degree of overall performance illustrates the model's usefulness in classifying flood-related tweets. By leveraging the abilities of the XLNET model, we aim to provide a valuable guide for responsible governance, aiding in making timely and well-informed choices during flood situations. This, in turn, will assist reduce the impact of floods on the lives and property-affected communities around the world.

Keywords: Text classification, LSTM, Multi-head Attention, Flood, Tweets

1. INTRODUCTION

Natural disasters, like floods, can cause excessive destruction to communities and residences. In the contemporary generation, social media platforms have emerged as treasured resources of statistics during and after such disasters. Twitter, in particular, plays an important position in disseminating real-time updates. However, the full extent of tweets generated through these events can overwhelm responsible governance, making it hard to identify pertinent information.

To deal with this problem, we advocate a text class framework making use of a machine learning model known as XLNET. XLNET is one of the latest deep learning models well-known for its splendid performance across various natural language processing (NLP) responsibilities. By leveraging XLNET's competencies, our goal is to construct an effective solution for classifying flood-related tweets. This framework could be an effective tool that can be utilized by government agencies, helping them quickly identify the relevant information amidst the vast amount of information on Twitter, thereby helping to enable appropriate and timely decision-making.

2. LITERATURE REVIEW

Text classification is a key task in the broader framework of NLP, which aims at grouping text into predefined classes or categories. This task constitutes various jobs, such as spam detection, sentiment analysis, and theme categorization, among others. It is noteworthy, that there have been significant studies carried out on detecting natural disasters and the usage of social media and satellite imagery [2]. In recent years, social media structures, specifically Twitter, have emerged as precious assets of facts, in particular, especially during times of crisis [3].

Several studies have been carried out on taking benefit from social media for disaster reaction and management. For instance, Palen et al. (2010) [4] analyzed Twitter usage during the 2009 Red River Valley flood, identifying various sorts of tweets, together with situational recognition updates, legitimate alerts, requests for help, and emotional support. Similarly, Imran et al. (2015) [5] tested Twitter's function during the 2013 Colorado floods and highlighted its significance in supplying precious facts for disaster reaction and control.

The utility of NLP techniques in analyzing social media statistics for the duration of disasters has been widespread. Caragea et al. (2011) [6] applied machine learning methods to categorize catastrophe-related tweets, attaining an F1 score of 0.79 on a dataset comprising 9000 tweets from three exclusive disasters.

Furthermore, the use of gadget-mastering models for flood detection, as demonstrated in Flood Detection in Urban Areas Using Satellite Imagery and Machine Mastering [7], signifies the ability of such procedures to improve situational attention in the course of emergencies. Extracting geographically rich know-how from microtexts like tweets becomes important for location-based structures in emergency services to correctly respond to diverse natural and man-made disasters, including earthquakes, floods, pandemics, vehicle accidents, terrorist attacks, and shooting incidents [8].

3. METHODOLOGY

Figure 1 represents the block diagram of the overall methodology used in this study to illustrate the sequential steps involved in achieving our objectives. These steps are explained in the following subsections.



Figure 1: Block diagram of our steps to be followed

3.1 Text Stream

In our investigation, we aim to discover the capacity of using tweets as a precious source of facts and communicate during and after disasters. To accomplish this, we rent Twitter APIs to extract relevant tweets associated with particular catastrophe events.

For the purpose of model training, we've amassed and processed a substantial dataset of 5313 tweets. The preprocessing step entails cleansing the data, removing noise, and transforming the text into a suitable format that can be used for training the model efficiently.

3.2 Text Pre-Processing

In the sphere of NLP, the pre-processing stage holds massive significance as it includes important tasks consisting of data cleansing and transformation. The objective here is to convert the raw text statistics into a format suitable for training the model [15].

3.3 Text Normalization

Normalization of the text (the next stage in natural language analysis) normalizes the text information which later on is used to run the analysis. It is surrounded by a chain of operations that involves punctuation removal, lower-casing of all text, and special characters removal.

By using these normalization techniques, we attain a greater uniform representation of the textual content, facilitating powerful evaluation and enabling NLP models, including advanced ones like XLNET, to recognize the underlying semantic content. This system lays the foundation for robust model training, complementing the overall accuracy and comprehension of the language used within the data.

3.4 Tokenization

Tokenization can be seen as the foundational technique in NLP. It contributes a lot by working with texts such as tweets. It involves segmenting text into tokens, which are the basic units upon which most NLP processors operate, facilitating various tasks including morphological analysis.

Therefore, tokenizing our tweet dataset provides the necessary structure and training for the required analysis and processing of the text. Hence, we proceed to the next step which is very essential for an actualization of the various roles within the NLP tasks and the extraction of meaningful information from the Tweets posted.

4. MODEL TRAINING

After preprocessing, the next step taken was the training of processed data. The choice of a 2-way type model was motivated by [9].

4.1 XLNET MODEL:

XLNET is a complex transformer-based language model that employs a permutation-based training approach to efficiently capture dependencies amongst all tokens within an input series. Interestingly, the architecture of XLNET is

comparable with many transformer-based models like BERT. However, these models have an additional feature that sets them apart from their counterparts. Besides the above-mentioned attributes performance, this language processing model can be identified as unique and indelible in the field of NLP [10].

XLNET distinguishes itself from BERT in phrases of its training goal. While BERT relies upon left-to-right or masked left-to-proper training to predict masked tokens in a set order, XLNET employs a sequence-based training method [13]. In this method during learning, XLNET considers all viable changes of the input tokens in preference to being constrained to a specific order. As a result, XLNET captures two-way context and dependencies among all tokens in the enter dataset, main to a greater complete and nuanced learning of the language.

The architecture of XLNET is a hit fusion of the strengths discovered in automated fashions like Transformer-XL and masked language fashions like BERT. This composition boosts XLNET as the top neural language representation model, thus achieving a range of activities from NLP tasks at large [12]. This approach that shares both the input and the output contexts reinforces its ability to handle sophisticated language modalities, hence it is a proper tool in the realm of multi-lingual natural processing.

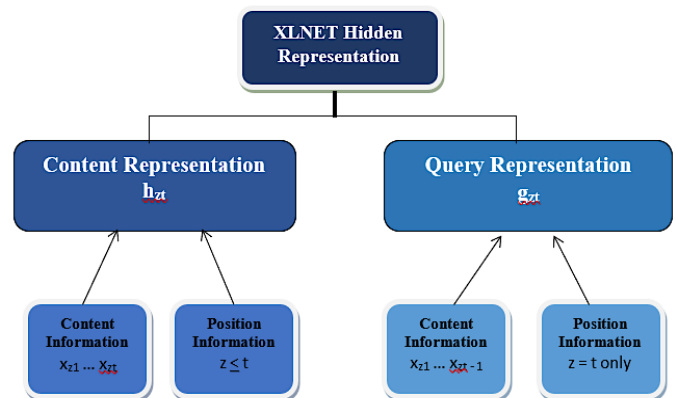


Figure 2: XLNET with 2 sets of hidden representations

This self-attention structure is designed as an improvement over the conventional transformer version to overcome its limitations. The first component is the content movement representation, which is similar to the same old self-attention in Transformer. It considers both the content $x_{z,t}$ and the position information z_t of the target token within the input sequence as depicted in Figure 2. By doing this, the model can capture even more relevant interactions between objects and topics within the context.

Another approach, termed illustration, is for this model to function as a substitute for BERT's [MASK] mechanism. It employs query circulation attention to predict the target token $x_{z,t}$ based solely on its positional information, excluding the actual content. The model has entry to the placement information of the goal token and the context statistics before that token for making predictions.

By incorporating those two types of self-attention mechanisms, the Two-Stream Self-Attention architecture pursuets to better seize both local and global dependencies

in the input sequence, thereby enhancing the accuracy of predictions for the target tokens.

The Two-Stream attention mechanism in XLNET leads to a target-conscious prediction distribution. The key difference between XLNET and BERT lies in their pretraining strategies. Unlike BERT, XLNET does not rely on data corruption and masking during pre-training. By avoiding BERT's masking limitations, as referred to earlier in the autoencoder model, XLNET achieves its target focus and offers improved overall performance.

XLNET improves its ability to capture lengthy-variety dependencies in comparison to RNNs and general Transformers by incorporating Transformer-XL's relative encoding scheme and section repetition mechanism. The relative positional encoding is implemented primarily based on the original sequence, while the segment-level repetition mechanism prevents context fragmentation and enables the reuse of past sentence segments with new ones, thereby maintaining long-term period context. By including phase-stage repetition in hidden states, XLNET stands apart from Transformer, resulting in enhanced overall performance and better handling of remote dependencies.

XLNET integrates Transformer-XL into its improved training framework, permitting the incorporation of the repetition mechanism. This mechanism is used in the proposed permutation setting of XLNET to reuse hidden states from previous segments. However, the factorization order within the permutation from preceding segments is not cached and reused in future computations. Only the content representation of the section is retained in the hidden states, allowing for efficient handling of long-range dependencies without the need to store and reuse the whole factorization order.

LSM and Multi-head Attention Layer: This layer is comprised of two sub-layers working together:

LSM (Likely Long Short-Term Memory): This is a type of recurrent neural network (RNN) adept at handling sequential data like text. LSTMs can capture long-term dependencies within sentences, crucial for tasks like sentiment analysis or machine translation.

Multi-head Attention: This is a mechanism that allows the model to focus on specific parts of the input text that are most relevant to the current processing step. It essentially helps the model pay attention to different aspects of the input simultaneously.

Output Layer: This layer takes the processed data from the previous layer and generates the final output, which could be a variety of things depending on the specific NLP task. Examples include classification which is classifying the sentiment of a text review (positive, negative, neutral), machine translation which is converting text from one language to another and text summarization which is creating a concise summary of a lengthy piece of text.

Overall, Figure 3 represents a simplified illustration of how a deep learning model can be structured to process and analyze textual data. By leveraging LSTMs for capturing long-term dependencies and multi-head attention to focus on relevant parts of the input, the model can learn complex patterns within the language and perform various NLP tasks.

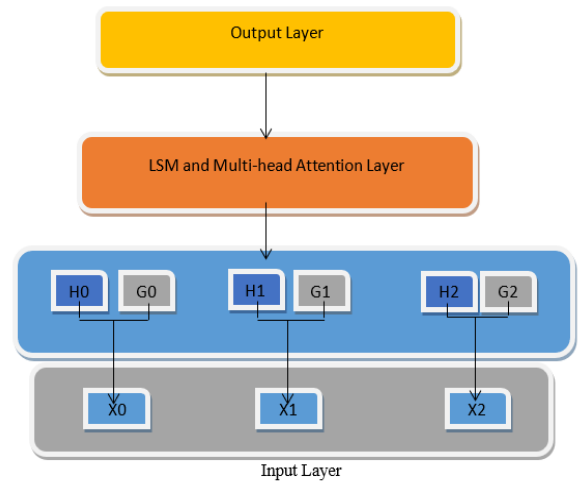


Figure 3: Layers of XLNET

4.2 MODEL STRUCTURE

The XLNET model architecture can be divided into four major parts. The initial stage involves processing the entered text, where it is tokenized, meaning it is divided into individual units or words for further analysis. This process essentially operates with word vectors, resulting in low-dimensional representations of the textual input being used.

Subsequently, the XLNET layer dissects the text representation and further extracts features. Such operations capture crucial patterns from the input text and then the result is transferred into Long Short-Term Memory (LSTM) and Multi-head Attention layers.

The LSTM is a type of recurrent neural network designed to capture context information from input sequences. It achieves this by using input gates, forget gates, and output gates to control the flow of information. The LSTM selects relevant memory feature vectors and integrates them to generate meaningful output.

Finally, the Multi-head Attention Layer calculates the probability of attention from multiple perspectives. This lets the model assign distinctive weights to the extracted feature vectors, essentially giving greater importance to certain parts of the input text. By doing so, the model aims to acquire effective text classification.

5. DATA SPLITTING

We split our dataset into two components, training and validation sets, as shown in Figure 4.

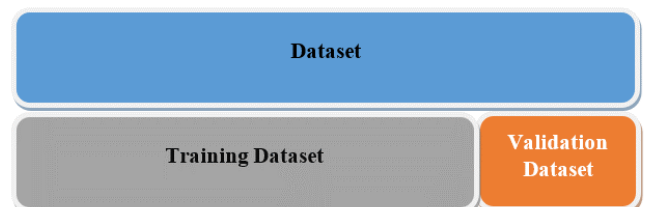


Figure 4: Splitting of the datasets

5.1 Training Dataset:

This is a subset of the data used to train the model. The model learns patterns and relationships from this data.

5.2 Validation Dataset:

This is another subset of the data used to assess the model's performance on unseen data. It's essential to prevent overfitting, which occurs when a model performs well on the training data but poorly on new data.

The training dataset constituted 80% of the complete dataset, at the same time as the validation dataset consists of 20% of the dataset. These separate datasets were used to train the model, investigate its performance at some point of validation, and eventually, compare its generalization on the test set.

6. TESTING

To evaluate the effectiveness of our model in classifying tweets as it should be, we tested it on a set of tweets. The model achieved an outstanding F1 score of 0.96 on the test dataset. This high F1 score shows that the model demonstrates sturdy overall performance and reliability in classifying tweets with an excessive stage of accuracy.

7. RESULTS AND DISCUSSION

From the results presented in Table 1, it is evident that our model achieved an accuracy of 94.16% on the test set. This accuracy rate reflects the model's ability to correctly classify tweets with a high level of precision. Looking ahead, we intend to further improve the model's performance by fine-tuning various parameters. Despite the impressive accuracy achieved, we also evaluated the model's performance using the F1 score metric, which demonstrated an outstanding score of 96%. This high F1 score indicates a robust performance in both precision and recall, underscoring the effectiveness of our model in classifying tweets accurately.

Table 1: Results of XLNET Model on test set

Method	F1 Score	Accuracy
XLNET	0.960	0.9416

8. CONCLUSION

In our paper, we tackled the significant challenge of identifying relevant tweets in the after a flood occurrence, recognizing the important role of timely and accurate information for informed decision-making by governmental authorities. Leveraging advanced techniques in NLP, particularly XLNET, enabled us to efficiently sift through the immense volume of tweets generated during such events.

Our findings underscore the potential of deep learning models, such as XLNET, in addressing complex challenges at the intersection of natural disasters and social media. By effectively harnessing these technologies, we have demonstrated the capability to extract valuable insights from large-scale social media data, thereby facilitating more effective disaster response strategies.

Furthermore, our research highlights the importance of ongoing innovation in the field of NLP, particularly in the context of disaster management and response. As the volume and complexity of social media data continue to

grow, leveraging cutting-edge techniques like XLNET will be essential for improving the efficiency and accuracy of information extraction and decision-making processes in disaster scenarios.

REFERENCES

- [1] N. Pourebrahim, "Understanding communication dynamics on Twitter during natural disasters," sciencedirect.com, pp. 10-20, July 2019
- [2] N. Said, "Natural disasters detection in social media and satellite imagery," Springer.com, Vols. 78, pages31267–31302 , p. 80, 2019
- [3] W. Ahmed, "Using Twitter as a data source:," google.com, vol. CC BY 3.0, 2017
- [4] K. Starbird, "Microblogging during two natural hazards events: what twitter may contribute to situational awareness," acm.org, 2010
- [5] M. Imran, "Towards a Data-driven Approach to Identify Crisis-Related Topics in Social Media Streams," acm.org, 2015.
- [6] D. C. & X. O. Su Zhang, "An Empirical Study on Using the National Vulnerability Database to Predict Software Vulnerabilities," psu.edu, 2011.
- [7] A. H. Tanim, "Flood Detection in Urban Areas Using Satellite Imagery and Machine Learning," 2022
- [8] S. Vieweg, "Microblogging during two natural hazards events: what Twitter may contribute to situational awareness," acm.org, 2010.
- [9] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Advances in neural information processing systems 32 (2019).
- [10] Luo Pengcheng, Wang Yibo, Wang Jimin. Research on automatic classification of literature subjects based on deep pre-trained language model [J]. Journal of Information Science, 2020(10).
- [11] Yu Shujuan.Attention-based LSTM, GRU, and CNN for short text classification[J].Journal of Intelligent&Fuzzy Systems,2020,39(1): 333-340.
- [12] Liang Xiaobo, Ren Feiliang, Liu Yongkang, et al. Machine reading comprehension model based on double-layer Self-attention [J]. Chinese Journal of Information, 2018, 32(10): 130-137.
- [13] Zhang Chong. Research on Text Classification Technology Based on Attention-Based LSTM Model [D]. Nanjing: Nanjing University, 2016.
- [14] Jason D. M. Rennie. 1999. Improving Multi-class Text Classification with Naive Bayes. Master's thesis. Massachusetts Institute of Technology. <http://qwone.com/~jason/papers/sm-thesis.pdf>
- [15] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In Advances in neural information processing systems. 5754–5764.

Exploring Political Emotions: Sentiment Analysis of Urdu Tweets

Ehtisham Ur Rehman

Department of Computer Science
UET, Peshawar, Pakistan
ehtishamrehman@uetpeshawar.edu.pk

Muhammad Nouman Khan

Department of Computer Systems
Engineering UET Peshawar, Pakistan
eng.noumankhann@gmail.com

Najam Aziz

Department of Computer Systems
Engineering UET Peshawar, Pakistan
najamkhattak@gmail.com

Abstract—This research is a multi-text categorization based on a collection of Pakistani political texts. The major goal of this research is to use Natural Language Processing (NLP) and Machine Learning classification models to categorize multi-text for Urdu. Political tweets from 13 different Pakistani famous leaders were collected for this research. These politicians make use of the platform to promote themselves and engage with their supporters. To analyze the model accuracy the desired dataset is divided into six categories which have been composed of their official Twitter account. We also collect top trends from Pakistan and around the world to examine current trends regularly. In the proposed research, the major political corpus data comprises of 1300+ tweets in the Urdu language, encompassing political policies, campaigns, opinions, and so on.

Sentiment analysis is an essential component of every deep learning approach. For that, we have used the deep learning approach i.e. sentiment analysis of the politician since it provides insight their moods and views on a certain topic. Furthermore, text corpus pre-processing is conducted utilizing NLP techniques, such as data cleaning, data balancing, and stop word removal. TF-IDF is used as word filtering for feature extraction count vectors. Machine Learning classification algorithms such as SVM, Decision Tree, XGboost, Random Forest, and for implementation of neural network we have used Word2vector.

1. Introduction

Sentiment analysis in politics uses Natural Language Processing (NLP) and linguistic computing to understand how people feel about a political term [1]. Textual data mining is used to identify linguistic patterns that reveal an individual's emotional, affective, cognitive state, attitude toward something or someone, personality traits, and other psychological constructs. Sentiment analysis can be used to gauge the response of masses towards a politician's political attitude on any topic. For example, if people are satisfied with the stance of a political leader on a specific policy matter, the same is expressed in their views which may be propagated through various mediums such as interviews, facebook posts, tweets etc. Gathering and analyzing political sentiments of masses is important, as public opinion has a significant impact

on elections and policy. It enables citizens' views to be analyzed. Anger and delight are the two most commonly expressed emotions in politics. These feelings can be found on both sides of the political aisle, but they are often aimed at politicians [1][2].

Looking at the classical methods which were used previously were mainly based on traditional surveys. The purpose of conducting political surveys is to understand registered voters' opinions and emotions. Many different groups, including political parties, Political action committees, consultants, council members, state departments, local school districts, and candidates, use questionnaires of this type. Political survey questions can be used to learn more about base of support and the wants and requirements of the general populace. By asking voters these questions, political campaigns and activist groups can better understand their voters' priorities and develop policies that are more likely to win their support. In addition, such surveys can better understand the political environment and political campaign strategies and increase support from potential voters [3]. However, extensive surveys need considerable money and time. Political parties now emphasize the use of social media as a better and cheaper alternative for collection and analysis of political sentiments. Social media provides a power tool for monitoring and analyzing political sentiments/opinions, resulting in an increased focus on social media platforms from political parties. Recent years have seen a dramatic increase in the use of Twitter as a platform from which political opinions may be shared and a conversation with ordinary citizens can occur.

In recent times, text mining has become more relevant due to the abundance of various data kinds from various sources, primarily in the form of semi-structured and un-structured data. The main objective of text mining is to enable users to extract data from diverse sources and then carry out a variety of activities, including data retrieval and classification (supervised, unsupervised, and semi-supervised), data mining, and NLP for automatic classification [4]. Political sentiment analysis has been the subject of multiple research projects in various languages, including English, German, Chinese, and others [5]. The 2009 German federal election was the primary focus as discussed by the Tumasjan et al. [6]. Twitter was used to monitor public opinion and predict the outcome of the election. In the neighborhood of a hundred thousand tweets, they

looked for mentions of politicians or political parties. They analyzed the tweets for sentiment using the LIWC2007 tool [7]. LIWC is a reliable text analysis program designed to extract emotions, ideas, and personality from their actual text. They reasoned that the more tweets a candidate had, the better their chances of being elected. Online sentiment analysis was performed by [8] to predict the percentage of votes each contender will receive in the 2011 Singapore presidential election. Using Twitter data, [9] demonstrated a real-time sentiment application system for the 2012 US presidential election.

According to Ringsquandl et al. [5], the campaign on Twitter of the presidential candidates from the Republican Party was examined. The authors of the study came to the conclusion that politicians and the topics they address have a stable semantic link by combining the frequency of noun phrases with their PMI value and placing a constraint on aspect extraction. This new notion was presented in their study. Arabic text classification using the WEKA software was utilised to centre on the 2012 Egyptian presidential elections [10]. The scientists concluded that, of the various methods they tested, the Naive Bayesian approach was the most accurate and error-free. The purpose of studying Hindi tweets prior to India 2016 general state election was to enable informed prediction-making [11]. They extracted 42,235 Hindi tweets from the Twitter Archiver, analysed them using the SVM, dictionary-based, and Naive Bayes approach, and categorised them as “positive” “negative” or “neutral.” Based on the results, the SVM predicted that the BJP had a 78.4 percent chance of winning additional seats in the general election as a result of the strong emotional response they received in their tweets. The Indian National Congress came in second with 26 out of 126 constituencies in the 2016 general election, while the Bharatiya Janata Party (BJP) won 60. This was significantly higher than any other political party [11].

Urdu is categorized as an Indo-Aryan language and is spoken extensively throughout South Asia. In addition to being the official language of Pakistan, it is also one of 22 languages on India’s schedule that enjoy de jure official status. Nepal too has its own regional dialect of Urdu. Over 70 million people use it as their first language and over 100 million people use it as a second language, mostly in Pakistan and India [12].

Even though English is the de facto social media language, people all around the globe still prefer to keep their words to themselves. Social media users ought to have the ability to communicate in a variety of languages, including Urdu, in addition to English. A major language in South Asia, Urdu is spoken by a huge number of native speakers. The absence of a conventional writing system in Urdu, however, makes it challenging to extract valuable information from written texts. Urdu sentiment analysis is still in its early stages, and the field of digital linguistics is booming, although studies on the language are few and far between [13][14]. This study lays the groundwork for future research into the feasibility of using Twitter to assess public sentiment on Urdu politics. Examining the sentiments expressed in the tweets collected for this study allowed us to

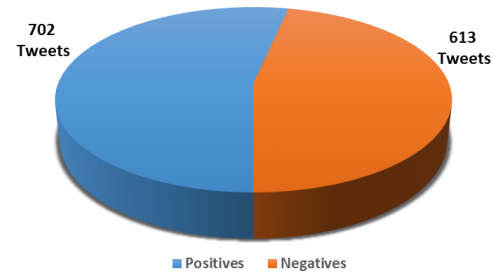


Figure 1. Total Number of Corpus (1300 Tweets)

identify their polarity, whether they were good or negative. Our contribution through this research is that we have created the “Urdu Sentiment Corpus (USC)” dataset and insights from Urdu tweets for sentiment analysis and polarity recognition. The dataset comprises tweets that cast a political shadow and present a competitive climate between political parties and the Pakistani government. We have gathered the tweets manually without using any application programming interface (API) or Python library. Approximately 1300 tweets were collected from the Urdu language from thirteen different Pakistani Politicians, with 702 entries being positive (P) and 613 being negative (n), as shown in Figure 1. This research describes visual insights into literary similarities, manifold learning, and other topics. In addition, this research proposed a Parts-of-Speech-wise analysis and applied Machine Learning (ML) algorithms, i.e., Logistic Regression, Decision Tree Classifier, Xgboost, Random Forest, and word2vector, on Urdu texts. We have investigated different approaches for developing Urdu-based sentiment analysis applications. Some of the following factors that we considered and accounted through this research are as follows: 1) Improving political sentiment classification accuracy. 2). Recognizing and classifying emotions in Urdu tweets is a challenging task. 3). Different machine learning and deep learning techniques were investigated for Urdu sentiment analysis.

1.1. Methodology

1.2. The Dataset

This section discusses the dataset’s history, data gathering process, problems, and the dataset cleaning procedure, followed by the data labeling mechanism.

1.2.1. Background. The tweets were gathered on February 17, 2022. Mr. Imran Khan, Chairman of Pakistan The reek-e-Insaf (PTI), was the 22nd Prime Minister of the Islamic Republic of Pakistan during the period given. Meanwhile, other opposition groups such as the Pakistan Muslim League (PML-N), Pakistan People’s Party (PPP), and Jamiat Ulema-e-Islam (F), among others, were criticizing the government and forming the Pakistan Democratic Organization (PDM), a political movement in Pakistan. It was formed in September

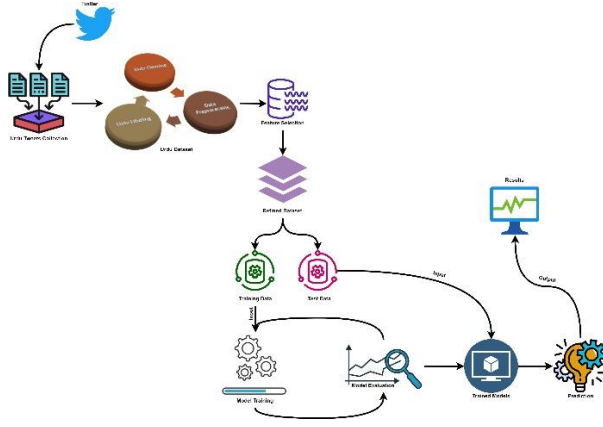


Figure 2. General Methodology

2020 as an opposition alliance against Prime Minister Imran Khan, accusing his administration of manipulating the 2018 Pakistani general election, bad governance, political persecution of opponents, and mismanagement of the economy and foreign affairs. Several dissidents from Khan’s Pakistan Tehreek-e-Insaf (PTI) also joined the battle. The campaign successfully ousted Imran Khan in a no-confidence vote on April 10, 2022. As a result, the national assembly of Pakistan chose Mr. Shehbaz Sharif as the 23rd Prime Minister of the Islamic Republic of Pakistan. Imran Khan said the U.S. was behind his ouster because he pursued an independent foreign policy and maintained close relations with China and Russia. His expulsion sparked outrage among his supporters across Pakistan. Imran Khan said that the PTI would begin the anti-government march. Although the data presented in the dataset exhibit a competitive political attire, we cannot say the political showcase dominates the entire dataset.

1.2.2. Twitter Top Trends. Based on who you follow, your interests, and your location, an algorithm creates trends that are tailored to you. To help you find the most talked-about new topics on Twitter, this algorithm finds popular topics right now, not ones that have been popular for a long or even every day. One of the factors the algorithm takes into account while ranking and recognising trends is the quantity of Tweets connected to them. Additionally, if trends and hashtags are connected to the same topic, an algorithm will group them [15].

Some trends may include a # indicator before the word or phrase. This is known as a hashtag, and it is used primarily in Tweets to identify them as related to a given topic so that others can follow the conversation in search.

In our research, we have occasionally collected top trends in Pakistan and worldwide. Figure 3 shows some people political interest, which changes with time. These trends are managed manually.

1.2.3. Data Collection and Challenges. Most previous papers extract data, i.e., tweets, through API and some

S.No	Date	Top Trends in Pakistan Trends	Top Trends in World Wide Trends/Tweets
1	17-Feb-2022	#Petrol Prices	N/A
2		#Ukraine	
3		#Prime Minister ImranKhan	
1	21-Feb-2022	#FakeNews	N/A
2		#UkraineRussiaCrisis	
1	22-Feb-2022	N/A	##UkraineRussiaCrisis
2		#Ukraine	#Putin
3		#World War 3	N/A
1	25-Feb-2022	#Ukraine	#Moscow
2		2859.3K tweets	45.8K
3		#worldwar	#WWIII
1	28-Feb-2022	3 26.1K tweets	572K
2		#PSLFinal	#UkraineRussiaWar
1	01-March-2022	122K	497K
2		#LahoreQalandars	#Japan
1	02-March-2022	54K	460K
2		#pslfinal	# ファミマ春のおむすび祭
1	05-April-2022	12K	102K
2		#ShabEMeraj	# 無料引換クーポン
1	11-April-2022	27K	108.3K
2		#sunrisewithadeel	#SOTU
1	5-April-2022	453K	549K
2		#UkraineRussiaWar	#StateOfTheUnion
1	31-Oct-21	N/A	Kansas
2		#surprise	176K
1	27-Nov-21	N/A	pt
2		30K	25K
1	2002	N/A	Lina
2		#ImranKhan	231K
1	2002	484K	Tadeu
2		#BajwaSurrender	48K
1	2002	220K	Eliezer
2		54K	54K

Figure 3. Top Trends on Twitter (Pakistan and Worldwide)

Python libraries. Our research gathers the tweets manually without using any API or python library. Approximately 13K+ tweets were collected in the Urdu language from thirteen different Pakistani Politicians. Figure4 shows the overall view of the dataset, which was collected manually from Twitter. Although it can be seen as overwhelming in size, there are significant challenges in dealing with the data, which are listed below:

Tweets	Sentiment	Tweet Date	Reply	Retweets	Likes
ٹک میں کوئی لکھی لکھی اللہ ہے جا لہ کا کتر کا کتا ہے کہ شہید رہیں جو ہے کسی	p	31-Oct-21	1712	3939	16300
کر بہت زیادہ دکھ پول اولاد کا غم سہا دینا کا سب سے بڑا دکھ ہوتا ہے اللہ بھائی اسیں	p	29	1295	2798	15200
بھروسہ بھر آئی آرٹسٹس کی منقطع شوق فی الفور واپس لی جائے بغیر جان کو ہر	n	23	826	3113	10200
۳۳ کوڑ کی ملک کو منگانی، لافوت، بالائی اور بالائی کی دلدل میں ڈھکل کر ہر	n	20	3040	6091	20900
ہ اور مسلمان لگ ہ کو بہت سے مبارکات جن سے مبارکات اور ریسٹ اور کارکنوں سے جنت کی۔ کام کر	p	16	997	2615	10800
الحمد للہ رب العالمین شہو جاہلوں کی سادار فتح مبارک ہوا صرف نام ہی کافی ہے	p	16	1775	3890	18400
اللہ کا شکر شہروں کو مبارک	p	5	52	875	3184
اک واری ہے۔ شہرورورور الحقائق	p	5	2743	4325	19900
بجائ اور ہمارا اور آئی والی سناؤں کا مستقبل ہے؟ کیا ہم ملک ایک محفوظ ملک تصور ہو	n	3	2504	4798	17400
اور یہی واقعات وہ آئندہ ہوتے ہیں جس سے ہا خلیفہ تیار آتار کر سکتے ہیں، نا ہی جھوٹا	n	27-Nov-21	2002	4102	9798
وہ دلت جو چند روز کے افکار کے لیے عمران خان سے عیوب کا نشانہ، عمران	n	25	1179	2880	6160
، تکلف اور ہے سے ہستی کی داستان ہے میری دعا ہے انہی فظوں میں عمران خان،	n	25	1238	3258	8721
کی آئینا ہے، خلق خدا کا رہی ہے، تڑپ رہی ہے ہر جہاں ہے ظالم حکومت کو کوئی فر	n	25	2633	4239	14500
بھروسے میں کی کھائی ہے	n	24	550	1382	4838
اللہ اکبر	p	21	2189	4065	19000
مادی انعام اور ظلم سدہ کر ایک نسر پاکستان کی بنیاد رکھی۔ انتہاء اس جھٹیلے	p	16	2586	4295	10500
ا شکر ہے جو مشکل وقت میں ساتھ بھائی رہے، ہوا صرف کیے ساتھ اب سب کو بھی آ	p	15	1041	2049	7284
ی اللہ ہے کہ کت یا صرف مسرورہ کا کتا، انکی خاندان میں جلیے، اور وہ کو دینا اور یا	p	15	1901	4907	10900
ہے بڑا جائے جو ضرر کرے اور آقا قبیلہ اللہ ہے جو وہ ہے اس میں فاطمہ کی لیے بہت	p	15	4098	5845	18500
اجلاس، صد، حاد سمجھ کر ہوت ہے، نہ کتا ہوا ہے نہ جتے سکتی، سر کر حاد اجلاس ملتا	n	10	1392	3978	12100

Figure 4. Urdu Tweets Dataset

1.2.4. Word Segmentation. One of the most challenging difficulties to solve in Urdu (or any Perso-Arabic script language, such as Sindhi) or English words. It commonly

happens when a user searches the Urdu corpus for English terms. The reader may read the word/words without difficulty, but the total corpus/words are treated as a single token for the computing task [16].

1.2.5. Duplicate Data. While collecting the tweets, some of them were duplicated. Later, while training the desired dataset, these duplicated data were removed.

1.2.6. Has-Tags. As discussed previously, a hashtag is a term or phrase that begins with the # symbol and is used to classify and associate any text/tweet with a trend/topic. Similarly, has-tags can produce a redundant set of tweets if these are added in replies.

1.2.7. Data Preparation and Cleaning. Following the completion of the tweets collection activity, the procedure of data cleaning begins. While collecting tweets, we manually cleaned the data formally. Still, there were some regular expressions and URLs left. The issues are resolved in two steps. Thus, for a tweet, the first step involves the removal of URL anchors, Twitter handles, and has-tags therein, through the utilization of Regular Expressions, such that doing so will transform into the modified tweet [17]. Let $\Theta(\beta, \Gamma)$ be the function that takes a *Regex* pattern B and text and gives you the $\Delta\tau$ in return. The next step, $\Delta\tau$ is inducted into a hash-set (H) to gather the unique set of tweets. Algorithm1 defines the whole process applied to the entire collection of tweets T , such that $T = T_1, T_2 \dots T_n$.

Algorithm 1: Data Cleaning and Redundancy Removal

Result: Set of distinct tweets H

$T \leftarrow$ be the set of tweets

$H \leftarrow$ be an empty hash - set

$\mu \leftarrow$ be the *Regex* pattern for URL's

$\alpha \leftarrow$ be the *Regex* pattern for Twitter handles

$\lambda \leftarrow$ be the *Regex* pattern for hash - tags

for every τ in T **do**

 | $\Delta\tau \leftarrow \Theta(\lambda, \Theta(\alpha, \Theta(\mu, \tau)))$;

 | $H \leftarrow H \cup \{\Delta\tau\}$;

end

We have employed the has-set because, under the hash function, the redundant tweets collide to have the final result in the form of a set of separate tweets. Dictionaries or Hash-maps can be used as an alternative to hash-sets.

Lastly, we have tokenized the dataset and extracted the distinct words in the corpus. Then we manually performed the segmentation on the tokens, where space was not inserted after non-joiners.

1.2.8. Data Labelling. We created the 'Urdu Sentiment Corpus (USC)' dataset and insights from Urdu tweets for sentiment analysis and polarity recognition as part of this research. The dataset is composed of tweets that cast a political shadow and present a competitive climate between political parties and the Pakistani government. Overall, the

dataset contains around 1300+ tokens, with 702 entries being positive and 613 being negative. In addition, this research describes visual insights into literary similarities, manifold learning, and other topics.

Each tweet in the dataset was manually labelled with two sentiments, i.e., Positive (P) and Negative (N). The label was considered positive, where the corpus shows the good aspects of a situation based on thought, feelings, and judgment. On the other hand, the negative sentiment shows the facts, concerns, or experiences based on unpleasant moments, depressing and emotional idealism.

1.3. Sentiment Analysis on Individual Behaviour

They are starting from the assumption that the factors orienting political choices are always heterogeneous and multidimensional. Since Pakistan's inception, the improper functioning of political parties, ineffective leadership, and dismemberment of units have been significant contributors to a paralyzed political system, which has resulted in a slew of problems such as poverty, unemployment, crime, low women's status, child marriage, rape, and gender inequality. These are Pakistan's social difficulties, which, if addressed, may make a living more comfortable and society more productive.

This research study explores an individual's political tweets, allowing us to understand their sentiment regarding a specific theme. For that, we took two famous and well-known Pakistani Politian. Politician 1 and Politician 2. In this research, we haven't mentioned their names—why we are not targeting a specific political party or person. As we have already said, this research is entirely based on manually extracted data from their official Twitter accounts. We explored the political factors influencing politician attitude and behaviour through these tweets. According to their tweets, the latest development in Pakistani politics changes their behavior ranges from the first phase. We have collected little tweets. But it shows the limits of a superficial inquiry, as established by looking at a lot of data. So, for future research, it might be useful to use a different technique to realise a comparison and to see if the size of the sample under consideration and the depth of the analytic technique can alter the results and the subsequent consideration. Based on political advancement and leading aspects, Figure 5 shows the tweets of both politicians' actions.

2. Machine Learning Classification Algorithms

A dataset will often have significant particulars that may be utilized for generating decisions quickly. No system can make intelligent decisions in the absence of the classification of such datasets. Therefore, classification algorithms make the process easier by removing unnecessary steps, identifying important data categories, and generating valuable models. Each of the relevant papers can be classified into one of three different groups: supervised, unsupervised, or semi-supervised. Texts may be classified using a variety of different approaches, including SVM, KNN, Logistic

Politician 1		Date	Politician 2		Date
Tweets	Sentiment		Tweets	Sentiment	
<p>رات کی آغوش میں سکر دو حکمگانا ہے اور ہماری بھابی کلکتہ بلتستان میں برف پر باکی کھولتی ہیں۔ سکر دو میں بنی الاقوامی ہوائی اڈے کی تعمیر کے ساتھ ہم علاقہ اب سرما کی کھولوں کا ایک عالمی مرکز بنے گا۔ انشاءاللہ</p>	p	29-Jan-2022	<p>ایک شیر، تین شکار مہنگائی، نااہلی، اور لوٹ مار خانبوال کا الیکشن جنوبی پنجاب میں پاکستان مسلم لیگ (ن) کی عوامی خدمت کا اعتراف ہے۔ میں پارٹی قائدین اور کارکنان کا مسلمان روزِ محبت پر شکرگزار ادا کرتا ہوں اور شہادت، دنیا ہے۔</p>	n	16-Dec-2021
<p>بکجینی کے اطباء+امریکی حمایت سے مقامی صحافیوں کے بل پر حکومت بدلیے+صاحبوں پر پرا ڈاکوٹ کے لیے حمایت ڈولے کو اقتدار دلوانے کے خلاف جدیبات سے بیرونی اور احتجاج کیلئے بڑی تعداد میں نکلے برہمن اہل پاکستان کامنٹریکس اور بیرونی ملک مقیم پاکستانیوں کے لیے سہولتیں</p>	n	11-April-2022	<p>مہینت کو درست سمت اسٹوار کرتے مہنگائی کے سناٹے عوام کی دادرسی کرنی ہے، روگار کی فراہمی سے عربوں کے چولے پھر سے چلائے ہیں۔ عالمی سطح پر پاکستان کے مقام اور تعلقات کو قومی وقار اور مفادات کی بنیاد پر بحال کرنا ہے۔ قیادت کا امتحان ہی مشکل ترین حالات میں ہوتا ہے۔</p>	p	8-April-2022
<p>سوالات اٹھائے۔ مہینت ہلیدی نہادی کے کڑھے میں اتر چکی ہے۔ ڈاکوٹ اور ان کے آقاؤں کو میں خیردار کرنا چاہتا ہوں کہ ہم غیر جمہوری اور سماجی اقدامات معاشی صورتحال میں مزید تیزی کے موجب نہیں ہے اور ملک کو طوائف الملوک کی جانب دھکیلی ہے۔</p>	n	24-May-2022	<p>آج پورے ایک خاص دن سے بڑھ کر ایک خاص موقع ہے، بے مثال اتحاد و یکجہتی کا۔ آج کلہ کرنے کا نہیں کیے ملنے کا دن ہے، فور کے بگ فالٹ ہونے کا دن ہے اور مل کر مسکرائے کا دن ہے۔ تمام پاکستانیوں اور عالم اسلام میں آباد ہر مسلمان کو میری طرف سے عید کی دلی مبارکباد</p>	p	03-May-2022

Figure 5. Political Behaviour on Individual Personality

Regression and Decision Trees, among others. Word2vec was utilized for data evaluation in the present research; specifically, for the Support Vector Machine (SVM), Decision Trees, XGboost, Random Forest, and neural network.

2.1. SVM Classifier

A supervised machine learning approach for text categorization that was presented by Salton et al. [18] provides a framework for the SVM. For text classification, SVM has been selected by several researchers. For instance, The SVM has several benefits, including excellent accuracy and a lower risk of overfitting. SVM are great for text classification jobs due to their speed and ability to come up with solutions on the fly. It is also categorised as a technique for categorising data as linear or non-linear, to put it another way. The SVM method employs a non-linear mapping methodology to transfer linear training data into a higher dimension and look for linear optimum separation hyperplanes [19].

2.2. Logistic Regression

Logistic regression, often known as Linear Regression, is a classification issue solution. Its predictive analysis is based on probability. The sigmoid function, which is one of the more sophisticated algorithms, is the most commonly employed in Logistic Regression [20]. According to this study, Linear Regression fared well on text categorization, ranking first among all classification algorithms with an accuracy of 85%.

2.3. Decision Tree Classifier

The decision tree algorithm is a supervised learning technique used in machine learning. The If-then rule structure on which it is based makes it easy to use. The ability to comprehend results and deal with relationships between features are two of the decision tree’s strongest suits. Not only does it function well with numerical data, but also with text and other types of data. Avoiding overfitting in decision trees is possible through a process called “tree pruning” [21].

2.4. XGboost Classification

The supervised-learning algorithm XGBoost (Extreme Gradient Boosting) is widely used. On a predefined Urdu dataset, we have utilized it for regression and classification. It uses a training method that is extremely scalable and avoids overfitting by building short decision trees consecutively [22].

Wrapper classes are provided by XGBoost so that models may be used in the Scikit-learn environment in the same way as classifiers and regressors are. Therefore, the entire Scikit-learn package may be used with XGBoost models.

2.5. Random Forest Classifier

It can classify data and run regressions. A decision tree is built on the training side. From each tree, it predicts the means of regression and classification and assigns classes. The idea behind Random Forest is that features are selected at random during induction, and that the classification it generates is based on a random selection of data points/samples from the training data. As far as text classification algorithms go, this study is the third. In order to create different decision trees, it uses random subspace and tree bagging methods, as well as randomly selected data samples from the training data [19]. Here, separate trees are built using random samples, and the classification decision is anticipated to be made by each tree in the random forest. The tagged data is permitted to traverse the trees once the entire forest has been built. Now we get to the proximities; two occurrences are considered closer together by one if they both occur on the same leaf node. The total number of trees in the forest is used to adjust the proximities [23].

3. Word2vec and TF-IDF

The algorithm has difficulty processing raw text and can only make sense of numerical data. Therefore, as a first step, we must transform the plain text into numerical form. NLP can help change the raw text into vectors [24].

The feature generated by word embedding features is dense and low-dimensional, while the part caused by TF-IDF is sparse and high-dimensional. So, it’s also an outstanding example of semantic meaning.

Word2vec embedding, developed by Google in 2013, is one of the significant advances. Word2vec’s embedding

superiority over TF-IDF is the determining factor in many situations [25][26]. In 2017, the Transformer network was introduced, and after extensive study, the Research, Bidirectional Encoder Representation from Transformer, was published (BERT).

3.1. Word2vec

As a single-hidden-layer neural network model, word2vec is essential. Each word in the sentences or corpus has its neighbouring words predicted. The model's hidden layer's learning weights can act as word embeddings. Thus we'll need to get them. Word2vec, in its simplest form, transforms a word or phrase into a vector in D-dimensional space [27].

The purpose of pre-processing data is to organize it into a collection of word arrays, which enables additional pre-processing. Case folding, tokenization, stemming, stop-word removal, and padding are all pre-processing components. As a processing advantage, case folding converts all words to lowercase, making them uniform and suitable to be represented in the Word2Vec method in the same way they were before. The next step is tokenization [28], in which the text is split into smaller, more manageable chunks called tokens. Tokens might be symbols, phrases, or other discrete items with a clear meaning. The purpose of eliminating filler words is to narrow attention to only the most relevant terms in preparation for further processing. Lemmatization is a method for standardizing text that transforms each word into its stem form. The lemmatizer function from the UrduHack library is used in the research. During the preparation phase, padding is the last step. This is a crucial stage since it ensures that all LST training input papers are of the same length. Finally, all the documents in the data collection are under the minimum required word count. This occurs when the document's size exceeds the maximum allowed length when the token "*<pad>*" is inserted [29].

3.2. Training of Word2vec

The author [30] recommend using Word2Vec. By using a neural network architecture, the Word2Vec training process allows the system to grab vector representations of words. The pre-processed Urdu political tweet data is used as the input for training the Word2Vec algorithm. Word2Vec takes as input a set of pre-processed Urdu political tweets and produces vector representations of each word. The creation of a vocabulary from input data is the first step in the process of developing a Word2Vec model.

After that, we learn a vector representation of the word. Both negative and positive sampling are utilised to evaluate the performance of the Word2Vec model embedded in this study. Figure6 provides a visual breakdown of the various Word2Vec layouts. The original proposal for the hierarchical sigmoid function can be found in Morin et. al. [31], which proposes a binary tree structure of the vocabulary set with a word count (W), with each word represented by a leaf node and each node having a determined probability for its child nodes

[32]. The probabilities of the words are assigned using a random walk. As in Generative Adversarial Networks [33], presented negative sampling, which is predicated on the estimate of noise contrast; that is, a decent model should be able to distinguish between the genuine signal and the fake signal using logistic regression. The dimensions of word2vec are chosen to be in the range of 100 to 300 since this is the range most frequently used in existing research. The features of high-dimensional spaces are also lost if we choose dimensions smaller than 100.

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 52, 128)	1058560
conv1d (Conv1D)	(None, 48, 512)	328192
max_pooling1d (MaxPooling1D)	(None, 9, 512)	0
conv1d_1 (Conv1D)	(None, 5, 256)	655616
max_pooling1d_1 (MaxPooling1D)	(None, 1, 256)	0
flatten (Flatten)	(None, 256)	0
dense (Dense)	(None, 32)	8224
dropout (Dropout)	(None, 32)	0
dense_1 (Dense)	(None, 1)	33

```

Total params: 2,050,625
Trainable params: 992,065
Non-trainable params: 1,058,560

```

Figure 6. Word2vec Model Layers

3.3. Word2vec using LSTM

The LSTM is trained to develop a model for the sentiment classification model. On the other hand, testing aims to evaluate how well the classification model functioned after training [32].

By "word embedding," we mean the technique of modeling a document's vocabulary with a set of vectors. Word embeddings may be learned more easily with the help of the embedding layer. Glove and Word2Vec are two methods used to study word embeddings. The Word2Vec word embedding method [34].

The LSTM technique is a subset of the RNN series designed for processing models containing long input sequences. The RNN method builds on the foundation of regular feed-forward neural networks. But RNN models have problems with explosions and gradient vanishing. The LSTM model was developed to overcome these challenges. This capacity arises from the LSTM model cell memory, which helps maintain the system current state [35].

In addition to the cell memory, the LSTM model has three gates: the input gate, forget gate, and the output gate. The gate input specifies which bits of information are modified and delivered into the memory cell. The forget gate's function is to determine whether or not the data being input and output is safe to proceed on. Forgetting happens if

the forget gate's output is near zero; retaining occurs if it's close to one. As a result of this forget gate operation, LSTM can deal with the exploding problem and the vanishing gradient problem. The state of the cell is unaffected by the gate output; however, the date differentiates between the cell state and the valid information.

```

Model: "sequential_1"
-----
Layer (type)                Output Shape                Param #
-----
embedding_1 (Embedding)     (None, 52, 128)            1058560
bidirectional (Bidirectiona (None, 512)                788480
l)
dense_2 (Dense)             (None, 128)                65664
dropout_1 (Dropout)        (None, 128)                0
dense_3 (Dense)             (None, 64)                 8256
dropout_2 (Dropout)        (None, 64)                0
dense_4 (Dense)             (None, 1)                  65
-----
Total params: 1,921,025
Trainable params: 862,465
Non-trainable params: 1,058,560

```

Figure 7. Word2vec Model using LSTM

3.4. Pooling layer

The pooling layers compress the spatial input, which in turn decreases the number of network parameters and accelerates computation while also controlling overfitting. It is common practice for the pooling layer to use both maximum and average pooling. Their names suggest that maximum and average values are used in the respective methods.

3.5. Fully Connected Layer

Given that the fully connected layer requires a vector as an input, a transformation from the pooling layer's multidimensional array output is needed.

4. Evaluation Metrics

An assessment metric quantifies a predictive model's effectiveness. This process typically entails training a model on a data set, using the model to make predictions on a holdout data set that was not utilised during training, and then comparing the predicted values to the expected values in the holdout dataset [36].

In this research, we have different evaluation metrics as mentioned below;

4.1. Precision

Precision is a measure of performance that is used in pattern recognition, information retrieval, and classification

(machine learning) to evaluate information that has been obtained from a dataset. In our case, we have used Urdu Political tweets. The purpose of the performance is to determine how many positive observations were predicted accurately as a proportion of all positive occurrences [37].

4.2. Recall

The recall is the ratio of correctly predicted positive observations to all observations in the actual class [38].

4.3. F1-Score

Precision and Recall are weighted to produce the F1 Score. True positives and false negatives are both factored into this score. F1 is typically more useful than accuracy, especially if your class distribution is not uniform, despite the fact that it is not as intuitively appealing. When the costs of false positives and false negatives are roughly equal, accuracy works best. It is more prudent to examine both Precision and Recall if the cost of false positives and false negatives is substantial [39].

4.4. Accuracy

The proportion of the total set of data that was accurately predicted is how prediction accuracy is measured. More precisely, it is the ratio of the total number of positive and negative results divided by the total number of positive and negative results, which includes both true and false positives and negatives [40].

5. Results and Discussion

Several machine learning classifiers, selected for training text data classification, have been covered in this section. In this research, SVC classifiers are the most widely used supervised learning classifiers [19], Decision Tree [21], XG-Boost [22], and Random Forest. For the neural network, we have used Word2vec. Table1 shows the Accuracy and other evaluation metrics results based on Precision, Recall, and F1-score, obtained through Urdu text classification from the corpus (Tweets).

It is worthwhile to mention that the Logistic regression classifier attained the highest accuracy (overall) among the rest of the tested algorithms. The neural network i.e. word2vec has the lowest accuracy of all classifiers. Most of the models predicted well during the test on Urdu Tweets text data to analyze the efficiency and accuracy of each classification model.

Sentiment analysis on Twitter can be viewed as a classification problem in which positive and negative tweets must be sorted into separate categories. As a result, we have used a variety of classifiers for the task, and we have checked their performance by measuring their Accuracy of prediction. Precision, Recall, and F1 Score are also calculated to measure performance. Parameters are derived from the confusion matrix.

TABLE 1. URDU TEXT CLASSIFICATION RESULT

Models	Sentiment	Precision	Recall	F1-Score	Support	Accuracy
SVM Classifier on TF-IDF Vectorizer	Negative	0.85	0.80	0.82	186	0.84
	Positive	0.83	0.87	0.85	209	
Logistic Regression Classifier on TF-IDF Vectorizer	Negative	0.85	0.82	0.83	182	0.85
	Positive	0.85	0.87	0.86	213	
Decision Tree Classifier on TF-IDF Vectorizer	Negative	0.58	0.82	0.68	125	0.75
	Positive	0.89	0.73	0.80	270	
XGboost Classifier on TF-IDF Vectorizer	Negative	0.52	0.82	0.63	111	0.73
	Positive	0.91	0.70	0.79	284	
Random Forest Classifier on TF-IDF Vectorizer	Negative	0.67	0.80	0.73	147	0.78
	Positive	0.87	0.77	0.81	248	
word2vec	Negative	0.61	0.56	0.58	140	0.58
	Positive	0.54	0.60	0.57	123	
word2vec using LSTM Layer	Negative	0.12	0.56	0.19	27	0.52
	Positive	0.91	0.53	0.67	236	

TABLE 2. CONFUSION MATRIX

Predicted Class	Positive Prediction	Negative Prediction
True Positive	TP (True Positive)	FN (False Negative)
True Negative	FP (False Positive)	TN (True Negative)

Although there is no scarcity of classifiers (or classification models), we have focused on the most suitable for text analysis. Accuracy detection is a key tool for evaluating the classifier's performance. For this purpose, accuracy alone is not enough for evaluating classifier performance; other metrics such as precision, recall, and F1-score measure are required. Table 2 shows an example of a confusion matrix, which compares the true results with those predicted by a classifier.

The F1-score metric represents a harmonic mean of the Precision and Recall measures. Determining a decision based on Precision and Recall alone can be challenging, especially when one of the scores becomes extremely high or extremely low. Then we can decide upon the validity of the result by considering the F1-score measure.

Looking at the Table 1, Word2vec shows worse result compared to other classifier models. The reason behind is the Word2Vec always struggles with words that are not in its dictionary. Out-of-vocabulary (OOV) words are given a potentially unsatisfactory random vector representation. Local lexical knowledge is used for this purpose. Words' semantic representations are based entirely on their surrounding contexts, which isn't always ideal. It is not possible to share parameters used in the training of new languages. We have to start from scratch if we want to train word2vec in a new language like Urdu. It is incapable of dealing with ambiguity. If a word has multiple meanings, and there are many of these terms in the real world, embedding will reflect the average of these senses in vector space.

More data is needed for the network to converge (particularly when employing skip-gram). The dataset size of 1300 tweets is not large enough to train a word2vec neural network

6. Limitation

In the field of NLP, sentiment analysis is among the most successful and widespread uses. There are still many issues with it, despite all the hype it has garnered since it was founded.

One of the most difficult aspects of creating datasets is ignoring potentially biased perspectives of individuals due to the cultural backgrounds of those individuals or their own personal opinions. In addition to this, sentiment analysis of tweets based on biased views can choose just those results and data that support their major arguments, which, needless to say, may have an effect on our research. We have paid close attention to the explanation of the primary problem we tried to solve, and we make sure that we have collected accurate data.

Systems for sentiment analysis trained on data from political tweets often perform far worse when applied to data from other domains. Politicians do not always communicate sentiment in the same way that a critic does on social media, which is different from how a social media poster does. Therefore, it is not uncommon for machine learning systems trained on political tweet data to fail miserably when faced with the task of predicting sentiment in a more general domain.

Contextual understanding is another facet of NLP. This area is now experiencing a surge in research activity. Memory networks are a type of model that may derive answers to queries from context. Similar to how a person's brain processes separate words, attention networks can process different sections of a phrase independently. The larger context of a statement can be understood with the help of RNN, like bidirectional LSTM. Our research has led to the beginning of incorporating these into sentiment analysis.

By adjusting the parameter values based on their learning rate, epochs, batch size, etc., we have previously tried numerous fine-tuning strategies, as seen by the results. The results were still inappropriate. We can try to train more models for this specific domain, ideally using a more sophisticated technique (like deep neural networks). In terms of both time and money, this is the worst possible option. Yet, if we are able to accomplish this with our available

resources, we may end up with a valuable and exclusive piece of intellectual property.

7. Conclusion and Future Work

This study employed a variety of machine learning classification algorithms—including SVM, Logistic Regression, Decision Tree, XGboost, and Random Forest—using the TF-IDF measure—to classify Urdu text (Tweets). In order to extract the idea of relatedness across words, the neural network makes use of Word2vec.

We used data cleaning, feature extraction, and feature selection as NLP pre-processing approaches before we used Urdu-Text. Nevertheless, when compared to other classifiers tested on a comparable dataset for Urdu tweets, the results demonstrated that Logistic Regression and SVM achieved 85% and 84% accuracy, respectively.

The results demonstrate that the TF-IDF model outperforms the Word2Vec model because to the imbalanced data points in each emotion class and the large number of classes with an insufficient number of data points. Contrasted with the abundance of other emotions, the number of surprise feelings constitutes a small fraction of the total data set. The results of research may be impacted by data limitations. We will require a sizable dataset in the future that can handle these constraints.

References

- [1] G. L. Y. Londo, D. H. Kartawijaya, H. T. Ivaryani, Y. S. P. WP, A. P. M. Rafi, and D. Ariyandi, "A study of text classification for indonesian news article," in *2019 International Conference of Artificial Intelligence and Information Technology (ICAIT)*. IEEE, 2019, pp. 205–208.
- [2] Y. Zheng, "An exploration on text classification with classical machine learning algorithm," in *2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, 2019, pp. 81–85.
- [3] S. K. Dwivedi and C. Arya, "Automatic text classification in information retrieval: A survey," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 2016, pp. 1–6.
- [4] V. Korde and C. N. Mahender, "Text classification and classifiers: A survey," *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 2, p. 85, 2012.
- [5] M. Ringsquandl and D. Petkovic, "Analyzing political sentiment on twitter," in *2013 AAAI Spring Symposium Series*, 2013.
- [6] A. Tumasjan, T. Sprenger, P. Sandner, and I. Welpel, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 4, no. 1, 2010, pp. 178–185.
- [7] J. W. Pennebaker, C. K. Chung, M. Ireland, A. Gonzales, and R. J. Booth, "The development and psychometric properties of liwc2007: Liwc. net," *Google Scholar*, 2007.
- [8] M. Choy, M. L. Cheong, M. N. Laik, and K. P. Shung, "A sentiment analysis of singapore presidential election 2011 using twitter data with census correction," *arXiv preprint arXiv:1108.5520*, 2011.
- [9] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle," in *Proceedings of the ACL 2012 system demonstrations*, 2012, pp. 115–120.
- [10] T. Elghazaly, A. Mahmoud, and H. A. Hefny, "Political sentiment analysis using twitter data," in *Proceedings of the International Conference on Internet of things and Cloud Computing*, 2016, pp. 1–5.
- [11] P. Sharma and T.-S. Moh, "Prediction of indian election using sentiment analysis on hindi twitter," in *2016 IEEE international conference on big data (big data)*. IEEE, 2016, pp. 1966–1971.
- [12] A. Daud, W. Khan, and D. Che, "Urdu language processing: a survey," *Artificial Intelligence Review*, vol. 47, no. 3, pp. 279–311, 2017.
- [13] P. Rose and M. Greeley, "Education in fragile states: Capturing lessons and identifying good practice," *DAC Fragile States Group*, 2006.
- [14] S. W. Kazmi and H. Quran, "Role of education in globalization: A case for pakistan," *SAARC journal of human resource development*, vol. 1, no. 1, pp. 90–107, 2005.
- [15] R. Lu and Q. Yang, "Trend analysis of news topics on twitter," *International Journal of Machine Learning and Computing*, vol. 2, no. 3, p. 327, 2012.
- [16] J. R. Saffran, E. L. Newport, and R. N. Aslin, "Word segmentation: The role of distributional cues," *Journal of memory and language*, vol. 35, no. 4, pp. 606–621, 1996.
- [17] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," *Applied artificial intelligence*, vol. 17, no. 5-6, pp. 375–381, 2003.
- [18] G. Salton, R. Ross, and J. Kelleher, "Idiom token classification using sentential distributed semantics," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 194–204.
- [19] Y. Al Amrani, M. Lazaar, and K. E. El Kadiri, "Random forest and support vector machine based hybrid approach to sentiment analysis," *Procedia Computer Science*, vol. 127, pp. 511–520, 2018.
- [20] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.
- [21] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.
- [22] P. Elena, "Predicting the movement direction of omxs30 stock index using xgboost and sentiment analysis," 2021.
- [23] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random

- forests and decision trees,” *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 5, p. 272, 2012.
- [24] W. Zhu, W. Zhang, G.-Z. Li, C. He, and L. Zhang, “A study of damp-heat syndrome classification using word2vec and tf-idf,” in *2016 IEEE international conference on bioinformatics and biomedicine (BIBM)*. IEEE, 2016, pp. 1415–1420.
- [25] B. O. Deho, A. W. Agangiba, L. F. Aryeh, and A. J. Ansah, “Sentiment analysis with word embedding,” in *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*. IEEE, 2018, pp. 1–4.
- [26] C. Esteves, C. Allen-Blanchette, X. Zhou, and K. Daniilidis, “Polar transformer networks,” *arXiv preprint arXiv:1709.01889*, 2017.
- [27] K. W. Church, “Word2vec,” *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.
- [28] G. Grefenstette, “Tokenization,” in *Syntactic Wordclass Tagging*. Springer, 1999, pp. 117–133.
- [29] X. Rong, “word2vec parameter learning explained,” *arXiv preprint arXiv:1411.2738*, 2014.
- [30] Y. Goldberg and O. Levy, “word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method,” *arXiv preprint arXiv:1402.3722*, 2014.
- [31] F. Morin and Y. Bengio, “Hierarchical probabilistic neural network language model,” in *International workshop on artificial intelligence and statistics*. PMLR, 2005, pp. 246–252.
- [32] C. Steinruecken, “Compressing sets and multisets of sequences,” *IEEE Transactions on Information Theory*, vol. 61, no. 3, pp. 1485–1490, 2015.
- [33] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.
- [34] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, “Sentiment analysis using word2vec and long short-term memory (lstm) for indonesian hotel reviews,” *Procedia Computer Science*, vol. 179, pp. 728–735, 2021.
- [35] R. Huang, C. Wei, B. Wang, J. Yang, X. Xu, S. Wu, and S. Huang, “Well performance prediction based on long short-term memory (lstm) neural network,” *Journal of Petroleum Science and Engineering*, vol. 208, p. 109686, 2022.
- [36] M. Hossin and M. N. Sulaiman, “A review on evaluation metrics for data classification evaluations,” *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.
- [37] M. Kane, “The precision of measurements,” *Applied measurement in education*, vol. 9, no. 4, pp. 355–379, 1996.
- [38] M. Buckland and F. Gey, “The relationship between recall and precision,” *Journal of the American society for information science*, vol. 45, no. 1, pp. 12–19, 1994.
- [39] O.-M. Sulea, M. Zampieri, S. Malmasi, M. Vela, L. P. Dinu, and J. Van Genabith, “Exploring the use of text classification in the legal domain,” *arXiv preprint arXiv:1710.09306*, 2017.
- [40] F. Freese *et al.*, “Testing accuracy,” *Forest Science*, vol. 6, no. 2, pp. 139–45, 1960.

Stress Detection and Prediction Using CNNs from Electrocardiogram Signals

Jehad Ur Rahman*

Department of Electrical Engineering
UET Peshawar, Pakistan

jehadurrahman@uetpeshawar.edu.pk

Samad Riaz

Department of Electrical Engineering
UET Peshawar, Pakistan

samadriaz@uetpeshawar.edu.pk

Salma

Institute of Nursing Science
KMU Peshawar, Pakistan

salmafaiz09@gmail.com

Abstract— Stress prediction is a crucial aspect of mental health monitoring, with consequences for both psychological well-being and productivity. This work presents a unique way for stress prediction that uses binary and multiclass classification models. Through extensive experimentations with different durations and frequencies of Electrocardiogram Signal (ECG) signals, we identified a 5-second dataset sampled at 200Hz as the optimal configuration for our model. Moreover, we introduced an innovative feature i.e., the prediction of stress scores ranging from 0 to 100, providing nuanced insights into stress levels, where 0 represents no stress and 100 indicates high stress levels. The model obtains 95.04% accuracy, 95.27% precision, 94.95% F1 score, 86.69% sensitivity, and 99.44% specificity for the binary classification. With "Fun" added to the list of stress categories in addition to "Base" and "TSST," the model continues to perform well in the multiclass classification scenario, with accuracy of 88.10%, precision of 87.60%, F1 score of 87.35%, sensitivity of 95.97%, and specificity of 79.23%. These findings highlight how well this applied strategy predicts stress levels, providing important information for mental health and stress management strategies.

Keywords: Stress Detection, Stress Score, ECG Signals, Stress Levels, CNNs.

I. INTRODUCTION

People become increasingly stressed as societies expand because of the increased competition. This stress can have negative consequences for work, relationships, and safety. Our rapid and demanding society has made mental stress a widespread problem that has an impact on people's productivity and general well-being. Uncontrolled chronic stress can cause a variety of health issues, such as cardiovascular disease, depression, and anxiety [1]. Long-term stress can lead to depression, addiction, and heart and brain disorders [2]. Emotional stress is currently a major issue for both physical and mental health. Thus, creating efficient techniques for stress evaluation and management is essential to preserving public health. Stress is a physiological reaction to challenging events. It is distinguished by a sequence of physical and emotional changes, such as elevated heart rate, muscle tension, and anxiety. While acute stress can help mobilize resources to deal with urgent dangers, chronic stress, when extended, can be harmful to both physical and mental health. The neurological system in our bodies responds differently when we are under stress. Stress stimulates the sympathetic nervous system (SNS), which regulates heart rate and breathing. After stress, the parasympathetic nervous system (PNS) takes over to calm things down. We can detect stress by observing changes in parameters such as heart rate. Researchers have been looking

into how electrocardiogram (ECG) signals, which assess heart activity, can help detect stress. Traditionally, they used five minutes of ECG data, which is too long for real-time monitoring [3]. Some studies have successfully detected stress using only one minute of ECG data, but this is still not ideal because it requires wearing uncomfortable equipment and is too slow for real-time monitoring.

Healthcare is one of the many industries that artificial intelligence (AI) has changed. AI has shown great promise in the field of stress assessment as a means of detecting stress patterns and forecasting stress levels. Complex patterns can be extracted and analyzed from a variety of data sources, such as physiological signals, behavioral data, and self-reported assessments, by AI models, especially deep learning algorithms. Because ECG signals are constant, freely accessible, and non-invasive, they have become more important in AI-based stress evaluation. ECG signals are the electrical activity of the heart [4]. Rich information on physiological changes linked to stress can be found in ECG signals, including heart rate variability (HRV), heart rate (HR), and signal complexity. These minute variations in ECG signals can be examined by AI models to precisely identify and categorize stress levels. Traditional approaches for assessing stress from ECG signals typically rely on hand-crafted time or frequency domain features [5]. These approaches, however, may be limited in their ability to capture complex patterns and correlations within ECG signals

We've developed a new method for detecting mental stress and predicting stress scores based on a Convolutional Neural Networks (CNN) architecture. Our method entails obtaining ECG signals, cleaning them using a bandpass filter to reduce noise, and altering the frequency from 700Hz to 100Hz. These preprocessed signals are then sent into a CNN, which extracts unique stress patterns from the temporal data of ECG signals. We trained two models: one for binary classification, which distinguishes between stressed and non-stressed individuals, and another for multiclass stress prediction. The binary classification technique has produced groundbreaking results in predicting stress levels. We use the obtained information to reliably diagnose stress levels and predict stress scores.

II. LITERATURE REVIEW

In this article [6], the author investigated the analysis of ECG Raw Signal and Spectrogram pictures, using a dual method combining Raw ECG with 1D CNN and Spectrograms with ResNet-18 architecture. Their analysis produced complex results, with an accuracy of 66.6%, precision of 67.6%, and

recall of 66.6% across three unique categories: neutral, tension, and amusement. This extensive study combined Leave-One-Subject-Out (LOSO) methods with chest-worn ECG data. Furthermore, the study expanded its investigation to the RML dataset, where deep learning models showed notable performance measures, including an accuracy of 72.7%, precision of 76.6%, and recall of 72.7%. Notably, this study used datasets from LESO, RML, and WESAD, allowing for both binary and three-class classification. In [7] emphasis is on the use of raw ECG signal data, which was analyzed using CNN and Bidirectional Long Short-Term Memory (BiLSTM) architecture. The results of this investigation were positive, with an overall accuracy of 86.5% and a specificity of 92.8%. Furthermore, the study carefully classified stress levels into three categories: low (91.3%), moderate (89.4%), and high (79.8%). These conclusions are based on locally acquired data, demonstrating the study's relevance and applicability. Article [8] analyzed ECG and HRV data using CNN for categorization purposes. Their investigation produced remarkable performance measures, including 97% accuracy, precision, recall, and F1-Score. Notably, the study got its data locally, which ensured the dataset's validity and dependability. Furthermore, the categorization assignment had three unique classes, which provided insights into subtle changes in the dataset. In the [9] raw ECG data is used, which was classified using CNN and VGG-inspired architectures. The study produced strong results, with claimed accuracies of 83.55% for three classes and 93.77% for two classes. Notable is the use of the Drive DB and Arachnophobia datasets, using a VGG-inspired architecture for binary classification and a 1D CNN for categorization into three classes. This strategic approach demonstrated the flexibility and versatility of the approaches used across a variety of datasets. The [10] performed a detailed investigation of ECG and HRV features using K-Nearest Neighbors (KNN) and Probabilistic Neural Network (PNN) classifiers. The study found impressive accuracies of 91.66% (ECG) and 94.66% (HRV), along with thorough specificity and sensitivity data for both modalities. The use of locally obtained data is significant since it increases the study's relevance and application to real-world circumstances. Furthermore, the study's emphasis on binary categorization highlighted its practical applications in the healthcare domain. The study described in [11] included the integration of ECG and EEG data using a Radial Basis Function Support Vector Machine (RBF-SVM) and KNN classifiers. The results showed significant accuracies ranging from 86.13% to 87.75% across various stress characteristics, as defined in the Kaggle dataset. This extensive research enabled binary categorization scenarios, revealing light on different stress levels and their physiological manifestations. The study's thorough approach to feature integration and categorization has shown its importance in the field of stress detection and management.

In [12], the authors conducted a thorough study of ECG plot pictures, investigating both time and frequency domains using CNN and Long Short-Term Memory (LSTM) architectures. The study revealed appealing performance

data, including accuracies of 94.8% in the time domain and 98.3% in the frequency domain. Notable is the precise characterization of accuracy, sensitivity, and specificity measurements for each domain, which provides insight into the efficacy of the approaches used. The study's focus on binary classification tasks, which used the ST Change and WESAD datasets [13], emphasized its practical applications in healthcare and diagnostic contexts.

Within the scope of [14], the study focused on raw ECG data and used the CNN architecture for classification purposes. The research produced respectable findings, with a stated accuracy of 88.4% and an F1-score of 0.90. Notably, the study used data from the PhysioNet and SWELL databases, which allowed for categorization into three unique groups. This thorough technique demonstrated the resilience and usefulness of the used methodology in detecting small alterations within the dataset.

The study [15] investigated HRV features using Artificial Neural Network (ANN) and Naive Bayes (NB) classifiers. The study revealed impressive performance metrics, including an accuracy of 95.75% on the WESAD and SWELL-KW datasets for binary classification tasks. The full study of HRV characteristics is noteworthy, as it takes advantage of a synergistic method that combines the strengths of both ANN and NB classifiers. This intentional combination highlighted the study's effectiveness in detecting subtle patterns within the dataset, increasing its usefulness in therapeutic and diagnostic contexts. The research conducted [16] focused on the analysis of HRV features using a Support Vector Machine (SVM) classifier. The study provided insights into the dataset, with a reported accuracy of 72.82% on the SWELL-KW dataset [17] for binary classification tasks. Notably, the study's emphasis on HRV characteristics highlighted their importance in detecting minor alterations within the dataset, hence increasing its usefulness in therapeutic and diagnostic situations.

Using CNN architecture, a thorough study of HRV Features was initiated in the [18]. On the Spider Fear dataset, the study produced impressive performance metrics: 83.29% accuracy, 85% precision, and 82% recall for classifying the data into three different categories. Of particular note is the careful characterization of the accuracy, recall, and sensitivity measures, which sheds light on how well the used algorithm distinguishes minute differences in the dataset. This thorough analysis highlighted how important the study was in clarifying subtle patterns in the dataset, which increased its use in diagnostic and clinical contexts.

III. METHODOLOGY

The methodology includes custom data collection and preprocessing, selection of model architecture, and concluded results as shown in Figure 1.

A. Data Collection

The dataset employed in this research comprises raw sensor data recorded using a chest-worn device (RespiBAN) and a wrist-worn device (Empatica E4). Synchronization of these devices was achieved by having subjects perform a double--

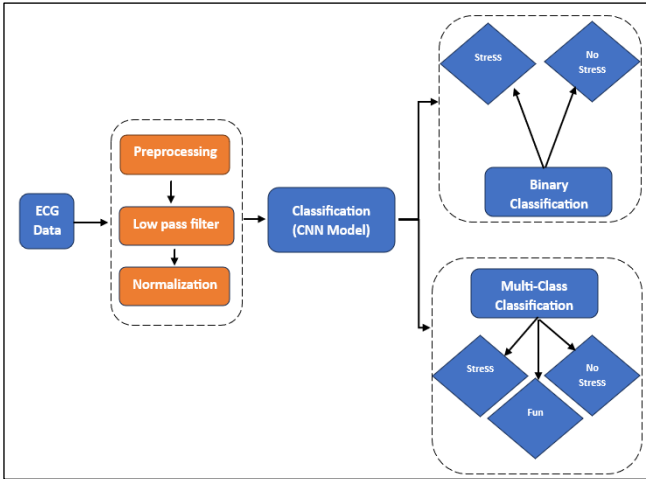


Figure 1: Block Diagram of the used methodology

tapping gesture on their chest, creating a characteristic pattern in the acceleration signal. The synchronized raw sensor data and labels were stored in files labeled SX.pkl. The dataset includes various physiological modalities such as ACC (acceleration), ECG, EDA (electrodermal activity), EMG (electromyography), RESP (respiration), TEMP (temperature), and BVP (blood volume pulse). Labels were assigned to different study protocol conditions, with 0 = not defined / transient, 1 = baseline, 2 = stress, 3 = amusement, 4 = meditation, and 5/6/7 = disregarded conditions. Ground truth information was available in SX_quest.csv.

B. Data Preprocessing

1) Data extraction

From ECG recordings the data for binary classification focusing on stress and baseline conditions, and for multiclass classification stress, baseline, and amusement conditions were extracted from the dataset. There are different duration signals for each class then the signal was chunked to specific time durations of 30 seconds, 20 seconds, 15 seconds, 10 seconds, 5 seconds, and 3 seconds, these all are used for creating different datasets of different duration and finding the best time for stress prediction. The WESAD dataset ECG signal frequency is 700Hz this is used as one dataset and then different sampling frequencies 350Hz, 250Hz, 200Hz, and 100Hz were experimented with to find the optimal configuration, and best model using these all datasets.

2) Removing Noise from signals

We implemented a bandpass filter to increase the quality of the ECG data. As shown in Figure 2, this filter was designed to allow frequencies ranging from 0.5Hz to 50Hz while rejecting others. We effectively reduced high-frequency noise from the data, retaining only the desired frequency range for further analysis.

3) Normalizing data

The clean and preprocessed data are then normalized using the following mathematical formula (1) bring the data in the range of 0 to 1 here.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

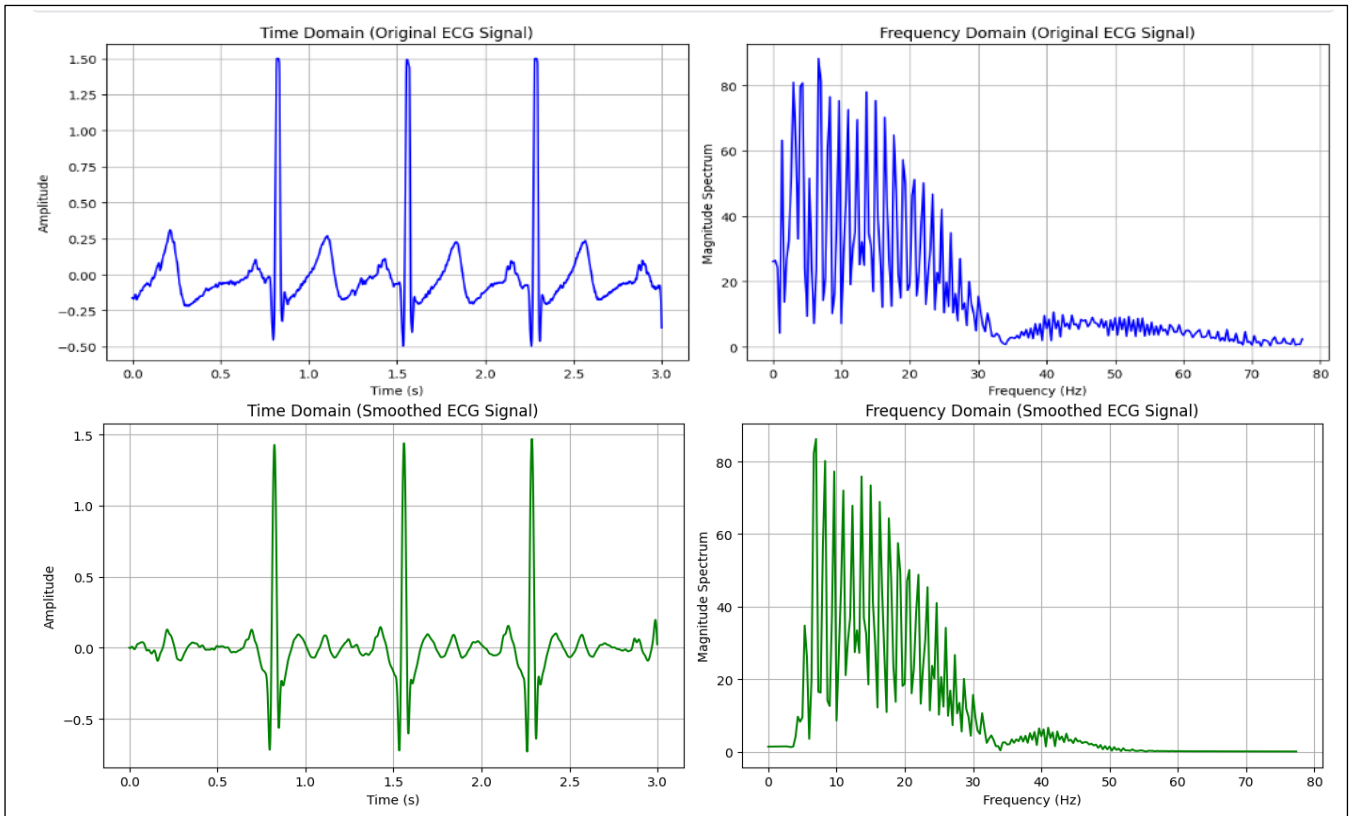


Figure 2: Raw and Filtered ECG signals

Where min and max are the minimum and maximum values in the dataset. This normalized dataset is used as input to the model.

C. Model Architectures and Selection

Several neural network architectures were explored, including CNN, Long Short-Term Memory (LSTM) [19], and combinations like ANN with LSTM [20] and CNN with LSTM, Resnet34, and ResNet50 [21]. Each of these models was trained on all datasets, as after preprocessing we get datasets 30 seconds dataset with 700Hz, 350Hz, 250Hz, 200Hz, and 100Hz and the same for 15 seconds, 10 seconds 5 seconds, and 3-second datasets. We have a total of 25 datasets and we applied each model on each dataset to get the best dataset duration and frequency and the best model that is less computational and accurate. And then we have the same datasets for multiclass classification.

For binary classification the selected CNN architecture shown in Figure 3 exhibited superior performance, achieving a training accuracy of 96.07%, a validation accuracy of 95.04%, and a test accuracy of 94.59%. while for multiclass classification the same CNN architecture shown in Figure 4 exhibits the best result 93.38% on training data, 88.67% on validation data, and 87.60% on test data. The complexity of the model depends on the size of the fed input sample size, this 5-second size and 200 Hz frequency selection methodology is 7 times more computationally efficient than the existing methodologies. For stress score prediction the binary model was used as the sigmoid activation function in the output layer was employed to predict stress scores in the range of 0 to 1, then it's multiplied by 100 to ensure a range from 0 to 100. A stress score of 100 means high stress and 0 means no stress.

IV. RESULT AND DISCUSSION

To detect stress levels in real-time, we created a deep neural network and compared its performance to more traditional methods that rely on manually built features. We proposed a 1D-CNN base model that takes the Raw ECG data of 5

seconds and a frequency of 200Hz. We implement different models for getting the optimal model for the data, the dataset of 30 seconds is used and the ANN model is trained the result of the ANN model for the 30 seconds datasets 700Hz, 350Hz, 250Hz, 200Hz, and 100Hz has accuracy 67.43%, 68.43%, 66.56%, 71.21% and 69.21% respectively. For multiclass classification the accuracy for this dataset of 30 seconds with frequencies of 700Hz, 350Hz, 250Hz, 200Hz, and 100Hz having an accuracy of 57.13%, 59.93%, 61.56%, 62.91%, and 59.01% respectively. All the models were trained in the same way CNN model with 9 layers and with batch normalization of each layer used for all the data the result is for 30 seconds it had the highest accuracy for 250Hz and the accuracy for this was 83.34% for binary and 78.67% for multiclass classification. For 15 seconds dataset, the CNN has the highest accuracy for 200Hz, 82.23% for binary, and 72.39% for multiclass classification, for the dataset of 10 seconds the 200Hz has good results, 90.32% for binary and 82.32% for multiclass classification, for the dataset of 5 second the model has the accuracy of 95.04 for binary and 88.67% for multiclass and the frequency for this result was 200Hz and this is our selected data and model. We also tried 3 second dataset with all frequencies but its result was not good as 5 seconds. The dataset for the 5 seconds contains 6595 examples for binary and 7987 examples are used for multiclass classification. As we have an additional feature in our method which is stress score prediction for that purpose, we used a binary model and the output layer used the sigmoid activation function (2).

$$S(z) = \frac{1}{1 + e^{-z}} \quad (2)$$

This equation gives the value from 0 to 1 range to classify it as no stress or stress the threshold is 0.5 and then predicts the stress score from 0 to 100 using that model output multiplied by 100. The final training and validation accuracy and loss plots are in Figure 5 for binary classification.

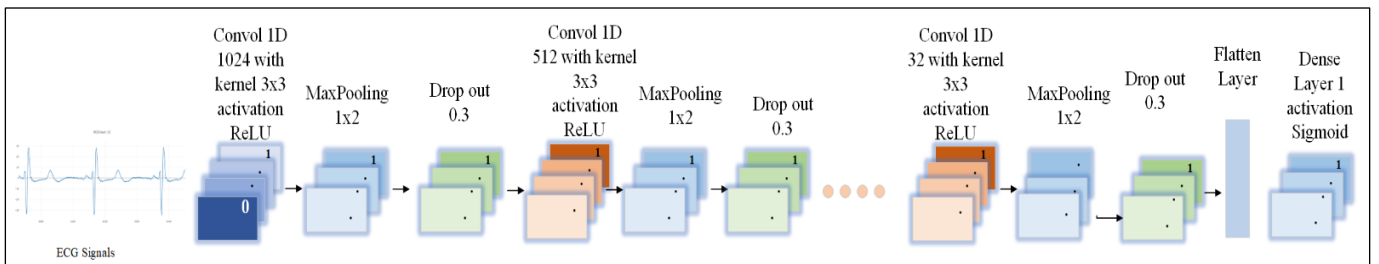


Figure 3: 1-D CNN architecture for binary class classification and prediction.

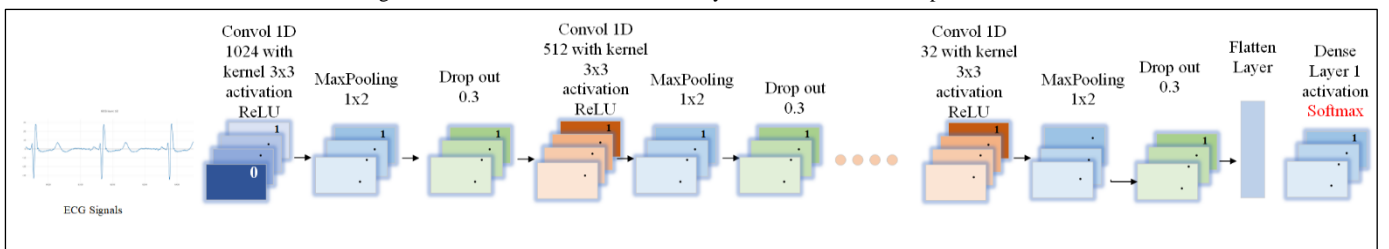


Figure 4: 1-D CNN architecture for multiclass classification and prediction.

The performance of the model is determined by the accuracy (3), precision (4), F1 score (5), sensitivity (6) and specificity (7).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$F1 \text{ Score} = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (5)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (6)$$

$$Specificity = \frac{TN}{TN + FP} \quad (7)$$

True positive (TP) = the number of cases accurately identified as stress.

False positive (FP) = the number of cases wrongly diagnosed as stress.

True Negative (TN) = the number of instances correctly diagnosed as having no stress.

False negative (FN) = the number of cases mistakenly categorized as "no stress."

The performance of the binary and multiclass model in terms of accuracy, precision, F1 score, sensitivity and specificity are mentioned in table 1.

The confusion matrix of the binary model is on validation test data is shown in the Figure 5.

Figure 6 shows the multiclass classification model's confusion matrix based on validation and test data.

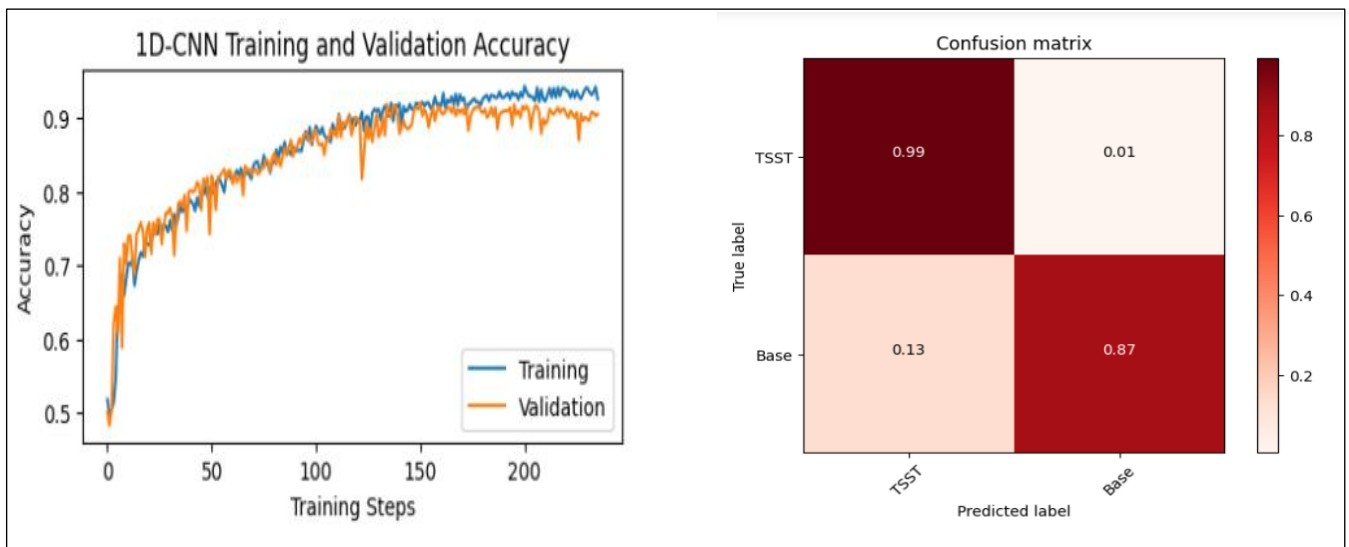


Figure 5: Performance metrics of the binary classification model on 5 seconds 200Hz dataset

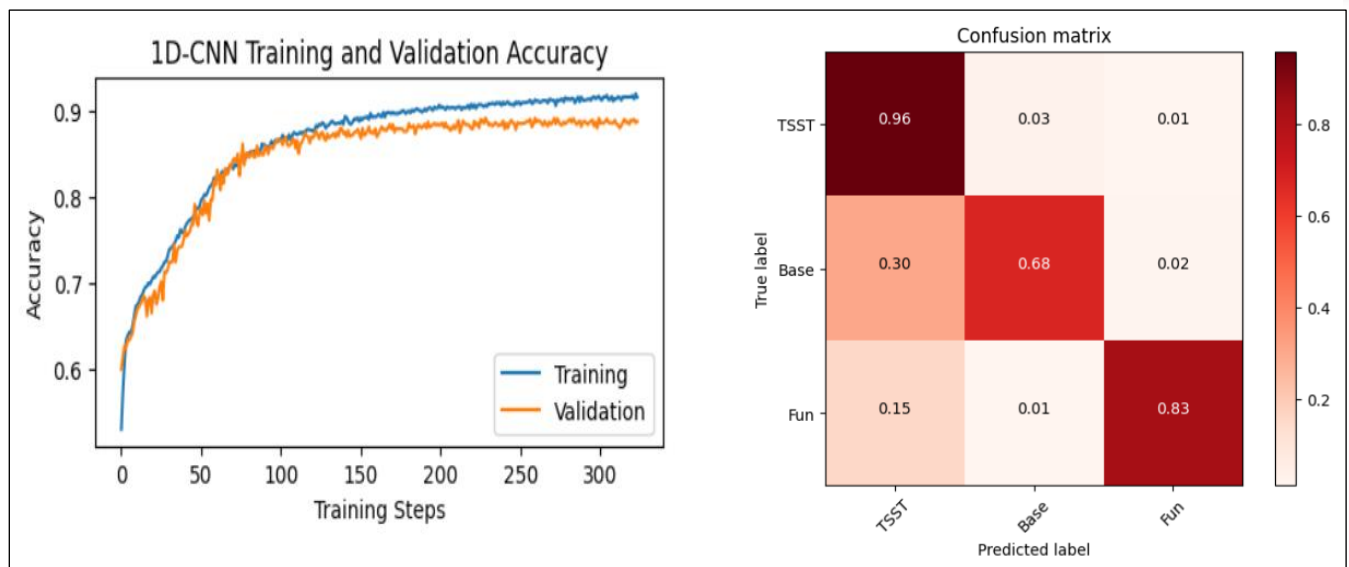


Figure 6: Performance metrics of multiclass classification model on 5 seconds 200Hz dataset.

Performance Matrices	Binary Model	Multiclass Model
accuracy	95.04%	88.10%
precision	95.27%	87.60%
F1 score	94/95%	87.35%
sensitivity	86.69%	95.97%
specificity	99.44%	79.23%
PPV	98.96%	85.55%
NPV	93.64%	78.97%

Table 1: Different evaluation metrics of binary and multiclass model.

V. CONCLUSION

In this study, we have developed a deep neural network-based approach for real-time stress detection utilizing electrocardiogram (ECG) data. By comparing our proposed 1D-CNN model against traditional methods relying on manually engineered features, we demonstrated superior performance in stress prediction. Our methodology involved rigorous experimentation with different dataset durations and sampling frequencies, aiming to optimize model accuracy and computational efficiency.

Through extensive model selection and evaluation, we found that a Simple 1D Convolutional Neural Network architecture yielded the best results for both binary and multiclass stress classification tasks. Specifically, our model achieved notable accuracies across various dataset configurations, with the highest accuracy obtained for a 5-second dataset sampled at 200Hz, demonstrating the effectiveness of our approach in capturing temporal dynamics of stress patterns.

Additionally, we introduced an innovative aspect to our methodology by incorporating stress score prediction, enabling a finer-grained understanding of stress levels ranging from 0 to 100. Leveraging the sigmoid activation function in the output layer of our binary model, we accurately predicted stress scores, further enhancing the utility of our approach for comprehensive stress assessment. Our study underscores the importance of leveraging deep learning techniques for stress detection, offering valuable insights into individuals' well-being and mental health. The ability to predict stress levels in real-time has significant implications for personalized stress management interventions and improving overall quality of life. Moving forward, further research may explore additional physiological modalities and sensor data fusion techniques to enhance the robustness and generalizability of stress detection models in diverse real-world settings.

REFERENCES

[1] A. Mariotti, "The effects of chronic stress on health: new insights into the molecular mechanisms of brain-body communication," *Future Sci OA*, vol. 1, no. 3, Nov. 2015, doi: 10.4155/FSO.15.21.

[2] S. L. Sauter, L. R. Murphy, and J. J. Hurrell, "Prevention of work-related psychological disorders: A national strategy proposed by the National Institute for Occupational Safety and Health (NIOSH).," *Work and well-being: An agenda for the 1990s.*, pp. 17–40, Oct. 2004, doi: 10.1037/10108-002.

[3] "electrophysiology-1996-heart-rate-variability".

[4] M. A. Serhani, H. T. El Kassabi, H. Ismail, and A. N. Navaz, "ECG Monitoring Systems: Review, Architecture, Processes, and Key Challenges," *Sensors (Basel)*, vol. 20, no. 6, Mar. 2020, doi: 10.3390/S20061796.

[5] P. Zhang *et al.*, "Real-Time Psychological Stress Detection According to ECG Using Deep Learning," *Applied Sciences 2021, Vol. 11, Page 3838*, vol. 11, no. 9, p. 3838, Apr. 2021, doi: 10.3390/APP11093838.

[6] Z. Ahmad, S. Rabbani, M. R. Zafar, S. Ishaque, S. Krishnan, and N. Khan, "Multilevel Stress Assessment from ECG in a Virtual Reality Environment Using Multimodal Fusion," *IEEE Sens J*, vol. 23, no. 23, pp. 29559–29570, Dec. 2023, doi: 10.1109/JSEN.2023.3323290.

[7] P. Zhang *et al.*, "Real-Time Psychological Stress Detection According to ECG Using Deep Learning," *Applied Sciences 2021, Vol. 11, Page 3838*, vol. 11, no. 9, p. 3838, Apr. 2021, doi: 10.3390/APP11093838.

[8] R. Zhou *et al.*, "ECG-based biometric under different psychological stress states," *Comput Methods Programs Biomed*, vol. 202, p. 106005, Apr. 2021, doi: 10.1016/J.CMPB.2021.106005.

[9] K. Tzevelekakis, Z. Stefanidi, and G. Margetis, "Real-Time Stress Level Feedback from Raw Ecg Signals for Personalised, Context-Aware Applications Using Lightweight Convolutional Neural Network Architectures," *Sensors (Basel)*, vol. 21, no. 23, Dec. 2021, doi: 10.3390/S21237802.

[10] P. Karthikeyan, M. Murugappan, and S. Yaacob, "DETECTION OF HUMAN STRESS USING SHORT-TERM ECG AND HRV SIGNALS," <https://doi.org/10.1142/S0219519413500383>, vol. 13, no. 2, Apr. 2013, doi: 10.1142/S0219519413500383.

[11] A. Hemakom, D. Atiwiat, and P. Israsena, "ECG and EEG based detection and multilevel classification of stress using machine learning for specified genders: A preliminary study," *PLoS One*, vol. 18, no. 9, p. e0291070, Sep. 2023, doi: 10.1371/JOURNAL.PONE.0291070.

[12] M. Kang, S. Shin, J. Jung, and Y. T. Kim, "Classification of Mental Stress Using CNN-LSTM Algorithms with Electrocardiogram Signals," *J Healthc Eng*, vol. 2021, 2021, doi: 10.1155/2021/9951905.

[13] P. Schmidt, A. Reiss, R. Duerichen, and K. Van Laerhoven, "Introducing WeSAD, a multimodal dataset for wearable stress and affect detection," in *ICMI 2018 - Proceedings of the 2018 International Conference on Multimodal Interaction*, Association for Computing Machinery, Inc, Oct. 2018, pp. 400–408. doi: 10.1145/3242969.3242985.

[14] L. Angrisani *et al.*, "ECG-Based Stress Detection and Productivity Factors Monitoring: The Real-Time Production Factory System," *Sensors 2023, Vol. 23, Page 5502*, vol. 23, no. 12, p. 5502, Jun. 2023, doi: 10.3390/S23125502.

[15] M. R. S. Zawad, C. S. A. Rony, M. Y. Haque, M. H. Al Banna, M. Mahmud, and M. S. Kaiser, "A Hybrid Approach for Stress Prediction from Heart Rate Variability," *Lecture Notes in Networks and Systems*, vol. 519 LNNS, pp. 111–121, 2023, doi: 10.1007/978-981-19-5191-6_10/COVER.

[16] S. Sriramprakash, V. D. Prasanna, and O. V. R. Murthy, "Stress Detection in Working People," *Procedia Comput Sci*, vol. 115, pp. 359–366, Jan. 2017, doi: 10.1016/J.PROCS.2017.09.090.

[17] S. Koldijk, M. Sappelli, S. Verberne, M. A. Neerincx, and W. Kraaij, "The Swell knowledge work dataset for stress and user modeling research," *ICMI 2014 - Proceedings of the 2014 International Conference on Multimodal Interaction*, pp. 291–298, Nov. 2014, doi: 10.1145/2663204.2663257.

[18] A. Vulpe-Grigorasi and O. Grigore, "A Neural Network Approach for Anxiety Detection Based on ECG," *2021 9th E-Health and Bioengineering Conference, EHB 2021*, 2021, doi: 10.1109/EHB52898.2021.9657544.

[19] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Comput*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/NECO_A_01199.

[20] N. Faris Ali and M. Atef, "An efficient hybrid LSTM-ANN joint classification-regression model for PPG based blood pressure monitoring," *Biomed Signal Process Control*, vol. 84, p. 104782, Jul. 2023, doi: 10.1016/J.BSPC.2023.104782.

[21] B. Koonce, "ResNet 50," *Convolutional Neural Networks with Swift for Tensorflow*, pp. 63–72, 2021, doi: 10.1007/978-1-4842-6168-2_6.

Beyond CNNs: Encoded Context for Image Inpainting with LSTMs and Pixel CNNs

Taneem Ullah Jan

Computer Science & IT

University of Engineering and Technology Peshawar

Peshawar, Pakistan

taneemishere@gmail.com

Ayesha Noor

Computer Science & IT

University of Engineering and Technology Peshawar

Peshawar, Pakistan

ayesha.noor2324@gmail.com

Abstract—Our paper presents some creative advancements in the image in-painting techniques for small, simple images for example from the CIFAR10 dataset. This study primarily targeted on improving the performance of the context encoders through the utilization of several major training methods on Generative Adversarial Networks (GANs). To achieve this, we upscaled the network Wasserstein GAN (WGAN) and compared the discriminators and encoders with the current state-of-the-art models, alongside standard Convolutional Neural Network (CNN) architectures. Side by side to this, we also explored methods of Latent Variable Models and developed several different models, namely Pixel CNN, Row Long Short Term Memory (LSTM), and Diagonal Bidirectional Long Short-Term Memory (BiLSTM). Moreover, we proposed a model based on the Pixel CNN architectures and developed a faster yet easy approach called Row-wise Flat Pixel LSTM. Our experiments demonstrate that the proposed models generate high-quality images on CIFAR10 while conforming the L2 loss and visual quality measurement.

Index Terms—Image Inpainting; GAN; Pixel CNN; LSTM

I. INTRODUCTION

Image in-painting involves reconstructing damaged or missing parts within an image, commonly applied in the restoration of old photos or paintings and image editing tasks. Notably, tools like Photoshop feature a robust completion tool that doubles as a removal tool. Despite Convolutional Neural Networks (CNNs) surpassing human classification accuracy on ImageNet [1], in-painting outcomes still fall short of human predictions. The challenge lies in the vast number of possible ways to fill an $8 \times 8 \times 3$ section, around 50,000 possible ways, while ImageNet has only 32,000 classes. Interestingly, humans effortlessly mentally reconstruct missing image sections by comparing context with their knowledge of the world, enabling scene and object recognition, as well as extrapolating missing elements from memory. Artificial and computer based methods leverage similar principles in their approach.

There exist two categories of approaches: local methods [2] [3] solely rely on contextual information, such as color or texture, and aim to extend and blend these details seamlessly. These techniques demand minimal previous perception and training. For instance, if there are no eyes on the head part, a local method might replace it with a patch of skin-textured

pixels. However, these methods face limitations in scenarios where larger patches are absent, excelling primarily in tasks like watermark removal. On the other hand, more advanced methods adopt a global, context-based, and semantic approach [4] [5] [6]. These methods identify patterns within images, like a door or a cabin, which leverage this understanding to infuse the empty spots. Unlike local techniques, they hold the importance of specific elements, such as the necessity of a nose in a particular facial position, constructing a fitting replacement based on their broader knowledge of the context.

An attractive aspect of such problems lies in the effortless generation of extensive datasets for training. Some datasets of images like ImageNet and CIFAR10 [7], which we used for our convenience, can be readily pre-processed by introducing alterations to the images. This approach allows the generation of hundreds of millions of training examples, enabling the training of larger deep networks.

A. The Problem

Every image is divided into two segments: the portion that is absent and under reconstruction, and the contextual part. To enhance simplicity, we suppose that the section which is missing, is a square of dimension $n \times n$. However, it is worth mentioning that the functionality of the network remains consistent even when dealing with arbitrary removals. See figure 1.

The concept of image in-painting is commonly presented as a constrained image generation challenge. The network is tasked with receiving a contextual input and generating an image with identical dimensions as the absent patch. The ultimate assessment hinges on the average element-wise L2 [8], the distance between the original missing section $Y \in R^{n \times n \times 3}$ and the predicted counterpart $\hat{Y} \in R^{n \times n \times 3}$. In our illustrative instance using CIFAR10, $n = 8$. For a given sample i , the loss is calculated as follows:

$$L = \frac{1}{n^2} \sum_{p,q,r} \left(Y_{p,q,r}^{(i)} - \hat{Y}_{p,q,r}^{(i)} \right)^2 \quad (1)$$

We introduce an evaluation metric termed approximate exact-match (AEM), which is solely used for assessment

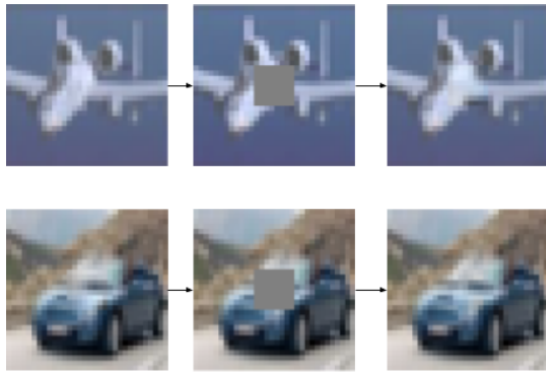


Fig. 1. Image In-Painting

purposes. We observe that a slight move of one or two elements in the value of the pixel channel has minimal visible impression, the effect can be seen in figure 2. Therefore, if a calculated pixel value falls from the accurate image value in the range of ± 5 at each channel, this qualifies as the same and or equal. We present the mean-AEM, denoted as MAEM, where a value of 100% implies that the visual impression of the image is almost identical to the original image, a simpler version of Generated Image Quality Assessment (GIQA) [9].



Fig. 2. On the left is ground truth example and on the right is a ± 5 randomly added to each pixel channel

In-painting comes in two forms: blind [10] [11], where the network lacks information about the position and shape of the missing area. On the other hand in the non-blind [12], such details are provided within the inputs. Extensive research indicates that blind in-painting poses a hard challenge. While non-blind in-painting is more extensively documented, there remains considerable scope for enhancement. Consequently, our emphasis is placed on the latter, reflecting a deliberate choice to concentrate efforts on non-blind in-painting, recognizing its potential for further advancements.

Our primary goal is to make use of the latest computer vision methodologies to develop a resilient and well run in-painter. Here we aimed to attain satisfactory outputs in the form of mean square error or L2 loss, benchmarked against current models. Initial results and the existing methods indicate minimal empirical distinctions between utilizing a square-shaped mask and employing randomly selected rectangular shape masks in the middle of given images. Therefore, for

implementation simplicity, our focus primarily revolves around centered square-shaped cover ups of a consistent size. Specifically, on dataset like CIFAR10, this involves the removal of a patch from every image at the center of size 8×8 .

II. RELATED WORK

Various researchers have investigated a diverse range of methods to tackle such challenges. A notable work by Pathak et al. [13], from where our initial inspiration took root. They adapted the conventional Generative Adversarial Network (GAN) [14] model by incorporating contextual information of image, rather than using stochastic noise, for predicting the incomplete section. Highlighting the importance of Leaky ReLU as detailed in [15] within the discriminator, and exclusion of pooling layers, they implemented compression and decompression operations with strides differing from 1. Their model training involved a combination of L2 loss and adversarial loss, measuring the success of the generator in deceiving the discriminator. However, our study revealed that this increased the risk of overfitting by utilizing fully connected layers in several instances. On the other hand, they have used relatively simple CNN model architecture as for encoders and decoders. In our work, our aim was to not only explore this but also investigate modern methodologies renowned for state-of-the-art results in domains like VGG [16], Inception [17], and others similar.

In a publication [18], their proposed approach revolves around modeling images as the conditional probability product distribution. Here the objective was to calculate the image pixels in sequential order, such as to the bottom-right point from the top-left. These functions which calculate conditional probability are shaped to either capture the contextual information of pixels within the upper rows with the help of recurrent networks or by employing CNNs to operate on local pixels (Pixel CNN [18]). Although originally designed for image generation, this method proves adaptable for our objective of optimizing the probability of the reconstructed image by providing pixel data.

Yang et al. [19] present a technique aimed at addressing the in-painting of large sections within extensive images. Traditional models encounter challenges in producing sharp results for such tasks, often resulting in blurry outputs with noticeable edges between the contextual information and the reconstructed region. The authors reduce this issue by incorporating hierarchical approaches to introduce fine-grained features above the regenerated spots, enhancing the resolution of in-paintings. Their approach involves training two distinct networks; first is a feature extractor that is assessed using a content-based loss same as in [13], second is texture-based network which employs minimizing the texture-based local loss. The incorporation of perceptual equivalence ensures that the patterns within the generated spots closely align with the context of local texture. This strategy significantly improves visual sharpness, it may be considered unnecessary in our case, given the context of working with smaller images.

The structure and process of our work represent a substantial adaptation of the CIFAR10 classification pipeline, originally derived from resources. The only retained elements are the queuing system and monitoring session. It is noteworthy that the autoencoders, GANs, and Wasserstein GANs (WGAN) [20] utilized in our study are also publicly available through standard sources. Moreover, there are adaptations of [13] in existence, their inefficiency and non-functionality led us to avoid their use. In the exploration of advanced models such as VGG, Inception, and ResNet [21], we adapted them from model repositories, and made adjustments to tailor them to CIFAR10 specifications.

III. METHODS

A. Dataset

Our primary dataset is CIFAR10, comprising images of dimensions $32 \times 32 \times 32$. It consists of around 50 thousand training and about 10 thousand test samples. Notably, the CIFAR10 has been taken from the Tiny Images dataset [22]. When our model achieves stability, we extend our training efforts to leverage this larger dataset. The inclusion of a more extensive variety of images from the larger dataset proves helpful, particularly given that CIFAR10 is limited to only 10 classes. The substantial number of samples in the larger dataset serves as an efficient countermeasure against overfitting. Our model facilitates seamless scaling for training on this expanded dataset, resulting in high training performance.

Additionally, in the middle we also incorporate data augmentation steps on our CIFAR10 that helps in amplifying the dataset further enhancing the adaptability of this model for minor variations effectively. This involves introducing small, random adjustments to hue, saturation, contrast, and applying random Gaussian blur to the images. Also we put zeros in place of the middle $8 \times 8 \times 3$ crop.

B. Autoencoder

In-painting constitutes a subset of a broader category of image generation problems involving the creation or modification of pixels. Tasks like deblurring, denoising, and small-scale blind in-painting, such as text removal, are commonly addressed using autoencoders. Autoencoders typically consist of two main components, an encoder and a decoder. Initially, an image is encoded in a latent feature space, or embedding. On the other hand, the decoder reconstructs the original image based on this embedding. The training process involves jointly optimizing both networks to decrease the input-output disparity. This architectural configuration requires the encoder to maximize the information encoded into the embedding. The encoder must learn the abstract intelligent features to compress the information in least possible loss, that too within the limited size of this latent space. In CIFAR10 experiments, where input images consist of vectors with a length of 3072, it has been found that employing a bottleneck size of 512 and or 1024 produces satisfactory results. Larger bottlenecks do not provide any attraction to autoencoders to discover accurate representations but keeping raw sections of pixels proves to be

sufficient. On the other hand, using smaller bottlenecks lack sufficient capacity to encode the input information. This can be seen in figure 3.

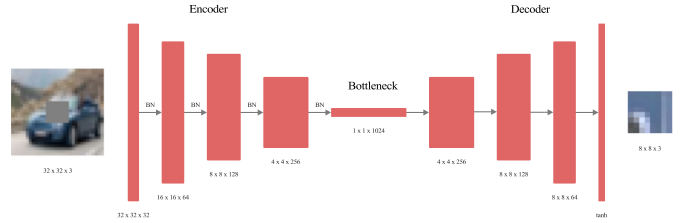


Fig. 3. Autoencoder architecture of our model

The typical ratio between the image and latent space dimension in the existing literature often exceeds than what we employ, ranging from 3 to 6. Empirically, our choice stems from working with small images, introducing a challenge in identifying unified entities and components in the target image. Achieving improved encoding ratios, as seen in ImageNet; 12 or more, for instance, is more feasible because it is comparatively easier to isolate high-level features in larger images like those in ImageNet.

To establish a baseline, we initially constructed a straightforward CNN architecture. The input dimension is $32 \times 32 \times 3$, and the output dimension is $8 \times 8 \times 3$. Guidance from the literature advocates for the use of several small filters rather than larger ones. The rationale behind this approach lies in the ability to achieve an equivalent receptive field with deeper networks. Throughout, we utilized filters of size 3 exclusively. While filters of size 5 and 7 were tested on multiple occasions, their performance consistently lagged behind that of size 3 filters.

Our experimentation involved a $[Conv - Conv - Pool2] \times 3$ architecture, followed by either 2 fully connected layers or 2 convolutional layers. This architecture closely resembles a well-performing design on CIFAR10, featuring smooth dimension reduction coupled with an increment in the number of filters, initiating at 32 or 64 and doubling at each pooling step. Subsequently, we opted to eliminate max pooling by entirely substituting it with convolutional layers of stride 2, maintaining the same filter progression. Transitioning from the $4 \times 4 \times 256$ convolutions to the bottleneck can be conceptualized as a stride 4 operation with 1024 filters. Across all scenarios, ReLU activation consistently outperformed alternatives. The incorporation of Batch Normalization on each layer given substantial performance improvements, approximately around $\pm 15\%$, with comparable execution times.

In terms of qualitative assessment, as previously mentioned, a prominent issue is blurriness: while colors are generally accurate, details and textures tend to be lost, resulting in predicted sections that often resemble indistinct dots, failing to seamlessly blend into the image. To address this challenge, our objective is to diminish visible continuity errors between the predicted section and the context. We aim to achieve this by predicting a patch that slightly overlaps with the context

and imposing a robust penalty on the loss specifically within this overlap region.

C. Generative Adversarial Networks (GANs)

The first figured-out method demonstrated notable L2 loss. Nevertheless, the generated output images consistently exhibited blurriness and a lack of intricate details, as illustrated in figure 4. The inherent nature of L2 loss encourages the network to adopt a risk-averse approach, generating safe predictions characterized by a lack of sharp shapes and substantial changes across the patch. The inclination to generate blurred, average-color images derived from context minimizes the occurrence of substantial errors. Consequently, while the model excels in terms of the L2 norm, the generated outputs fall short of realism when evaluated by a human eye.

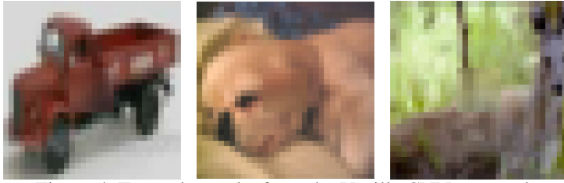


Fig. 4. Results from normal CNN architecture

1) *Deep Convolutional GANs (DCGANs)*: To encourage our network to take more risks and generate realistic outputs, we opted to explore GANs. Here the underlying objective is the emulation of visual and perceptual evaluation and assessment. For instance, if a model predicts a human head without eyes or mouth, this is considered superior to generating an image with a black spot at the center. The rationale is that an averagely blurry image will never appear realistic to a human observer.

In the realm of GANs, the key concept involves training a discriminator network (D) concurrently with a generator network (G). The discriminator learns to assess the authenticity of an image, distinguishing between real and or generated spots and dots. Normally they are on both real and generated examples with distinct labels for each. The generator faces a penalty with an increased loss if the output image is viewed as generated by the discriminator. To outsmart the discriminator, our generator aims to create natural looking and visually clear images. As the discriminator improves, both networks benefit from the feedback loop, driving mutual enhancement.

The introduction of adversarial networks may not necessarily lead to an improvement in L2 loss, as the generator could throw significant errors. Although, the primary objective is to enhance the realism of the generated images. In the end, the crucial aspect is whether the predicted human eye, for example, appears authentic within the context of the image. Even if the L2 loss indicates a substantial difference from the ground truth; it constitutes an acceptable result, if the predicted sections are admissible in the context of the human head. Here the focus shifts from minimizing pixel-wise differences to

creating outputs that are visually convincing and contextually appropriate.

Now, loss $L = \alpha L_{rec} + (1 - \alpha) L_{disc}$, here L_{rec} is the same mean square error, while L_{disc} known as the probabilistic output from the discriminator for the generated images. This also can be called sigmoid cross entropy. Given that changes the scale of the loss, using the loss itself for the optimization of this crucial hyperparameter is impractical. This challenge is to intricately linked to the observation that a lower L2 loss should not automatically correlate with script and visually better output. The choice of is aimed at optimizing the results of our output samples and ensuring. Overall this can be calculated as:

$$L_{disc} = - \sum_i \log(p_i) = - \sum_i \log \left(D \left(\hat{Y}^{(i)} \right) \right) \quad (2)$$

The discriminator, in our setup, is a network consisting of six convolutional layers, having a stride value of 2. After this, there is a final convolutional layer that reduces the depth value to 1, resulting in a normal value output. Conceptually, a typical classification task is addressed by our discriminator. Therefore, various recent methods can allow them to function as a typical generator network. We leveraged pre-trained models, including VGG, Inception, and ResNet, to enhance our discriminator. Since these models are designed for larger inputs like ImageNet, we attempted to pad our images, but the results were unsatisfactory. As a solution, we adjusted these techniques to operate on our smaller inputs. This involved removing the first layers with dimensions higher than 32×32 and injecting our input into the initial smaller layers. To prevent overfitting and improve execution time, we also reduced the depth and the number of filters in consideration of the fact that our input contains less information. The same adaptation strategy was applied to the encoder, where we experimented with a variety of high-performing models.

The discriminator, in our setup, is a network consisting of six convolutional layers, with every other convolutional layer having a stride of 2. This is followed by a final convolutional layer reducing the depth to 2, producing a scalar output. Conceptually, the discriminator can be viewed as addressing a classification problem. Therefore, various state-of-the-art models can be adapted to function as a generator. We leveraged pre-trained models, including VGG, Inception, and ResNet, to enhance our discriminator. Since these models are designed for larger inputs like ImageNet, we attempted to pad our images, but the results were unsatisfactory. As a solution, we adapted and retrained these models to operate on our smaller inputs. This involved removing the first layers with dimensions higher than 32×32 and injecting our input into the initial smaller layers. To prevent overfitting and improve execution time, we also reduced the depth and the number of filters in consideration of the fact that our input contains less information. The same adaptation strategy was applied to the encoder, where we experimented with a variety of

high-performing models.

2) *Learning tricks*: As usual we encountered challenges in effectively training a GAN. Visual inspection of the predicted images revealed the presence of colored artifacts, as illustrated in figure 5. Upon closer inspection, it became apparent that shapes and color gradients aligned with the context, but the colors were distorted, particularly noticeable in the case of the plane. Due to difficulties in achieving satisfactory convergence with GANs, the L_2 loss was higher compared to vanilla autoencoders. The current architecture exhibits a sensitivity to randomness, where the exact same method may result in convergence or divergence under different circumstances.



Fig. 5. Results from DCGAN architecture

Training GANs poses inherent challenges as they are often unstable and highly sensitive to network architectures and parameters setting. Finding the right hyperparameters and architecture details can be a tedious task. Another common issue encountered is the imbalance between the discriminator and the generator, where one may exceed and overwhelm the other. As an example, if our network's discriminator becomes excessively robust, then our generator will struggle to deceive it, leading to a scenario where the adversarial loss sharply rises at high values, increasing the L_2 loss. Conversely, if the generator becomes too dominant, it can successfully outsmart the discriminator, causing it to halt learning and assign identical probabilistic values to both fake and real samples. Then, it is symmetrical to have no adversarial loss. To address these challenges, we experimented with a few learning tricks aimed at mitigating these issues.

- **Separate Batches**: We provided two distinct batches: one containing solely real samples and the other composed entirely of fake samples. This approach facilitates clearer and more direct updates for each batch, allowing the network to focus distinctly on improving its understanding of real examples and enhancing the detection of fake ones.
- **Soft and Noisy Labels**: Soft and noisy labels were employed following the approach outlined in [23]. True images were assigned random labels between 0.9 and 1.1, while fake images received labels ranging from 0 to 0.2. Additionally, labels were occasionally flipped randomly between classes. These strategies introduce noise and enhance the robustness of the discriminator, contributing to improved training stability.
- **Maximizing $\log D$ Instead of Minimizing $\log(1 - D)$** : Rather than minimizing $\log(1 - D)$, we chose to

maximize $\log D$. Here, D represents the discriminator's output. While these formulations are equivalent in this context, the latter avoids the issue of vanishing gradients early in the training process, enhancing the stability and effectiveness of the learning procedure.

3) *WGANs*: To address the challenges in training GANs, as discussed above, WGANs [20] have also been explored as an alternative to traditional GANs, relying on the Earth-Mover distance [24] for learning distributions. WGANs minimize the Earth-Mover distance, allowing for the estimation of this metric during training, which correlates well with visual quality. WGANs also eliminate the need for a realness probability from the discriminator, using an unbounded score, and enforce Lipschitz continuity through gradient clipping. Importantly, WGANs do not require a delicate balance between the generator (G) and discriminator (D), allowing for training the discriminator to convergence at each step. In practice, we train the discriminator 10 times at each iteration for stability and convergence.

4) *Reducing continuity errors*: The adoption of WGANs marked a significant improvement in training a robust adversarial network. However, a noticeable issue persisted; the distinct visibility of the border between the context and the reconstructed region. Despite the challenge in visually explaining this clear border appearance, a detailed analysis revealed the absence of an organized color correction with pixels was highly noticeable, but a random error tended to blend more seamlessly, refer to figure 2. This border problem remained even in instances with otherwise accurate predictions.

To address this issue, we implemented a clever trick: predicting a 16×16 center square that encompasses the original 8×8 target along with a 4-pixel overlap on each side. While we penalized our reconstruction loss more than 20 times, it is relatively simple for the model to accurately calculate and predict it. Here this strategic approach significantly enhances the quality of border merging. By ensuring that there are no major discontinuities within the output, this effectively reduced gaps between the input and prediction. It is important to note that, for visualization purposes, we retain and display only the 8×8 target in the final results.

Prior to using this strategic trick, feeding the entire image to the discriminator posed challenges, as the obvious border served as a key indicator for determining the authenticity of the image. However, with the introduction of the overlap and the subsequent reduction in the visibility of borders, we successfully transitioned to providing the complete image to the discriminator. In this improved setting, the discriminator could assess the entire image as a cohesive unit. Notably, an 8×8 square, either from a real or fake image, typically lacks sufficient meaningful information on its own, making it challenging for the discriminator to make a decisive meaning. In contrast, evaluating the entire image facilitates a clear and more straightforward return for the discriminator. Overall the results can be seen in figure 6.



Fig. 6. On the left is the result without overlap trick and on the right is with overlap

D. Density based models

1) *The Idea*: We opted to implement unsupervised models alongside our supervised methods for filling the contextual portion in the center of an image. In this approach, our objective is to approximate the function of Probability Mass (PMF), p of all images by multiplying conditional probabilities. Therefore, for a given image x of dimension $n \times n$, where the pixels are denoted as $\{x_1, x_2, \dots, x_{n^2}\}$, the PMF is expressed as:

$$p(x) = \prod_{i=1}^{n^2} p(x | x_i, \dots, x_{i-1}) \quad (3)$$

Where $p(x | x_i, \dots, x_{i-1})$ denotes the probability of the i^{th} pixel or x_i assuming the values x_1, \dots, x_{i-1} have been observed in the previous pixels.

Each pixel is represented by a triplet of integers as: $(R, G, B) \in \{0, 256\}^3$. Now using this notation $x_i = (x_{i,R}, x_{i,G}, x_{i,B})$ and $x < i = (x_1, \dots, x_{i-1})$, the conditional probabilities can be written as:

$$p(x_i | x < i) = p(x_{i,R} | x < i) \times p(x_{i,G} | x < i, x_{i,R}) \times p(x_{i,B} | x < i, x_{i,R}, x_{i,G}) \quad (4)$$

Therefore the final probability $p(x)$ is obtained by multiplying $3n^2$ terms. Consequently, while in-painting images, our aim is to populate the space with the pixels by maximizing this probability.

The approach we take is greedy, which means that assuming we have learned the characteristics of the function p and are given an image with m missing pixels at positions $i_1 < i_2 < \dots < i_m$, we will first set the value $x_{i_1,R}$ given $x < i_1$, then the value of $x_{i_1,G}$ given $(x < i_1, x_{i_1,R})$, and eventually we set $x_{i_1,B}$, given $(x < i_1, x_{i_1,R}, x_{i_1,G})$. Hence, as this approach is easy to implement and is computationally efficient, there is no assurance that our reconstructed image maximizes the likelihood $p(x)$.

An intuitive approach is to establish an order on the pixels, progressing from the top-left corner to the bottom-right corner, scanning rows consecutively. Once the order is defined, the next step involves selecting the type of conditional probability to be learned. We will introduce two distinct methods, drawing inspiration from [18]. The first method involves a

Pixel CNN, where the conditional probability of a pixel is determined through a convolutional neural network operating on pixels in the surrounding neighborhood. The second method is a flattened row Long Short Term Memory (LSTM), changing significantly from the one proposed in [18]. In this alternative approach, the probability is computed based on pixels from preceding rows, which are then fed into an LSTM network.

2) *Pixel CNN*: In order to calculate the conditional probability $p(x_i | x < i)$, a convolutional neural network is implemented. Various architectures were explored, incorporating combinations of convolutional and leaky ReLU layers. The model concludes with three fully connected layers, each followed by a softmax layer corresponding to the three color channels (R, G, B). Our model applies softmax loss function, computing the estimated probabilities for the true pixel values $\hat{p}_{i,true}$ in an image with n^2 , are computed as:

$$L_{\text{softmax}} = - \sum_i^{n^2} \log(\hat{p}_{i,true}) \quad (5)$$

An alternative perspective argues that the softmax loss may not be directly applicable, considering that the task at hand is not a classification one. In this context, calculating the value of 117 rather than 118 could not be as crucial as predicting 117 instead of 245. While it is obvious that a low probability $\hat{p}(x_{i,R} = 120)$ may not pose a significant issue. If the emphasis on values within the range [112, 124] is sufficiently enough, the softmax loss function has shown interesting results, even given the extensive training examples compared to the 256 possible values for each channel. The model we came up with finally incorporates a 5x5 section within the image for the initial convolutional layer, positioned at the top-left corner of the pixel grid. See figure 7.

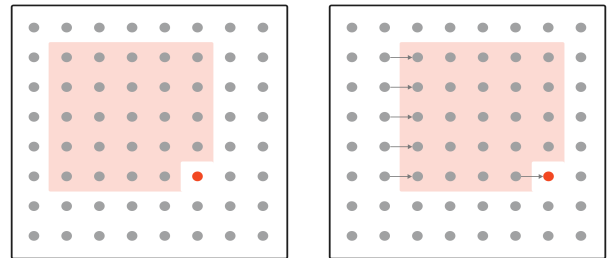


Fig. 7. The red pixels are surrounded by gray dots, these are used for predicting the conditional distribution which form the neighborhood. Subsequently, to estimate the distribution of the next pixel we the same network, shifted one step to the right

Our method offers the advantage of utilizing a similar model, not only this but provides a similar set of learning parameters to compute the final conditional probabilities for each. The complexity remains persistent, irrespective of the image size, as we consistently use the same number of pixels;

24 to calculate the distribution of the next pixel. However, in the case of resolutions, like 128×128 images, this may be necessary to employ a greater relative section for more accurate predictions.

The outcomes obtained with our Pixel CNN model exhibit a slightly lower performance compared to GANs when evaluating the L_2 loss. Two factors contribute to this disparity:

- Our Pixel CNN was trained using a softmax loss, which explains the higher L_2 loss value of 6.98, as opposed to the results that came from our GANs.
- Iterating through the image from left to right, the Pixel CNN determines the value of each pixel based on its 24 neighbors located at the top-left corner. Consequently, during the gap-filling process the pixels at the bottom of the image remain unused. See figure 8.



Fig. 8. On the left is ground truth example and on the right is a reconstructed image from the Pixel CNN model

At the bottom and right portions of the image, clearly, the reconstruction of the bird reveals a neglect that is predominantly characterized with colors like light gray and white. This omission is particularly noticeable in the reconstructed square, where the discontinuity is more apparent at the bottom and right sides. Additionally, the limited number of dark feathers initially situated at the top-left corner of the absent section have now expanded to encompass a significant portion of the reconstructed square.

We also recognize the presence of the Row LSTM and Diagonal BiLSTM [25] from the same article. We successfully implemented and executed these methods, which yielded satisfactory results. While we refrain from providing an exhaustive report on these techniques due to their complexity in terms of implementation and performance, we attempted to propose a significantly simpler architecture. Despite its simplicity, this alternative architecture demonstrates the capability to deliver relatively good performance on CIFAR10, enabling insightful analyses.

3) *Flattened Row LSTM*: Finally, we present a model that incorporates data from all preceding rows of the image to compute the probability distribution of each pixel. Termed as the Row Flattened LSTM, the architecture of this model can be seen in figure 9.

The entire image is flattened, executing to the bottom-right from the top-left corner. Supplied to the LSTM are pixel channels, represented as 256-dimensional one-hot vectors, in the

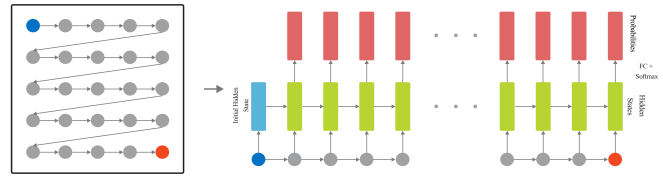


Fig. 9. Row Flattened LSTM Architecture

prescribed RGB sequence. With a set of 64 hidden dimensions, we proceed to convert these hidden vectors into output vectors from a fully connected softmax layer, each comprising 256 dimensions. Our Flattened Row LSTM reconstructed image example can be seen in figure 10.



Fig. 10. On the left is ground truth example and on the right is a reconstructed image from our Row-wise Flat Pixel LSTM

Our obtained result exhibits certain limitations compared to the Pixel CNN model. Several factors may contribute to these observations:

- The model assigns more influence to the pixels immediately on the left of the reconstructed pixel the way the LSTM scans. This differs from the Pixel CNN approach, where we employed a concentrated region of 24 neighboring pixels located at the top-left corner of our target pixel.
- As our current LSTM model may be too simplistic to effectively interpret this information, its strength lies in capturing all information from previous rows before making predictions. Enhancements, such as incorporating multiple LSTM networks or increasing the hidden dimensions, could potentially lead to more accurate predictions.

IV. RESULTS

A. Autoencoder

We used the Adam optimizer for training and conducted cross-validation for the learning rate. Learning rate decay proved beneficial across all our tests, striking a balance between swift initial learning and a gradual reduction in loss during later stages. However, we encountered challenges in cross-validating the decay rate and decay factor due to the intricacies involved in this process. Furthermore, we implemented dropout at a rate of 0.5 for every activation, and our experiments revealed no substantial changes in accuracy

within the reasonable range of values for the probability drop from 40% to 80%. Also, we investigated the impact of the bottleneck size on our fundamental architecture, recognizing it as one of the most critical parameters. But, it is worth noting that the best results, shown in table I; did not emerge from the basic architecture.

TABLE I
TRAIN VS. VALIDATION L2 LOSS

Size of Bottleneck	L2 Loss (Training) ↓	L2 Loss (Validation) ↓
256	7.02	7.96
512	6.36	7.41
1024	4.89	7.48
2048	4.23	8.14

It is apparent that overfitting poses a challenge. Significantly decreasing the size of bottleneck, and consequently it reduces the size of fully connected layer. This leads to a substantial decrease in the number of parameters, this is because this layer constitutes the majority of training parameters of the model. Overfitting diminishes notably from 2048 to 512, with 512 appearing to be the optimal choice, while 256 proves to be too small and potentially insufficiently expressive. Furthermore, to combat over-fitting more effectively, we plan to investigate strategies such as expanding the dataset and employing the data augmentation, integrating explicit L2 regularization, and experimenting with techniques like drop-connect, which involves randomly dropping connections in CNNs during training.

B. WGANs

As previously mentioned, unlike our vanilla autoencoder, GANs do not exclusively optimize for $L2$ loss. Theoretically, $L2$ loss results should not be superior for GANs. However, we dedicated more time to refining GANs because our best $L2$ only architecture gave visually poor results; the optimized loss function and visual quality did not align. Consequently, our GANs exhibit improved results as a consequence of this experimental bias.

Due to challenges in training a Deep Convolutional GAN (DCGAN [26]), we exclusively present results for our implementation of WGAN, which has proven effective. The reported score in the paper was slightly inconsistent, representing a coefficient of 0 for the adversarial loss, making it a normal CNN. The most favorable outcomes were achieved with a VGG-like architecture. It is important to note that we do not argue that this is the optimal architecture, as our exploration of alternative options has been limited.

C. Density methods

Our conditional probability function in the Pixel CNN uses several convolutional layers with dropout. We opted not to include any pooling layers, as their addition did not appear to enhance the quality of our reconstructed images. Given that the parameters of the CNN are shared across all pixels, training the Pixel CNN essentially involves a classification task using a 24-pixel section; $5 \times 5 \times -1$, as explained earlier, as input and an

integer within the range $[0, 256]$ as the target. For optimization, we utilized the Adam Optimizer with cross-entropy loss, and the learning rate was cross-validated. The dropout rate was set at 50%.

Our Flattened Row LSTM model achieved a quadratic loss of 7.63 on the test set, employing the softmax cross-entropy loss. The hidden dimension 64 goes with cross-validation, although due to computational constraints, we could not test as many values as desired. With the incorporation of several LSTM networks, it is likely that we could have achieved a lower $L2$ loss.

D. Comparison

The outcomes of the optimal run for each model type on the test set are depicted in table II below. Additionally, several outcomes from our top-performing run are illustrated in figure 11.

TABLE II
DIFFERENT MODELS' L2 LOSS VALUES

Model	L2 Loss ↓
Row Flattened LSTM CNN	7.63
Pixel CNN	6.98
Wasserstein GAN	4.26
CNN	7.49



Fig. 11. Results from Out-of-sample example

V. CONCLUSION

Our unique CNN-based image in-painter with notable efficiency recognizes the significance of adversarial loss, as we incorporated a GAN into our model. Employing various tricks, we extended the approach from [13] to incorporate WGANs and utilized well-established architectures like VGG, Inception, and ResNet. Additionally, our overlap trick enhances border smoothing and also aids the training of discriminators. Our exploration not only delved into density-based methods, implementing Pixel CNNs based on [18], and introducing our model, Flattened Row LSTM. Through qualitative and quantitative comparisons, we looked to comprehend the limitations of these models. Overall, we are highly content with the outcomes achieved by our proposed architectures.

Future improvements involve adapting our models for larger images, offering exciting possibilities for handling more complex scenes with larger objects. Despite potential performance challenges, scaling up would open directions for addressing intricate scenarios. Additionally, refining our Flattened Row LSTM model to enhance symmetry and reinforce its generative capabilities stands as another goal for future improvement.

REFERENCES

- [1] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115, 211-252.
- [2] Ružić T, Pižurica A. Context-aware patch-based image inpainting using Markov random field modeling. *IEEE Trans Image Process.* 2015 Jan;24(1):444-56. doi: 10.1109/TIP.2014.2372479. Epub 2014 Nov 20. PMID: 25420260.
- [3] K. H. Jin and J. C. Ye, "Annihilating Filter-Based Low-Rank Hankel Matrix Approach for Image Inpainting," in *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3498-3511, Nov. 2015, doi: 10.1109/TIP.2015.2446943.
- [4] Z. Yan, X. Li, M. Li, W. Zuo, S. Shan, Shift-net: Image inpainting via deep feature rearrangement, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 1-17..
- [5] C. S. Weerasekera, T. Dharmasiri, R. Garg, T. Drummond, I. Reid, Just-in-time reconstruction: Inpainting sparse maps using single view depth predictors as priors, in: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 1-9.
- [6] J. Zhao, Z. Chen, L. Zhang, X. Jin, Unsupervised learnable sinogram inpainting network (sin) for limited angle ct reconstruction, *arXiv preprint arXiv:1811.03911*..
- [7] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
- [8] Cortes, C., Mohri, M., & Rostamizadeh, A. (2012). L2 regularization for learning kernels. *arXiv preprint arXiv:1205.2653*.
- [9] Gu, S., Bao, J., Chen, D., & Wen, F. (2020). Giga: Generated image quality assessment. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI* 16 (pp. 369-385). Springer International Publishing.
- [10] Wang, Y., Chen, Y. C., Tao, X., & Jia, J. (2020). Vcnet: A robust approach to blind image inpainting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV* 16 (pp. 752-768). Springer International Publishing.
- [11] Liu, Y., Pan, J., & Su, Z. (2019). Deep blind image inpainting. In *Intelligence Science and Big Data Engineering. Visual Data Engineering: 9th International Conference, IScIDE 2019, Nanjing, China, October 17–20, 2019, Proceedings, Part I* 9 (pp. 128-141). Springer International Publishing.
- [12] C. J. Schuler, H. C. Burger, S. Harmeling and B. Schölkopf, "A Machine Learning Approach for Non-blind Image Deconvolution," 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 1067-1074, doi: 10.1109/CVPR.2013.142.
- [13] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2536-2544).
- [14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
- [15] Xu, B., Wang, N., Chen, T., & Li, M. (2015). Empirical evaluation of rectified activations in convolutional networks. *arXiv preprint arXiv:1505.00853*.
- [16] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [17] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
- [18] Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., & Graves, A. (2016). Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29.
- [19] Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., & Li, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6721-6729).
- [20] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In the *International conference on machine learning* (pp. 214-223). PMLR.
- [21] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- [22] Prabhu, V. U., & Birhane, A. (2020). Large image datasets: A pyrrhic win for computer vision?. *arXiv preprint arXiv:2006.16923*.
- [23] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, 29.
- [24] Rubner, Y., Tomasi, C. & Guibas, L.J. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* 40, 99–121 (2000). <https://doi.org/10.1023/A:1026543900054>
- [25] Van Den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016, June). Pixel recurrent neural networks. In the *International conference on machine learning* (pp. 1747-1756). PMLR.
- [26] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Comparative Study On Harassment Stance Detection Using Social Media Texts Through Deep Learning Methods

Ebtihaj Abdullah

Department of Computer Systems
Engineering, University of
Engineering and Technology Peshawar,
Peshawar, Pakistan
20pwce1885@uetpeshawar.edu.pk

Tooba

Department of Computer Systems
Engineering, University of Engineering
and Technology Peshawar,
Peshawar, Pakistan
20pwce1922@uetpeshawar.edu.pk

Kainat Robi

Department of Computer Systems
Engineering, University of Engineering
and Technology Peshawar,
Peshawar, Pakistan
20pwce1870@uetpeshawar.edu.pk

Waseem Ullah Khan

Secured IOT Devices Lab, University of
Engineering and Technology Peshawar,
Peshawar, Pakistan
waseem@uetpeshawar.edu.pk

Salman Ahmed

Department of Computer Systems
Engineering, University of Engineering
and Technology Peshawar,
Peshawar, Pakistan
sahmed@uetpeshawar.edu.pk

Safdar Nawaz Khan Marwat

Department of Computer Systems
Engineering, University of Engineering
and Technology Peshawar,
Peshawar, Pakistan
safdar@uetpeshawar.edu.pk

Abstract— Social media platforms are becoming pivotal venues for people to express their opinions, share ideas, and participate in debates on a range of topics in an era of information and digital communication. Stance detection of these digital dialogues is a big source of opinion-based data for many digital markets. Stance detection is the method of finding perspective about a certain subject using text related to that subject. However, accurate stance detection confronts specific challenges due to the dynamic nature of language in social media. The nuances, sarcasm, slang, and constantly evolving vocabulary inherent in these platforms are sometimes difficult to capture with traditional methods. To address these challenges and enhance the accuracy of stance detection, there is a need to make use of the developments in Deep Learning which are more accurate and scalable than conventional Machine learning models. Other than that, there has been limited work in detecting harassment in professional as well as general domains using Deep Learning methods. Moreover, there has been no work in detecting Indirect Harassment or quoted here as Harassment Stance in contrast to the conventional direct harassment detection. BERT has been very useful in detecting general stance. This paper discusses the use of BERT which is a deep learning transformer method and to enhance it to detect indirect harassment in professional domains. We then compare that to LSTM in detecting Harassment.

Keywords: Stance Detection; Deep Learning; BiDirectional Encoders Representations from Transformers (BERT); Long Short-Term Memory (LSTM); Harassment; Cyber-Bullying; Natural Language Processing (NLP); Support Vector Machine (SVM), Convolutional Neural Networks (CNN); K Nearest Neighbors (KNN); Microsoft Research Paraphrase Corpus (MRPC); Corpus of Linguistic Acceptability (CoLA)

I. INTRODUCTION

A) Social Media and Background

Today's age is characterized by an endless world of Artificial intelligence or popularly known as AI, a domain in which technology forms and shape every single day on the edge of AI-led epoch which involves surrendering more duties to the computers such as making humanly decisions that entail a level of depth and detail. Today's focus on AI's capability to sense, recognize and interact with human emotions like sentiments and postures highlights this tremendous change of our relationship with technology. The AI revolution brings w interesting feature, being the ability to forecast how people would respond to different issues accurately [1].

B) Stance Detection

Stance detection is an essential part of this ability. Stance detection is an interesting and central task in NLP, which refers to the automated determination of the viewpoint of an author about respect to some subject or aim.

TABLE 1: RESULTS OF STANCE DETECTION USING DIFFERENT MODELS ON THE SEMEVAL2016 DATASET [1]

ID	S-CNN	TAN	TOP(MITRE)	Independent Encoding-CNN	Condition Encoding-CNN
E1	0.6334	0.5933	0.6147	0.4990	0.5593
E2	0.5269	0.5359	0.4163	0.5625	0.5935
E3	0.5133	0.5577	0.6209	0.4566	0.5515

E4	0.6441	0.6538	0.5767	0.5012	0.6228
E5	0.6109	0.6372	0.5728	0.5953	0.6850
Total	0.6733	0.6879	0.6782	0.5691	0.6601

TABLE 2: RESULTS OF STANCE DETECTION USING DIFFERENT MODELS ON THE NLPCC2016 DATASET [1]

ID	BILSTM-CNN	TAN	TOP(RUC_MMC)	Independent Encoding-CNN	Condition Encoding-CNN
C1	0.4494	0.7750	0.7730	0.5255	0.5789
C2	0.5281	0.5933	0.5780	0.6642	0.6831
C3	0.5126	0.5919	0.5814	0.5643	0.6221
C4	0.7553	0.6500	0.6209	0.6462	0.6789
C5	0.4355	0.7238	0.7652	0.6078	0.6229
Total	0.5656	0.7288	0.7106	0.6054	0.6229

This technique involves a detailed analysis of a written text; there can be also additional investigation in author’s activities, for example with his contributions to various debates or communication which takes place within the network of society. What makes stance detection interesting is that sometimes there could be no specific target named in the text and expression of one’s position may show up differently ranging from openly expressed to hidden undertones. Actually, numerous practical tasks are focused on estimating a stance rather than simply evaluating sentiment of the text [2].

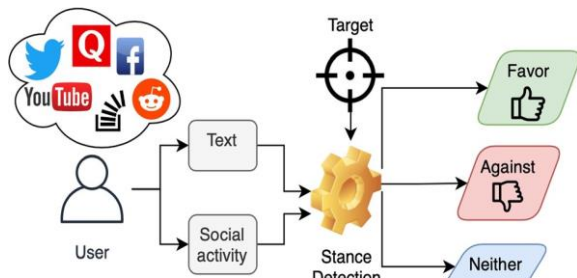


Figure 1: General Representation of Stance Detection Model

This has been caused by the increased relevance of mining attitude information from textual databases that determine the location of a particular field in large corpus. Today’s artificial intelligence has undergone tremendous change in terms of us interaction with it in the current landscape of artificial intelligence. Today’s AI systems have become not only tools but intelligent associates that can grasp and react to the human thoughts and feelings [3].

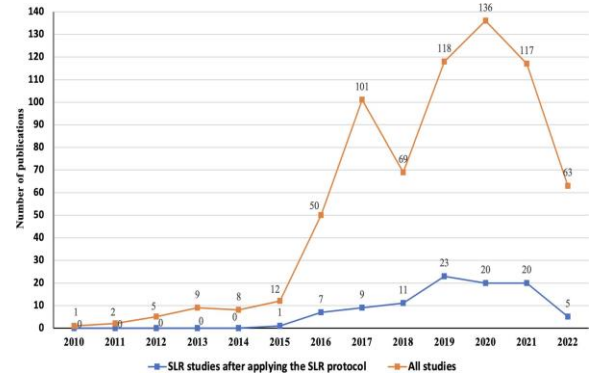


Figure 2: Number of Stance Detection Studies between 2010 and 2022 [3]

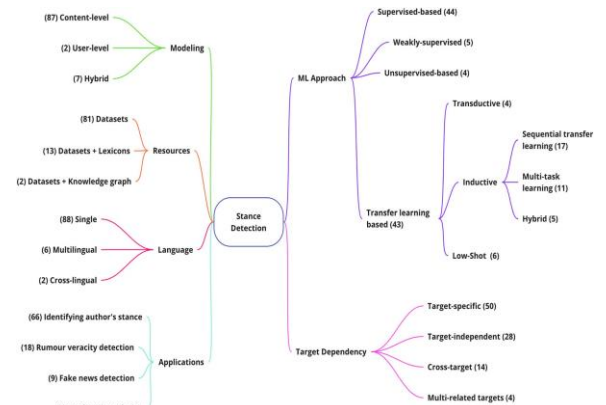


Figure 3: Proposed Taxonomy of Stance Detection [3]

C) *Stance Detection Challenges in Social Media*

The advancements in this field have led to a point where AI can predict much beyond routine functions and now extends to understanding the human behavior and psyche. In this regard, stance detection comes forth as the first step in the realm of NLP leading towards integrating human expressions with the potentiality for AI.

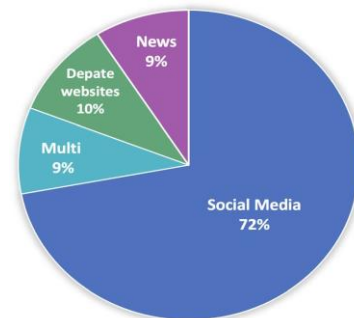


Figure 4: Distribution of the data source platforms for selected Stance Detection studies between 2010-2022 [3]

It entails decoding subtle as well as overt indications within textual data to establish what an author stands for or against. Adding further value to this task by incorporating social media data,

where the complexities of human interactions present numerous additional dimensions [4].

TABLE 3: Dataset Statistics for CoVaxNet and COVID-19-Stance [4]

Dataset	Target	Source	Labels	Samples
CoVaxNet	COVID-19 Vaccination	Twitter	Pro, Anti	1,831,220
COVID-19-Stance	Face masks, Fauci, School closures, Stay a home order	Twitter	Favor, Against, None	7,122

In today’s digital landscape, where people express their thoughts and debate them openly, finding out what authors really mean is both challenging and a chance. Stance detection is a captivating phenomenon for this reason. To communicate their viewpoints, authors apply varying levels of linguistic devices, including explicit statements for asserting agreement or disagree. The same applies to subtle linguistic features that call for detailed analysis. Such complexities are mirrored in human communication where various aspects like context, tone, and **cross-cultural** understanding shape how people express their attitudes on any issue of concern [5].

Attention Visualization Examples	Target	Prediction
من أسباب تمكن المرأة للعزوة أن تستفيد الدولة الرأسمالية من الضرائب والأجور التي تدفعها المرأة One of the alleged reasons for empowering women is that capitalist states benefit from the taxes and wages paid by women	Women Empowerment	Against
أنا أخذه ليش لأجبره إذا. أما تطعيم أو لا تخي عملك؟؟ هنا هو منطق. أنا راضي بموت بكورونا لكن ممنون من الخوف بعدن خلاص اللي أخذ بالتوقيع لك لأنه كنا نسوي مثلك I don't want to take it, why this compulsion? Either vaccination or no work?? This isn't logical. I'm willing to die from covid, but I won't die out of fear. Anyway, good luck to those who take it, we don't all have to do the same	COVID-19 Vaccine	Against
التحول الإلكتروني فيه توفيق منه لكل شيء من إنسان أو وقود أو زحمة طرق أو عاجز ولله الحمد لا تعد ولا تحصى Digital transformation brings value in everything, be it for humans, fuel, traffic congestion, or the disabled, and the benefits are countless and immeasurable.	Digital Transformation	Favor

Figure 5: Visualization of attention scores from SMTL-HW model on testing examples of Mawrif Dataset, along with their target and correct predictions, Darker colored words are more significant to the model’s prediction [5]

D) Importance of Stance Detection

Stance detection has importance extending beyond the academic sphere, into areas of practical application outside the academy. When it comes to rapidly spreading and broad information dissemination in modern times, automatic evaluation of public opinions, tracking changes in social attitudes, and identifying persuading narrative is incredibly significant [6].



Figure 6: Fox Disney Stock vs Stance from 2017-18

As stance detection applies to a wide array of fields such as politics, marketing, journalism, or public opinion research, it plays a key role in making decisions and forming strategies with respect to the position taken by individuals or groups [7]. For instance, in this age of proliferation of social media and online forms, the demand for automated tools that can organize this voluminous textual data into meaningful information is enormous. Stance detection is not just research, but rather a reality in this online world. Stance Detection – New Challenges Presented by AI’s Evolution and Deep Learning. Indeed, traditional machine learning models are useful, but deep learning models have become a powerful tool for both researcher’s and practitioner’s armory [8]. Such models with capability to extract complex and dynamic patterns from data provide better stance detection accuracy and adaptability.

TABLE 4: Proposed SP-MT (Multitask stance detection and sentiment analysis) model in [9]

Model	Accuracy	F1-Score
Proposed SP-MT model in [9]	93.95	90.24
LR (Argyris et al. 2021)	81.48	81.00
ESD (Vychezhzhanin and Kotelnikov 2021)	89.65	85.11
HAN (Wang et al. 2020)	89.47	86.00

AT-JSS-LEX (Li and Caragea 2019)	88.02	84.01
MNB (Kabaghe and Qin 2019)	85.44	78.08
DNN (Chen, Zou, and Zhao 2019)	84.61	76.23
SVM-ngram (Sobhani, Mohammad, and Kiritchenko 2016)	85.55	66.33

Automated stance detection provides understanding of human perspective at large scale during the era of Information overload where there is plenty of text data [9]. Knowledge based systems (KBS) also have immense potential in fields such as content recommendation, social media analyzer, political discourse analysis, and many others. The objective of KBS is to utilize deep learning models to overcome the problems facing stance detection; therefore, the development of an effective and dependable stance prediction system [10].

TABLE 5: Proposed SP-MT (Multitask stance detection and sentiment analysis) model in [10]

Paradigms (KBS)	Knowledge	Sizes	Reasoning	Performance
WS-BERT	External	340M	Weak	74.5
CoT (Chain of Thought)	Internal	175B	Strong	68.9
LoT (Ladder of Thought)	External	780M	Strong	79.2

E) Past Work in Harassment Detection

In these efforts, we try to add to the progress of AI and NLP encouraging increased intelligence of conversations of human with machine. Literature shows us that there has been work done upon detecting harassment in social media. There has been a discussion on how to address the problem of people isolating themselves and not reporting harassment. A method has also been proposed where we can easily mark and block harassment text. This has all been done by introducing a method where new preprocessing steps are presented to identify phrases referring to a person. A pattern-based approach has been developed to identify online harassment. ML models used in the research can classify harassment by detecting patterns [11]. CNN, LSTM and Naïve Bayes classifiers have been used to detect Online Harassment. There was also the use of SVM, KNN and MLP have been used to identify emotion

intensity and to type in distinct categories of harassment.[12]

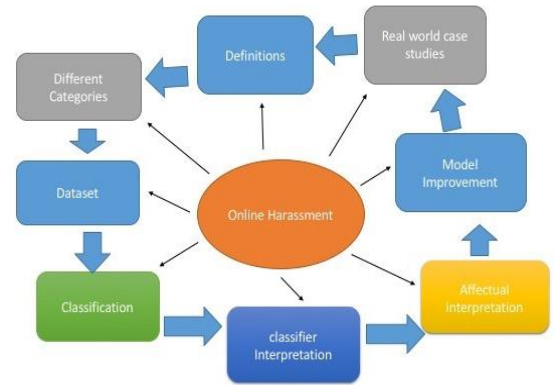


Figure 7: Online Harassment Detection Methodology

The topics are modelled, and documents are classified using Non-Negative Matrix Factorization which is a matrix factorization method in ML. The sources for such studies are Wikipedia, Facebook, and Twitter. Even though harassment can be toward both female and male genders, it is comparatively high for female genders [13].

F) Indirect Harassment or Harassment Stance

No literature has been found where studies have been conducted on detecting harassment stance. There have been studies about detecting direct harassment, but none related to the former. There is no use of Machine Learning or Deep Learning methods for detecting Harassment Stance.

II. METHODOLOGY

The general methodology summed up in the literature is as follows.

A) Acquiring Datasets

First gather the suitable datasets of social media texts. Review certain datasets on Kaggle and pick one which is most suitable for our need.

B) Data Compilation

If the dataset is not suitable, Add more paraphrases and sentences from other datasets like MRPC (Microsoft Research Paraphrase Corpus) and COLA (Corpus of Linguistic Acceptability).

C) Data Pre-Processing

Preprocess the data for our models. Clean the data. We pass it through Word2Vec which gives us the vector representation of the model thus capturing information about the meaning of the word.

D) Data Splitting

Split the data into two parts in the ratio of 7:3. 70 percent of the part is used for training purposes while the rest is for testing purposes.

E) Model Research

Extensively skim through other literature to learn about the deep learning models and which model best serves the purpose of our research.

F) Model Selection

Pick the most suitable model for our research. Here we go with BERT (Bi Encoder Representation through Transformers). BERT has been performing exceptionally in Stance and Sentiment detection tasks. If no model is selected, we go back to the model research step.

G) Model Architecture & Neural Layers Modification

Tune the parameters and modify the model to detect harassment referring to a particular subject. We adjust the neural layers in the model architecture.

H) Data Input

Input our processed data into the model for training purposes.

I) Model Training

The model trains upon the data. See for overfitting and other loss factors and adjust the model to best performance conditions.

J) Output Accuracy

At the end of the model training, acquire the accuracy on training data.

K) Accepted Accuracy

If the accuracy is in the accepted threshold, move on to the next step otherwise go back further steps and even model research and selection steps if the accuracy is still not improving.

L) Testing Data

Model training with training data is followed by model testing by running the model on test data. The accuracy is recorded in the end.

M) Test Success

If the test gives accuracy as per the accuracy achieved through training data, move on to the next step otherwise skim back directly to the model research and selection step.

N) Output Results

Record the output results. Do all the above steps for LSTM and then compare both models.

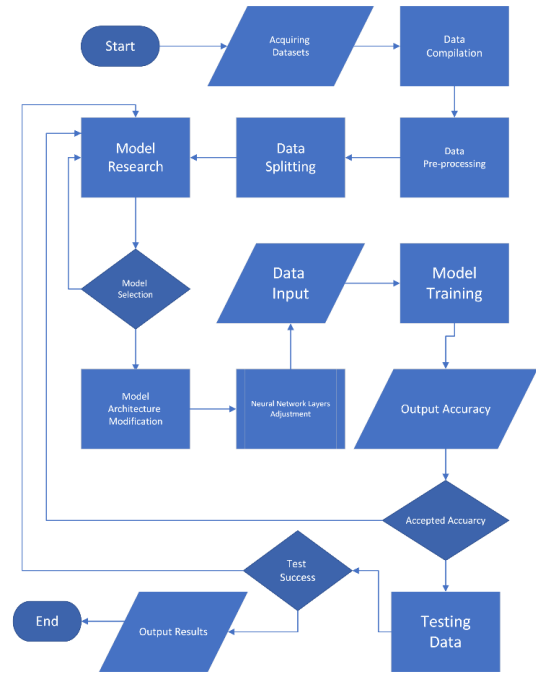


Figure 8: Methodology Flowchart

III. COMPARISON RESULTS AND DISCUSSION

Table 1 and 2 show that datasets can also affect the results and performance of the models. Same models can have different results for different datasets. With the advancement in ML, DL techniques, we have seen a high growth in research surrounding stance detection but compared to all the years, the covid years 2019 and 202 have been the peak years for this research as shown in figure 2. A comprehensive taxonomy of Figure 2 studies is shown in figure 3 representing the scale of work done in Stance Detection. In this taxonomy, 72 percent of data is sourced from social media which shows the reliability and authenticity of Social Media texts as a source for Stance Detection tasks. The two important and widely used datasets related to COVID and stance detection are described in terms of their targets, sources, labels, and samples in table 3. A cross cultural visualization is shown in figure 5 where we can see the use of attention-based mechanism in other languages as well. Stance is affected by public opinion, and this can in turn affect the business performance of the affected stocks (companies) as shown in figure 6. Table 4 gives us a multi-task stance detection and sentiment analysis model results showing how hybrid models can perform stance detection tasks. The sizes given in KSB systems in table 5 represent the variance of accuracy related to models and sizes. Different models can handle varied sizes of data. As CoT has 70% accuracy on 175B whereas WS-BERT has the same accuracy on 340M. BERT performs good at large datasets as compared to extremely large or small ones, so BERT is a favorable option for Harassment stance detection. BERT is enhanced to detect indirect harassment detection. It can detect whether someone's stance is favoring or opposing harassment.

IV. CONCLUSION

This research represents a significant step forward in the academic corpus of Stance Detection. Further contributing to the research about detecting, stopping, and controlling harassment in daily life, professional workspace culture.

ACKNOWLEDGMENTS

This work was supported by Secured IOT Devices Lab, UET Peshawar

REFERENCES

- [1] X. Zhang, C. Liu, Z. Gao, and Y. Jiang, "Text stance detection based on deep learning," in 2021 IEEE International Conference on Progress in Informatics and Computing (PIC), 2021. B. Rieder, *Engines of Order: A Mechanology of Algorithmic Techniques*. Amsterdam, Netherlands: Amsterdam Univ. Press, 2020.
- [2] V. Yantseva and K. Kucher, "Stance classification of social media texts for underresourced scenarios in social sciences," *Data (Basel)*, vol. 7, no. 11, p. 159, 2022.
- [3] Alturayef, H. Luqman, and M. Ahmed, "A systematic review of machine learning techniques for stance detection and its applications," *Neural Comput. Appl.*, vol. 35, no. 7, pp. 5113–5144, 2023.
- [4] N. Kim, D. Mosallanezhad, L. Cheng, M. V. Mancenido, and H. Liu, "STANCE-C3: Domain-adaptive cross-target stance detection via contrastive learning and counterfactual generation," arXiv [cs.CL], 2023.
- [5] N. Alturayef, H. Luqman, and M. Ahmed, "Enhancing stance detection through sequential weighted multi-task learning," *Research Square*, 2023.
- [6] C. Leyden, B. Chen, A. Chauhan, S. Au-Yeung, and M. Colak, "Predicting Merger and acquisition deal completion and stock movement with stance detection," Preprints, 2023.
- [7] S. Chuang, "Tutorials on stance detection using pre-trained language models: Finetuning BERT and prompting large language models," arXiv [cs.CL], 2023.
- [8] M. L. Carnot et al., "On Stance Detection in Image Retrieval for Argumentation," in Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023.
- [9] A. Upadhyaya, M. Fisichella, and W. Nejdl, "Towards sentiment and Temporal Aided Stance Detection of climate change tweets," *Inf. Process. Manag.*, vol. 60, no. 4, p. 103325, 2023.
- [10] . Hu, M. Yan, J. T. Zhou, I. W. Tsang, W. H. Chong, and Y. K. Yap, "Ladder-ofThought: Using knowledge as steps to elevate stance detection," arXiv [cs.CL], 2023.
- [11] Bretschneider, U., Wöhner, T. and Peters, R. (2014) " Detecting online harassment in social networks", 2014 Thirty Fifth International Conference on Information Systems
- [12] Sharifirad, S. (2019) Nlp And Machine Learning Techniques To Detect Online Harassment On Social Networking Platforms [Preprint].
- [13] Sharifirad, S., Jacov, A., & Matwin, S. (2019, July). Understanding Different Types of Online Harassment Using Convolutional Filters, Widening NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)* (Vol. 1, pp. 1644-1656). Association for Computational Linguistics.
- [14] M. Sharifirad, B. Jafarapour, and S. Matwin, "Boosting Text Classification Performance on Sexist Tweets by Text Augmentation and Text Generation Using a Combination of Knowledge Graphs," in Proceedings of the ACL Workshop on Abuse Language Detection (ALW2), EMNLP 2018, pp. 1-6.
- [15] Sharifirad, S., & Matwin, S. (2018, April). Classification of Different Types of Sexist Languages on Twitter and the Gender Footprint on Each of the Classes. In Proceedings of the Conference on Computational Linguistics and Intelligent Processing (CiCLing)
- [16] Alyahya, S., Khan, W. U., Ahmed, S., Marwat, S. N. K., & Habib, S. (2022). Cyber Secure Framework for Smart Agriculture: Robust and Tamper-Resistant Authentication Scheme for IoT Devices. *Electronics*, 11(6), 963.
- [17] A. Sarwar, S. Hasan, W. U. Khan, S. Ahmed and S. N. K. Marwat, "Design of an Advance Intrusion Detection System for IoT Networks," 2022 2nd International Conference on Artificial Intelligence (ICAI), Islamabad, Pakistan, 2022, pp. 46-51, doi: 10.1109/ICAI5435.2022.9773747.
- [18] Sarwar, A., Alnajim, A. M., Marwat, S. N. K., Ahmed, S., Alyahya, S., & Khan, W. U. (2022). Enhanced anomaly detection system for IoT based on improved dynamic SBPSO. *Sensors*, 22(13), 4926.
- [19] Sarwar, A., Alnajim, A. M., Marwat, S. N. K., Ahmed, S., Alyahya, S., & Khan, W. U. (2022). Enhanced anomaly detection system for IoT based on improved dynamic SBPSO. *Sensors*, 22(13), 4926.

Delineation of Crop Fields in Satellite Imagery, using SAM.

Mahmood Ali Khan
dept. Of Computer Systems Engineering
University of Engineering and Technology
 Peshawar
 Peshawar, Pakistan
 ORCID: 0009-0008-9041-6793

Dr. Waleed
dept. Of Computer Systems Engineering
University of Engineering and Technology
 Peshawar
 Peshawar, Pakistan
khanwaleed247@uetpeshawar.edu.pk

Hadia Khan
dept. Of Computer Systems Engineering
University of Engineering and Technology Peshawar
 Peshawar, Pakistan
khanhadia@gmail.com

Muhammad Hasanat
dept. Of Computer Systems Engineering
University of Engineering and Technology Peshawar
 Peshawar, Pakistan
imhasanat@gmail.com

Dr. Nasru Minallah
dept. Of Computer Systems Engineering
University of Engineering and Technology Peshawar
 Peshawar, Pakistan
minallah.nasru@gmail.com

Abstract— This study presents an automated approach for crop field parcel delineation using the advanced Segment Anything Model (SAM), a deep learning model designed for semantic segmentation. SAM's key strengths stem from its effective utilization of high-resolution satellite imagery, consistently delivering superior performance compared to conventional manual or supervised classification methods. Extensively evaluated on a diverse "SA-1B" Dataset, which features over one billion high-quality segmentation masks from a diverse array of 11 million images. By leveraging SAM's extensive pre-training, we adeptly navigated the complexities of agricultural segmentation without additional dataset-specific training. Our results validate the efficacy of SAM's pre-trained model; when applied to high-resolution satellite imagery for multiclass segmentation tasks, it achieved precision accuracy rates as high as 97.8% and true positive rates above 91.2%, with false-positive rates kept below 4.4%. The adoption of our method holds promise in reducing manual labor, making it an asset for precision agriculture applications, such as crop monitoring and yield estimation. Furthermore, its applicability in small-scale agricultural tasks can empower resource-constrained farmers to optimize crop management and enhance their livelihood. This method also extends to efficient field management practices by enabling precise land parcel delineation, leading to resource optimization and waste-reduction. The significance of SAM lies in its potential to transform crop field parcel delineation, benefiting both large-scale agricultural operations and small-scale farmers with dependable, time-saving, and cost-effective solutions. Automation in this critical task not only boosts agricultural productivity and sustainability but also empowers farmers of varying scales to leverage precision agriculture's advantages, ultimately contributing to global food security and economic well-being.

Keywords: SAM; Delineation; Small Holder Farms; Segmentation.

INTRODUCTION

The process of image segmentation plays a pivotal role in computer vision, offering extensive implications across various fields, with agriculture being a primary beneficiary [1]. This technique involves the division of digital images into

distinct segments, simplifying or altering their representation to facilitate meaningful analysis. In agriculture, proficient image segmentation holds paramount importance as it enables accurate identification and differentiation of crucial components such as crops, weeds, pests, and diseases. Such distinctions are indispensable for precision agriculture, which seeks to optimize field-level management concerning crop science and environmental preservation [2].

However, agricultural environments pose unique challenges to image segmentation due to their inherent complexity. Factors like fluctuating lighting conditions, diverse growth stages of crops, overlapping vegetation, and the presence of soil and water significantly complicate the segmentation process [3]. Traditional segmentation methods, including thresholding, edge-based techniques, and region-based approaches, often fall short in such intricate scenarios. These methods, reliant on predefined features and thresholds, lack the flexibility and adaptability essential for diverse and dynamic agricultural settings [4].

Edge detection emerges as a crucial aspect in agricultural image analysis, facilitating precise contour identification of agricultural products and laying the groundwork for subsequent tasks such as product classification, robotic picking, and quality assessment [5]. Despite the accuracy of methods like John Canny's edge detection operator, developed in 1986, challenges persist, particularly in coping with lighting variations and potential loss of edge information due to over-smoothing from fixed parameter settings [6]. Alternative operators based on first-order and second-order differentials or frequency domain operations such as wavelet-based detectors offer promising avenues for improvement [7].

Nevertheless, the advent of deep learning models, notably Convolutional Neural Networks (CNNs), has revolutionized image segmentation, exhibiting remarkable success in handling agricultural image intricacies by directly learning complex features from data [8]. These models excel in various agricultural tasks, including disease detection, weed control, and yield estimation [9]. However, their reliance on extensive labeled datasets presents a significant hurdle, given the time-consuming and expensive nature of data acquisition, especially in agriculture with its diverse conditions across geographies and seasons [10].

In light of these challenges, the Segment Anything Model (SAM) [11] emerges as a groundbreaking solution,

transcending limitations of traditional and deep learning-based segmentation methods. SAM represents a versatile model capable of both interactive and automatic segmentation, a feature particularly advantageous in agriculture's diverse segmentation needs. Trained on an extensive dataset of over 1 billion masks, SAM exhibits unparalleled generalization abilities, enabling effective handling of new objects and images not encountered during training [12].

This paper presents a comprehensive analysis of SAM's application in agricultural field segmentation. We explore SAM's architecture, highlighting its distinctive features that set it apart in the realm of image segmentation. Furthermore, we delve into SAM's training methodology, which underpins its robust generalization capabilities. Rigorous testing on various agricultural segmentation benchmarks is conducted to critically evaluate SAM's performance against existing methods. Additionally, we provide case studies showcasing SAM's effectiveness in addressing real-world agricultural challenges such as crop health monitoring, pest detection, and automated harvesting [13].

Through this study, we aim to underscore SAM's transformative potential in the agricultural sector. We envision SAM not only as a tool for enhancing crop management but also as a catalyst for innovation in sustainable farming practices. By elucidating SAM's capabilities and applications, this work aims to inspire researchers, technologists, and practitioners in agriculture to explore new avenues leveraging advanced image segmentation technology.

LITERATURE REVIEW

In recent years, there has been significant research interest in the realm of automatic field delineation, with notable advancements observed across various applications and environments. Precision management of forest resources has been a focal point, as evidenced by studies such as that of [14], who explored the utilization of deep learning models, particularly the Mask R-CNN technique, for individual tree delineation in planted forests. Their findings demonstrated the model's effectiveness, especially in multi-band imagery, marking a significant leap in forest resource management. Similar methodologies have been applied in different types of forests, as highlighted by [15], further affirming the versatility of deep learning in forestry applications.

In unplanned urban areas, challenges in characterizing high-density settlements have been addressed by researchers like [16]. By employing the U-Net model, their study achieved remarkable accuracy in segmenting and classifying individual buildings, even in complex landscapes. This success has opened avenues for urban planning and development, corroborated by related studies such as that of [17], which underscored the potential of deep learning models in urban segmentation tasks.

Expanding the application of field delineation beyond urban landscapes, [18] tackled the intricate monitoring of

mangrove forests. Their innovative approach, combining deep learning algorithms with marker-controlled watershed segmentation, effectively differentiated individual tree crowns across diverse mangrove ecosystems, outperforming previous techniques. This effectiveness extends to coastal ecosystem monitoring, as demonstrated by [19], emphasizing the importance of precise delineation for environmental conservation.

The scope of automatic field delineation extends to infrastructure planning and disaster mitigation, as evidenced by [20], who introduced Unet-AP, a CNN architecture for precise building footprint extraction from high-resolution satellite imagery. Their design showcased superior performance compared to established architectures, particularly across various urban settlement classes. This aligns with the findings of [21], recognizing the critical role of accurate building delineation in disaster management and urban planning.

In the geological domain, automated techniques for bedrock identification and mapping have shown promise, as demonstrated by [22], who utilized DEM terrain derivatives to achieve robust mapping results. Similarly, [23] highlighted the effectiveness of automated techniques in geological mapping.

In agriculture, remote sensing-based identification of oil palm trees by [24] stands out, achieving remarkable accuracy with two-stage CNNs trained on extensive satellite imagery datasets. This advancement is echoed in crop detection and management studies by [25], emphasizing the importance of deep learning in agricultural applications.

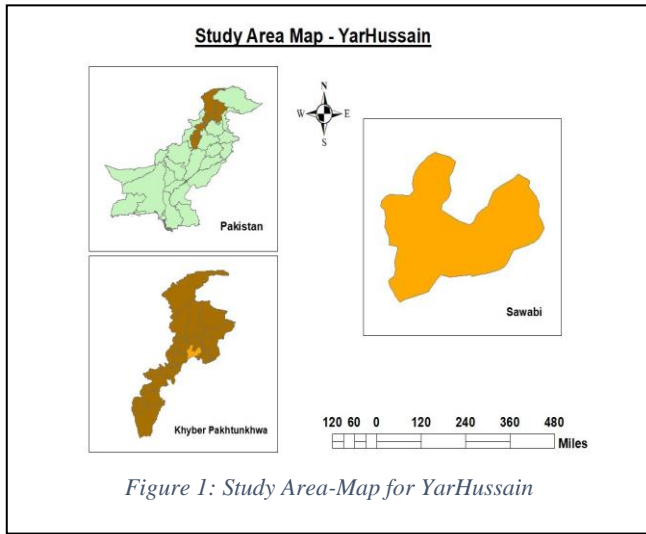
The urgency of swift disaster response was addressed by [26], who proposed a multi-scale segmentation and scene change detection approach for early recognition of structural damage in disaster-stricken areas. Their method, based on deep learning models, exhibited high accuracy in detecting collapsed structures.

In the specific context of small-scale farms, studies by [27] and [28] focused on delineating agricultural land and fields using deep fully convolutional networks and advanced segmentation methods, surpassing previous approaches in accuracy. This aligns with the work of Small Farm Innovations [29], demonstrating the potential of these technologies in enhancing small-scale farming operations.

In the broader context of improving agricultural productivity, the synthesis of these studies underscores the pivotal role of automatic field delineation, as highlighted by [30]. These diverse studies collectively illustrate the evolving landscape of automatic farm boundary delineation, offering valuable insights for sustainable land management and resource allocation.

I. METHODOLOGY

A. Study Area



The area we have targeted is Yar Hussain Swabi ($34^{\circ}10'19''$ N and $72^{\circ}16'19''$ E), Khyber Pakhtunkhwa Pakistan, situated at 82 kilometers (51 miles) from the province capital of Peshawar and 23 kilometers (14 miles) from the district capital of Mardan. Yar Hussain has a temperate climate with pleasant summers and moderate winters. The monsoon season brings most of the region's rainfall. The region's rich soil and pleasant climate have produced the perfect conditions for a flourishing agriculture industry. Yar Hussain Swabi's agricultural impact is closely linked to its socioeconomic profile. Most of the population finds work in farming, which is the backbone of the local economy. Wheat, maize, sugarcane, and an assortment of fruits and vegetables are the main crops grown in the area.

B. Data Collection

The area we have targeted is Yar Hussain Swabi ($34^{\circ}10'19''$ N and $72^{\circ}16'19''$ E), Khyber Pakhtunkhwa Pakistan, situated at 82 kilometers (51 miles) from the province capital of Peshawar and 23 kilometers (14 miles) from the district capital of Mardan. Yar Hussain has a temperate climate with pleasant summers and moderate winters. The monsoon season brings most of the region's rainfall. The region's rich soil and pleasant climate have produced the perfect conditions for a flourishing agriculture industry. Yar Hussain Swabi's agricultural impact is closely linked to its socioeconomic profile. Most of the population finds work in farming, which is the backbone of the local economy. Wheat, maize, sugarcane, and an assortment of fruits and vegetables are the main crops grown in the area.

C. Preprocessing for SAM

The dataset, which included 1.1 billion masks and 11 million images, was preprocessed to guarantee that it was in a consistent format and of good quality before the Segment Anything Model (SAM) was trained. A standard 256×256

pixel size was applied to all of the photographs in the dataset. By ensuring that every image was the same size, the model was able to identify patterns in the data more quickly. A mean of 0 and a standard deviation of 1 were applied to all the photos in the dataset. This enhanced the convergence and generalization capabilities of the model. The dataset's masks underwent preprocessing to guarantee their accuracy and completeness. This involved patching any holes and eliminating any noise from the masks. To expand the dataset and strengthen the model's resistance to various data kinds, more data was added. This includes rotation, cropping, and random flipping among other techniques.

The preparation procedures were thoughtfully created to guarantee that the data was of the highest caliber and that the model could benefit greatly from it. The model's generalization performance or its ability to function well on fresh data that it had not been trained on was also enhanced by the preparation procedures. That's why we used this model for our purpose of Field Boundaries Segmentation as it does work good for the fresh data as well.

D. SAM for Segment Geospatial

In our study, we introduce several key components to facilitate the segmentation and analysis of geospatial data. First, we developed an open-source Python module named Segment-geospatial (samgeo), which simplifies the segmentation process using the Segment Anything Model. Leveraging well-established Python libraries like leafmap, ipywidgets, rasterio, geopandas, and segment-anything-py, this package provides users with a user-friendly interface for segmenting remote sensing imagery. The generated results can be exported in various formats, including vector and raster data, and users can interactively visualize segmentation outcomes within a Jupyter notebook. This tool aims to address the need for an effective and adaptable geospatial segmentation solution without requiring deep learning model training, thereby filling a gap in the Python ecosystem.

Moreover, we recognize the computational complexity involved in segmenting geo-spatial data using conventional CPUs, rendering it impractical and time-consuming. To overcome this challenge, we employed a 32 GB GPU (NVIDIA RTX A6000) along with 64GB of DDR3 RAM, significantly reducing processing time and enhancing efficiency. Additionally, we discuss the creation of an interactive map facilitated by the leafmap package, which integrates Google Satellite imagery. The map allows users to specify latitude and longitude coordinates, pan, and zoom to select areas of interest, and employ draw tools to delineate polygons or rectangles for further analysis, yielding output

geotiff images. Furthermore, we highlight the significance of Google Earth satellites in providing high-resolution imagery for diverse applications such as educational exploration, environmental monitoring, and urban planning. The advanced capabilities of these satellites, including 3D modeling and real-time data updates, contribute to a comprehensive understanding of Earth's ecosystems and topography.

Moreover, we elucidate the initialization of the SAM class using Vision Transformer models and the masking process, which involves generating binary masks to highlight segmented areas within images. This enables specific analysis or modification of selected regions of interest, with options to save images in various vector formats such as .gpkg and .shp.

Lastly, we address the challenge of eliminating extraneous polygons from segmented images, opting for a Python-based approach to efficiently remove small vegetation parcels not relevant to our analysis. This step significantly improves the accuracy of our results and streamlines the segmentation process.

II. RESULTS

In our study, we utilize an original image obtained from Google Earth, offering a comprehensive aerial view of our study area, providing vital geographical context. Figure 2(a) displays the unaltered perspective of the study area, depicting its topography, land usage patterns, and surrounding features.

Figure 2(b) showcases the masked image generated by SAM, effectively identifying and delineating distinct segments within the agricultural landscape, setting the stage for further analyses. This masked image highlights SAM's capability to isolate regions of interest through initial segmentation.

In Figure 2(c), segmented polygons representing agricultural fields obtained from SAM's segmentation are presented. This figure includes unwanted objects such as urban areas, trees, and other non-agricultural features, termed as "noise" for our study's purpose. It demonstrates the segmentation process outcome, combining both the noise (depicted in red) and the filtered image (depicted in green), emphasizing SAM's effectiveness in producing detailed and accurate agricultural field segmentation.

To refine the segmentation results further for precision, Figure 2(d) illustrates the output after implementing noise removal techniques as discussed in above section. By systematically eliminating extraneous details and artifacts, SAM's segmentation output becomes more polished and precise, providing a solid foundation for subsequent analyses.

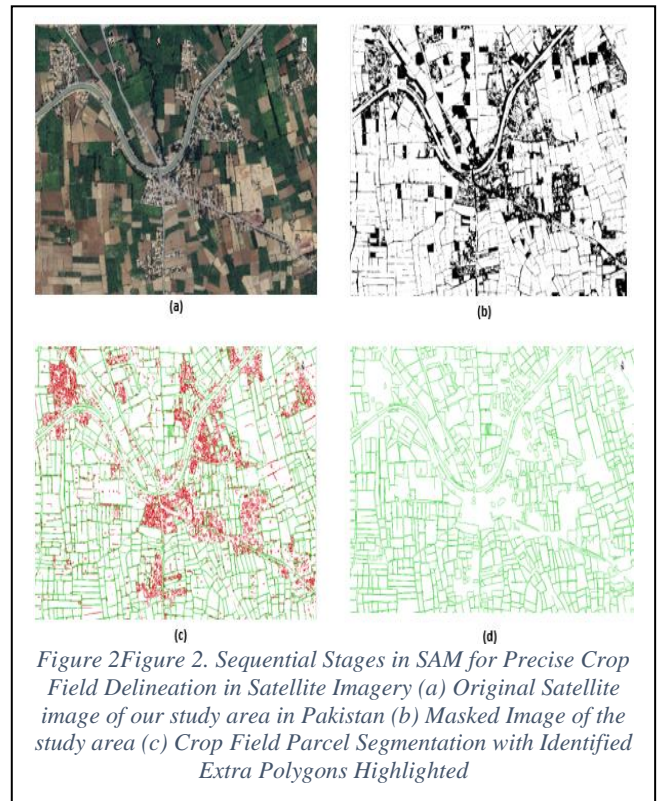


Figure 2. Sequential Stages in SAM for Precise Crop Field Delineation in Satellite Imagery (a) Original Satellite image of our study area in Pakistan (b) Masked Image of the study area (c) Crop Field Parcel Segmentation with Identified Extra Polygons Highlighted

III. WINDOWS FOR OPPURTUNITY

Employing SAM for crop field delineation holds transformative potential for crop insurance operations, particularly in regions with established insurance schemes. SAM's precision enables insurers to assess risks more accurately, adjusting insurance premiums according to actual field risks, thereby enhancing precision risk assessment. Moreover, SAM's accuracy provides credible field information, reducing fraudulent claims and fostering trust in insurance schemes. Beyond insurance, SAM's precise delineation capabilities offer valuable insights for land management and preservation, allowing policymakers to make informed decisions about land cover changes over time. By tracking transformations such as shifts from agricultural to urban landscapes or reforestation, policymakers can implement sustainable development initiatives while preserving agricultural landscapes. Additionally, SAM generates comprehensive field statistics, benefiting various stakeholders. Farmers gain access to detailed field statistics, enhancing their understanding of field conditions and yields, while agricultural extension services utilize SAM data to provide tailored recommendations and guidance, enhancing outreach and effectiveness. Governments and agricultural organizations leverage SAM's aggregated data for policymaking, resource allocation, and trend tracking in the farming industry, gaining insights at local, regional, and national levels.

IV. CONCLUSION

REFERENCES

In summary, our research underscores the transformative potential of utilizing the Segment Anything Model (SAM) for crop field delineation across multiple sectors. SAM's precision and accuracy play a pivotal role in revolutionizing crop insurance operations, particularly in regions with established insurance schemes. By providing insurers with precise field information, SAM enables more accurate risk assessment, allowing for the adjustment of insurance premiums based on actual field risks. This not only enhances the precision of risk assessment but also aids in combating fraudulent claims by providing credible field data, thus fostering trust and stability within insurance schemes.

Furthermore, SAM's capabilities extend beyond the realm of insurance, offering valuable insights for land cover change monitoring and informed decision-making in land management and preservation. Through SAM's ability to precisely delineate crop fields, policymakers gain valuable information regarding shifts in land cover, such as the transition from agricultural to urban landscapes or reforestation efforts. This enables policymakers to make informed decisions about land use, ensuring sustainable development while preserving vital agricultural landscapes.

Moreover, SAM generates comprehensive field statistics, benefiting various stakeholders across the agricultural sector. Farmers gain access to detailed field information, enabling them to make informed decisions about crop management practices and optimize yields. Agricultural extension services leverage SAM data to provide tailored recommendations and guidance to farmers, enhancing their outreach and effectiveness. Additionally, governments and agricultural organizations utilize SAM's aggregated data for policymaking, resource allocation, and trend tracking in the farming industry. This allows stakeholders to gain insights at local, regional, and national levels, facilitating evidence-based decision-making and contributing to the overall sustainability and resilience of the agricultural sector.

In essence, our research demonstrates the immense potential of SAM in transforming crop field delineation practices and enhancing decision-making processes across various domains within the agricultural sector. By leveraging SAM's precision and accuracy, stakeholders can make informed decisions, enhance efficiency, and foster trust in agricultural practices, ultimately contributing to sustainable development and livelihoods in agricultural communities.

ACKNOWLEDGMENTS

I want to extend my appreciation to everyone who contributed to this research project. Your support, guidance, and encouragement were instrumental in bringing this paper to fruition. Special thanks to those who provided valuable insights and assistance along the way. This endeavor has been enriched by the collective efforts of all involved. Thank you for being part of this journey.

- [1] Y. Yu *et al.*, "Techniques and Challenges of Image Segmentation: A Review," *Electronics*, vol. 12, no. 5, p. 1199, Mar. 2023, doi: 10.3390/electronics12051199.
- [2] R. Finger, S. M. Swinton, N. El Benni, and A. Walter, "Precision Farming at the Nexus of Agricultural Production and the Environment," *Annu. Rev. Resour. Econ.*, vol. 11, no. 1, pp. 313–335, Oct. 2019, doi: 10.1146/annurev-resource-100518-093929.
- [3] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis, "Machine Learning in Agriculture: A Comprehensive Updated Review," *Sensors*, vol. 21, no. 11, p. 3758, May 2021, doi: 10.3390/s21113758.
- [4] C. Zheng, A. Abd-Elrahman, and V. Whitaker, "Remote Sensing and Machine Learning in Crop Phenotyping and Management, with an Emphasis on Applications in Strawberry Farming," *Remote Sens.*, vol. 13, no. 3, p. 531, Feb. 2021, doi: 10.3390/rs13030531.
- [5] X. Yu, Z. Wang, Y. Wang, and C. Zhang, "Edge Detection of Agricultural Products Based on Morphologically Improved Canny Algorithm," *Math. Probl. Eng.*, vol. 2021, pp. 1–10, Jun. 2021, doi: 10.1155/2021/6664970.
- [6] M. Gonzalez-Hidalgo, S. Massanet, A. Mir, and D. Ruiz-Aguilera, "On the Choice of the Pair Conjunction—Implication Into the Fuzzy Morphological Edge Detector," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 4, pp. 872–884, Aug. 2015, doi: 10.1109/TFUZZ.2014.2333060.
- [7] M.-Y. Shih and D.-C. Tseng, "A wavelet-based multiresolution edge detection and tracking," *Image Vis. Comput.*, vol. 23, no. 4, pp. 441–451, Apr. 2005, doi: 10.1016/j.imavis.2004.11.005.
- [8] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," 2015, doi: 10.48550/ARXIV.1511.08458.
- [9] P. Bharman, S. Ahmad Saad, S. Khan, I. Jahan, M. Ray, and M. Biswas, "Deep Learning in Agriculture: A Review," *Asian J. Res. Comput. Sci.*, pp. 28–47, Feb. 2022, doi: 10.9734/ajrcos/2022/v13i230311.
- [10] V. G. Dhanya *et al.*, "Deep learning based computer vision approaches for smart agricultural applications," *Artif. Intell. Agric.*, vol. 6, pp. 211–229, 2022, doi: 10.1016/j.aiaa.2022.09.007.
- [11] L. P. Osco *et al.*, "The Segment Anything Model (SAM) for remote sensing applications: From zero to one shot," *Int. J. Appl. Earth Obs. Geoinformation*, vol. 124, p. 103540, Nov. 2023, doi: 10.1016/j.jag.2023.103540.
- [12] Y. Li, D. Wang, C. Yuan, H. Li, and J. Hu, "Enhancing Agricultural Image Segmentation with an Agricultural Segment Anything Model Adapter," *Sensors*, vol. 23, no. 18, p. 7884, Sep. 2023, doi: 10.3390/s23187884.
- [13] A. Carraro, M. Sozzi, and F. Marinello, "The Segment Anything Model (SAM) for accelerating the smart farming revolution," *Smart Agric. Technol.*, vol. 6, p. 100367, Dec. 2023, doi: 10.1016/j.atech.2023.100367.
- [14] K. Yu *et al.*, "Comparison of Classical Methods and Mask R-CNN for Automatic Tree Detection and Mapping Using UAV Imagery," *Remote Sens.*, vol. 14, no. 2, p. 295, Jan. 2022, doi: 10.3390/rs14020295.
- [15] C. Zhang *et al.*, "Multi-Species Individual Tree Segmentation and Identification Based on Improved Mask R-CNN and UAV Imagery in Mixed Forests," *Remote Sens.*, vol. 14, no. 4, p. 874, Feb. 2022, doi: 10.3390/rs14040874.
- [16] Z. Pan, J. Xu, Y. Guo, Y. Hu, and G. Wang, "Deep Learning Segmentation and Classification for Urban Village Using a Worldview Satellite Image Based on U-Net," *Remote Sens.*, vol. 12, no. 10, p. 1574, May 2020, doi: 10.3390/rs12101574.
- [17] B. Neupane, T. Horanont, and J. Aryal, "Deep Learning-Based Semantic Segmentation of Urban Features in Satellite Images: A Review and Meta-Analysis," *Remote Sens.*, vol. 13, no. 4, p. 808, Feb. 2021, doi: 10.3390/rs13040808.
- [18] G. Lassalle and C. R. De Souza Filho, "Tracking canopy gaps in mangroves remotely using deep learning," *Remote Sens. Ecol. Conserv.*, vol. 8, no. 6, pp. 890–903, Dec. 2022, doi: 10.1002/rse2.289.
- [19] B. Kang and O. Duran Vinent, "The Application of CNN-Based Image Segmentation for Tracking Coastal Erosion and Post-Storm Recovery," *Remote Sens.*, vol. 15, no. 14, p. 3485, Jul. 2023, doi: 10.3390/rs15143485.

- [20] K. Rastogi, P. Bodani, and S. A. Sharma, "Automatic building footprint extraction from very high-resolution imagery using deep learning techniques," *Geocarto Int.*, vol. 37, no. 5, pp. 1501–1513, Mar. 2022, doi: 10.1080/10106049.2020.1778100.
- [21] Z. Xing *et al.*, "Flood vulnerability assessment of urban buildings based on integrating high-resolution remote sensing and street view images," *Sustain. Cities Soc.*, vol. 92, p. 104467, May 2023, doi: 10.1016/j.scs.2023.104467.
- [22] A. J. Ganerød, V. Bakkestuen, M. Calovi, O. Fredin, and J. K. Rød, "Where are the outcrops? Automatic delineation of bedrock from sediments using Deep-Learning techniques," *Appl. Comput. Geosci.*, vol. 18, p. 100119, Jun. 2023, doi: 10.1016/j.acags.2023.100119.
- [23] H. Shang, Y.-G. Shen, S. Li, A.-B. Li, and T. Zhang, "An Automated Mapping Method of 3D Geological Cross-Sections Using 2D Geological Cross-Sections and a DEM," *ISPRS Int. J. Geo-Inf.*, vol. 12, no. 4, p. 147, Mar. 2023, doi: 10.3390/ijgi12040147.
- [24] W. Li, R. Dong, H. Fu, and L. Yu, "Large-Scale Oil Palm Tree Detection from High-Resolution Satellite Images Using Two-Stage Convolutional Neural Networks," *Remote Sens.*, vol. 11, no. 1, p. 11, Dec. 2018, doi: 10.3390/rs11010011.
- [25] M. Albahar, "A Survey on Deep Learning and Its Impact on Agriculture: Challenges and Opportunities," *Agriculture*, vol. 13, no. 3, p. 540, Feb. 2023, doi: 10.3390/agriculture13030540.
- [26] C. Liu, S. M. E. Sepasgozar, Q. Zhang, and L. Ge, "A novel attention-based deep learning method for post-disaster building damage classification," *Expert Syst. Appl.*, vol. 202, p. 117268, Sep. 2022, doi: 10.1016/j.eswa.2022.117268.
- [27] A. Taravat, M. P. Wagner, R. Bonifacio, and D. Petit, "Advanced Fully Convolutional Networks for Agricultural Field Boundary Detection," *Remote Sens.*, vol. 13, no. 4, p. 722, Feb. 2021, doi: 10.3390/rs13040722.
- [28] M. Li, J. Long, A. Stein, and X. Wang, "Using a semantic edge-aware multi-task neural network to delineate agricultural parcels from remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 200, pp. 24–40, Jun. 2023, doi: 10.1016/j.isprsjprs.2023.04.019.
- [29] X. Diao *et al.*, "The Future of Small Farms: Innovations for Inclusive Transformation," in *Science and Innovations for Food Systems Transformation*, J. Von Braun, K. Afsana, L. O. Fresco, and M. H. A. Hassan, Eds., Cham: Springer International Publishing, 2023, pp. 191–205. doi: 10.1007/978-3-031-15703-5_10.
- [30] S. Wang, F. Waldner, and D. B. Lobell, "Unlocking Large-Scale Crop Field Delineation in Smallholder Farming Systems with Transfer Learning and Weak Supervision," *Remote Sens.*, vol. 14, no. 22, p. 5738, Nov. 2022, doi: 10.3390/rs14225738.

Sherpa: Implementing a Hybrid Recommendation System for Next Gen Tourist Experience

Inam Ullah Khan

Department of Computer Systems
Engineering, University of
Engineering and Technology Peshawar,
Peshawar, Pakistan
20pwcse1862@uetpeshawar.edu.pk

Abdurrahman

Department of Computer Systems
Engineering, University of
Engineering and Technology Peshawar,
Peshawar, Pakistan
20pwcse1878@uetpeshawar.edu.pk

Muhammad Awais Khan

Department of Computer Systems
Engineering, University of
Engineering and Technology Peshawar,
Peshawar, Pakistan
20pwcse1871@uetpeshawar.edu.pk

Dr. Madiha Sher

Department of Computer Systems
Engineering, University of
Engineering and Technology Peshawar,
Peshawar, Pakistan
madiha@uetpeshawar.edu.pk

Dr. Yasir Saleem

Department of Computer Systems
Engineering, University of
Engineering and Technology Peshawar,
Peshawar, Pakistan
yasirsaleem@uetpeshawar.edu.pk

Abstract— In the digital era, Sherpa revolutionizes personalized tourism with an AI-driven recommendation system, fostering meaningful connections between travelers and local guides. This study explores Sherpa's integration of collaborative and content-based filtering—specifically, singular value decomposition (SVD) and cosine similarity—to tailor travel experiences uniquely. Our methodology includes a detailed examination of Sherpa's algorithm and its implementation within a cross-platform, MERN Stack-powered backend. We assess the system's efficacy in aligning recommendations with individual user preferences, based on quantitative user feedback and engagement metrics. Initial results demonstrate a significant improvement in personalized experience satisfaction. The paper concludes that Sherpa's innovative approach not only enhances the quality of travel recommendations but also sets a new standard for interactive and adaptive tourism platforms. Through continuous algorithmic refinement, Sherpa is positioned to lead a transformative shift in how travelers explore new destinations, offering not just journeys, but transformative experiences.

Keywords: Artificial Intelligence; Digital Tourism; Collaborative Filtering; Content-Based Filtering; Hybrid Recommender System; Singular Value Decomposition; Cosine Similarity Matrix.

I. INTRODUCTION

Travel isn't just seeing sights anymore. People want deeper experiences when they go places. They don't just want planned schedules and surface interactions. Now, travelers aim to connect with locals. They hope to uncover hidden gems and make lasting memories. The old tourism ways don't satisfy these changing desires. Connectivity and curiosity drive travel today. Travelers yearn for meaningful encounters, not just observing from outside. They aspire to immerse themselves in the cultures they visit.

Sherpa shows a change in travel. It pictures tourists acting as members of host towns, not just visitors. Sherpa aims to make guided tourism different. It connects tourists with local guides, creating a good partnership where exploration is a shared project.

Traditional tourism has limits - guidebooks and online sites offer insights but lack personal touch, real-time flexibility that only locals provide. Sherpa recognizes this limitation. It aims to give visitors and guides more influence by fostering cross-cultural dialogue, offering customized memorable experiences. Sherpa's genesis lies in addressing tourism's shortcomings. It provides a venue where cultures connect, making each trip unique through personalization.

Sherpa uses an AI matching system. This system matches tourists with guides. It looks at what tourists like. It also looks at guides' skills. The AI finds hidden patterns. It uses info from tourists and guides. The goal is great tourist trips. The system matches based on tourist wants.

The goal of offering suggestions suited to each person's particular tastes and interests remains a key issue for the travel industry, despite technological advancements. Sherpa tackles this challenge head-on with an innovative hybrid recommendation system. This approach blends content-based filtering using cosine similarity matrices with a seamless integration of collaborative filtering via singular value decomposition (SVD) models. By combining these methods, Sherpa provides travellers with immersive, personalised experiences that transcend limitations of conventional tourism models.

Simply, Sherpa isn't only a mobile app. It shows a new way to do guided tours. Sherpa means adventure, connecting, and real experiences. It will make travel better for visitors and locals. As Sherpa grows, it will keep changing how we travel. One personal tip at a time.

II. RELATED WORK

The impact of technology on guided tourism is significant. Early examples like TripMate, TripAdvisor, and Airbnb have developed services centered on personalized travel experiences. These platforms offer personalized suggestions, local experiences, and intelligent lodging options through AI and big data. However, they sometimes compromise traveler interests, as they cater to only a limited range of customer preferences.

On the other hand, Sherpa sets itself apart by using a hybrid recommender system that merges the advantages of collaborative filtering and content-based filtering. This combination enables Sherpa to offer personalized recommendations that cater to the distinct preferences of individual travelers. Lots of research had been done to improve the performance of basic Matrix Factorization used in Content based Filtering. In [2], Zhang and colleagues introduced Weighted Non-negative Matrix Factorization (WNMF) as a method to enhance NMF (Non-negative Matrix Factorization). They utilized weights as an indicator matrix to represent the visibility of entries in the matrix R. In [1] Lee et al. gave the idea of Non-negative Matrix Factorization (NMF) to enforce non-negativity in U and V, which was proved to be useful in computer vision fields. In [3], Salakhutdinov et al. showed Probabilistic Matrix Factorization (PMF), that used Gaussian distribution to initialize U and V, and applied logistic function to limit range of predicted to [0,1]. Koren et al. summarized this work in [4] and gave a generic framework for Matrix Factorization. Researchers also managed to incorporate information from other data sources. Zhang et al. used review sentiment analysis to construct virtual ratings for users who have not explicitly on the item [5]. Gu et al. proposed the Graph Weighted Nonnegative Matrix Factorization (GWNMF) [6] to use user/item neighborhood graph to preserve neighborhood information in user/item latent vectors [5, 1]. utilized social network information under the assumption that friends share similar tastes and interests. at the realm of recommender systems, collaborative filtering approaches like matrix factorization and nearest neighbor methods have received a lot of attention and have proven to be effective at recognizing similar people or products and capturing user preferences. Similar to this, item attributes and user profiles have been analyzed by content-based filtering techniques like cosine similarity and natural language processing (NLP) to produce personalized suggestions.

III. METHODOLOGY

Sherpa's development process is methodical and iterative, starting with the design and implementation of its fundamental components and system architecture. The MERN (MongoDB, Express.js, React.js, Node.js) stack powers the program's backend, while a cross-platform mobile application frontend created with React Native powers the application's architecture. This architecture makes sure that various devices and operating systems work together seamlessly, giving users a consistent and easy experience.

The cornerstone of Sherpa's personalized experience is its hybrid recommender system, which combines collaborative filtering with SVD and content-based filtering using cosine similarity matrices.

Content Based Filtering (Cosine Similarity Matrix):

Matrix factorization [4] is one of the most used approaches in recommender systems. Despite of its efficiency, MF still suffers from sparsity problem, i.e., users who rate only a small portion of items could not get proper recommendation, and items with few ratings may not be

recommended well. To cure the Sparsity problem, we have utilized the Novel approach of CosMF from [7].

Formula for Matrix Factorization is given in eq. (1):

$$R_{i,j} = U_i V_j^t \quad (1)$$

Where n is users and m are items. Each element of $R_{i,j}$

represents the rating of user i to item j (trip). U_i denotes the latent preference row vector of user i and V_j denotes latent feature row vector of item j, and both have k dimensions (latent factors).

Cosine Similarity is calculated using eq. (2):

$$R_{i,j} = \frac{U_i V_j^t}{\|U_i\| \|V_j\|} = \cos(U_i, V_j) \quad (2)$$

Collaborative Filtering analyzes a user preferences database to predict additional products or services in which a user might be interested[8]. Collaborative filtering techniques analyze user-item interactions and user preferences to identify similar users or items, while content-based filtering techniques analyze item attributes and user profiles to generate recommendations that match users' preferences.

Singular Value Decomposition is carried out using eq. (3):

$$A = U \Sigma V^t \quad (3)$$

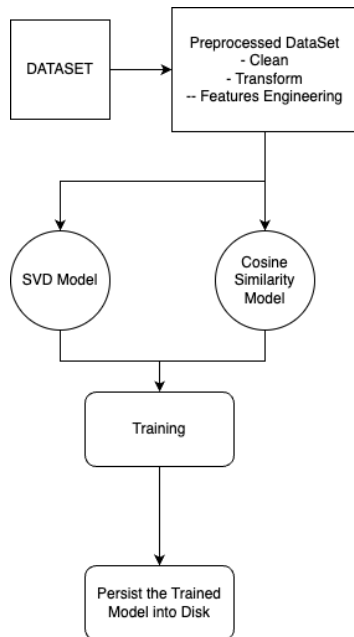
Where A is the input matrix, U are the left singular matrix, Sigma are diagonal/eigenvalues and V is the right singular matrix

By leveraging the strengths of both approaches, Sherpa can deliver highly personalized recommendations that resonate with individual users.

Sherpa's hybrid recommender system is trained on a curated dataset sourced from TripAdvisor, one of the world's largest travel platforms [9]. This dataset contains a diverse range of user interactions, reviews, and item attributes, providing Sherpa with rich and comprehensive data to inform its recommendation process. The dataset is preprocessed and transformed into a suitable format for training the recommender system, ensuring that it captures the underlying patterns and relationships in the data.

Figure 1: Training the Hybrid Recommender system

Figure 1 outlines the methodology adopted to train the hybrid recommendation system using the procured Trip Advisor Dataset. First, we've carried out necessary cleaning and transformations as well as feature extraction on the dataset. This dataset is then divided into training and testing sets by a 70/30 margin. The training set is used to train the SVD and cosine similarity Models of course each model utilizing different features of the dataset, it is subsequently tested and performance metrics are calculated before persisting the model onto storage.



As Figure 2 demonstrates, the hybrid recommendation system is then integrated with the Sherpa Backend by exposing the get recommendation function through a Flask Server Rest API, which the backend calls/hits to generate recommendations for each user on the go.

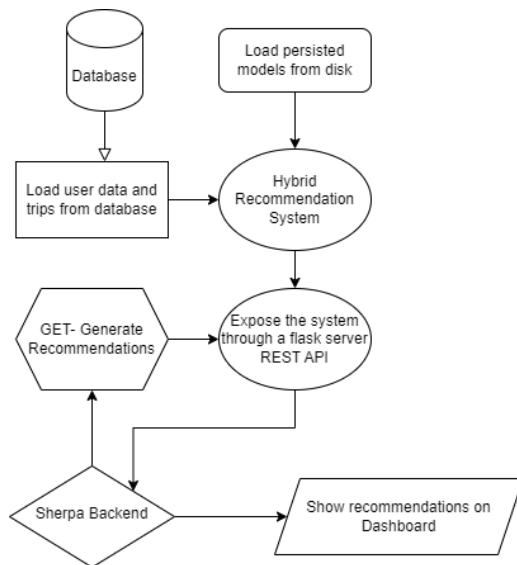


Figure 2: Detailed Process: Integration with Sherpa

IV. RESULTS

Metric	Value
RMSE	0.8
MAE	0.64

Table1: Performance metrics of the Hybrid Recommendation System

As shown in the table the Hybrid Recommendation System managed to achieve a Root Mean Square Error (RMSE) of 0.8 and a Mean Absolute Error (MAE) of 0.64 both of which are indicative of good and relevant recommendations on the part of Collaborative Filtering.

The Content Based filtering has also proved to suggest relevant recommendations based on tags and descriptions relevant to the user’s interactions with the app. The application backend and front end both have been quality tested for user experience, robustness and seamless performance. According to closed alpha test feedback the hybrid recommendation system has streamlined trip finding for the user and the issue of cold starts for new users is almost non-existent, “recommendations were organic similar to the likes of Netflix or YouTube”. We believe these results to be a good omen for the realization of our hybrid recommendation system and app “Sherpa”.

V. DISCUSSION

The heart of Sherpa's personalized experience lies in its hybrid recommender system. The content-based component utilizes natural language processing to analyze tour descriptions and reviews, creating user and item profiles. The collaborative component leverages user-item interactions to understand and predict preferences. The integration of these methods allows Sherpa to deliver highly relevant guide recommendations that resonate with individual user interests. The Hybrid Recommender sherpa engine is trained on a curated dataset, ensuring diversity in user preferences and guide offerings. The iterative development of Sherpa incorporates user feedback at every stage, aligning with Agile methodologies. The repetitive cycle guaranties continuous polishing and adjusting of the application to satisfy user anticipations. The engine behind, fueled by Node.js alongside Express.js, manages data handling and API oversight, whereas the front part, shaped using React, provides an easy-to-use and swift interaction. The RESTful APIs developed in the stage of execution permit effective dialogue between the smartphone app and the server backend.

Our dataset often appeared as a sparse matrix, making it difficult to derive meaningful insights or predictions. To address this, we employed Singular Value Decomposition (SVD), which helped reduce the dimensionality of the dataset and extract underlying patterns. Additionally, we tackled the cold start problem by constructing cosine similarity matrices, enabling us to measure the similarity between users or items based on their features or preferences. By implementing these solutions, we were able to enhance the effectiveness and robustness of our recommendation system, ensuring better recommendations even in the face of sparse data and new user or item entries. The results from Sherpa's deployment indicate a successful integration of AI within a mobile tourism application.

VI. CONCLUSION

Utilizing SVD alongside the Matrix of Cosine Similarity for crafting uniquely tailored journey suggestions from data on TripAdvisor, Sherpa stands as a pioneer in the evolving era of navigated travel. Prospective endeavors could focus on advanced processing of natural language for an enriched comprehension of inclinations, integration of data instantaneously and betterment through feedback from users. Sherpa will become the leader in personalized travel technology by including augmented reality previews, boosting sustainability, expanding suggestions to encompass more travel-related topics, and optimizing scalability. This ground-breaking method promises to revolutionize travel by providing unmatched customization and paving the way for further developments in the travel industry.

VII. ACKNOWLEDGMENTS

Our profound appreciation goes out to the instructors, fellow students, and mentors who were instrumental in making the Sherpa project a reality. Their leadership, encouragement, and contagious excitement helped to transform this project into a worthwhile and prosperous endeavor. We extend a special thanks to Trip Advisor for the dataset which made the conception of our Hybrid Recommendation System possible.

VIII. REFERENCES

- [1] D. D. Lee and H. S. Seung, "Algorithms for Non-negative Matrix Factorization," in *NeurIPS*, 2001.
- [2] S. Zhang, W. Wang, J. Ford and F. Makedon, "Learning from Incomplete Ratings Using Non-negative Matrix Factorization," *SemanticScholar*, 2006.
- [3] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization," *NeurIPS*, 2007.
- [4] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE*, 2009.
- [5] Zhang, W. & Ding, G. & Chen, L. & Li and Chunping, "Augmenting Chinese Online Video Recommendations by Using Virtual Ratings Predicted by Review Sentiment Classification," *IEEE*, 2010.
- [6] J. Z. C. D. Quanquan Gu, "Collaborative Filtering: Weighted Nonnegative Matrix Factorization Incorporating User and Item Graphs," *SIAM*, 2010.
- [7] G. D. C. L. & J. W. Hailong Wen, "Matrix Factorization Meets Cosine Similarity: Addressing Sparsity Problem in Collaborative Filtering Recommender System," in *APWeb 2014*, 2014.
- [8] W. W. J. F. F. M. Sheng Zhang, "Using Singular Value Decomposition Approximation for Collaborative Filtering," *IEEE*, pp. 1-1, 2005.
- [9] M. Brauhoffer and F. Ricci, "Researchgate," Trip Advisor, 2016. [Online]. Available: https://www.researchgate.net/publication/308968574_TripAdvisor_Dataset.

PERFORMANCE ANALYSIS OF A HYBRID RECOMMENDER SYSTEM

Uzair Sultan +
Department of Computer Systems
Engineering, UET, Peshawar,
Pakistan.
uzairsultan10@gmail.com

Hajra Khan +
Department of Computer Systems
Engineering, UET, Peshawar,
Pakistan.
hajrakh3@gmail.com

Yasir Saleem Afridi +
Department of Computer Systems
Engineering, UET, Peshawar,
Pakistan.
yasirsaleem@uetpeshawar.edu.pk

Mian Ibad Ali Shah
Department of Computer
Systems Engineering, UET,
Peshawar, Pakistan.
ibad@uetpeshawar.edu.pk

+ *These authors have made equal contributions*

Abstract—In the prevailing information age, human confrontation to extensive information makes it difficult to segregate the relevant content on the basis of choices and priorities. This gives rise to the need of effective recommendation systems that can be incorporated in distinct and diversified domains such as e-commerce, social media and news media websites and applications. By giving suggestions, these recommender systems efficiently reduce huge information spaces and direct the users towards the items that best match their requirements and preferences. Hence, they play an important role in filtering out the relevant user specific information. Based on the working principle, recommender systems can be classified into Content-Based System, Collaborative Filtering System or Popularity-Based System. However, to cope-up with the problems of cold-start and plasticity that are associated with the standalone recommender systems, hybrid recommendation systems are being introduced. This research is therefore focused on the development of a Weighted Hybrid Model that combines the scores of the three standalone recommender models in a linear fashion. The performance of the proposed hybrid model is tested against all three standalone models on an online News dataset. Using a Top-N accuracy metric, it is found that the accuracy of the weighted hybrid model is higher than the standalone Content-Based, Collaborative and the Popularity-Based models against the same dataset. An efficiency of 90% for the Hybrid model was achieved comparing to the best performing standalone model having an efficiency of 53%.

1. INTRODUCTION

Data and information have gained unprecedented pace in today's world of information and technology. People have become considerably dependent on the technology that living without it seems inconceivable. People are overwhelmed with the extent of data available, hence, it is almost impossible to browse through such vast information space in search of the required and related news stories [1]. Consequently, the interest of users in the consumption of news either gets lowered or is completely lost.

Recommendation of news articles faces many challenges because of the dynamic environment, such as, new updates in the articles and changes in user preferences. Therefore, an efficient news recommendation system must be able to process and address the rapidly evolving and continuous inflow of news [1]

Content-based and collaborative filtering recommendation systems (discussed in Section 2) can provide some effective recommendations but each of the approaches come with few disadvantages. [2] To tackle this issue, this research is therefore focused on the development of a "Hybrid Recommendation Model" which has been built by combining content-based filtering and collaborative filtering systems. This hybrid model will also be combined with the popularity model to increase the accuracy of recommendations and to cope with problems such as cold start.

1.1. LITERATURE REVIEW

Recently, recommending news articles or other documents in the format of web objects has gained more research attention. Several adaptive news recommending systems, such as Google News and Yahoo! News provide personalized news recommendation services for a substantial number of online users [3].

A personalized new recommendation system with the help of a popular micro blogging service "Twitter" has been proposed by [4]. News articles are recommended based on popularity of the article identified from Twitter's public timelines.

Models and memory-based algorithms have been used which utilizes the weighted average of past ratings from other users and the weight is proportional to the similarity between the users [5]. Measures that have been used to find similarity are Pearson Correlation Coefficient and Cosine Similarity. This approach models the user preferences based on prior information.

The research carried out by [5] was taken further by using information filtering that filters the relevant information from the unwanted information stream [6]. Firstly, an analysis of user's interests spanned over a period of 14 months was carried out by using click distribution. Then Bayes rule was applied to predict the users' interest for a particular period of time. These predictions made for the particular times were then combined and a final prediction was made for the user against a longer period of the time.

In the earlier times, Popularity was one of the only metrics used for the news recommendation [7], [8] and [9]. The popularity-based news recommendations were based on the number of times a news article has been read by the users. However, this approach had a lot of shortcomings especially in cases where the news gets old, even though it is popular but it might not be of interest to the users. On the other hand, the newly published news with low popularity might be of more interest to the users.

A notable work in the area of News Recommendation Systems is SCENE (Scalable two-stage Personalized News Recommendation system) [10]. The main focus in this research was over the news selection. Experiments were conducted on how to match the news with the user interests while maintaining the highest accuracy and diversity. The goal was achieved by maintaining a huge metadata of the users, thereby effecting the news selection and improving the news recommendation.

The main application area of Content Based Filtering (CBF) technique is semantic-based Recommender and specific news websites. In order to address the shortcomings of both Collaborative Filtering (CF) and CBF approaches, a Hybrid recommendation approach is adopted. It has been observed that Hybrid approach is exhaustively used for news Recommender since 2010 (i.e., 60%), because the challenges faced by CBF and CF techniques can be overcome by using a complementary technique such as hybrid model. [11]

Collaborative filtering approach in recommendation system is demonstrated in Scalable Collaborative Filtering with Jointly Derived Neighbourhood Interpolation Weights. This improves the prediction accuracy by improving the interpolation precision and simultaneously deriving the interpolation weights for all nearest neighbours. [12]. This technique has been called Singular Value Decomposition (SVD) recommenders in [13]

Other related research includes Europe Media Monitor (EMM) News Explorer [14] and Newsjunkie [15] that uses the news content and named entities for carrying out recommendation of news. However, EMM News Explorer does not provide personalized services and Newsjunkie

does not address the news selection, news presentation and scalability issues.

In [16], the recommendation is carried out by using two modules: an offline module that pre-processes the data to build reader and content models, and an online module that uses these models in real time to recognize the reader's needs and goals and predict a recommendation list, accordingly. The recommended objects are obtained by using a range of recommendation strategies mainly based on content based filtering and collaborative filtering approaches, each applied separately or in a combination [16].

1.2. NOVELTY OF THE RESEARCH

The related work from the literature review mostly uses algorithms that involves collaborative filtering or content-based filtering when it comes to news/blogs recommendation. To achieve more promising and accurate results, two algorithms (namely, Collaborative and Content-Based Filtering) has been combined to propose a hybrid recommendation system which takes the weighted average of collaborative filtering score with content-based scores and performs recommendation ranking by resultant scores. Further, to cope with the cold start issue, this developed model has also been combined with popularity-based model. The accuracy of results achieved by this novel hybrid recommendation model increased exponentially.

2. RECOMMENDATION SYSTEMS

The definition of recommendation systems has evolved over the past 14 years. Recommender systems aim to predict users' interests and recommend items that are most likely of interest to them. These are one of the most powerful machine learning algorithms that are used to increase the traffic. [17] Recommendation systems are typically used to speed up the search process and making it easy for the users to reach the content that truly interests them. Several Recommendation systems have been deployed over the decade for different domains [18]. The two main types of recommendation systems [19] are discussed in Section 2.1 and 2.2.

2.1. CONTENT BASED FILTERING APPROACH

In content-based filtering a user profile is created which is then used to make recommendations to the user [20]. The data is collected, either explicitly (rating) or implicitly (clicking on a link). So, the idea in content-based filtering is to tag products using certain keywords, understand what the user likes, look up those keywords in

the database and recommend different products with similar attributes [20].

The main focus in content-based approach is on the attributes of the items, where the similarity among items is determined by comparing the attributes of these items. In content-based filtering approach, users or items are considered as atomic units and better-personalized recommendation is being made by gathering more information about the user or item [20].

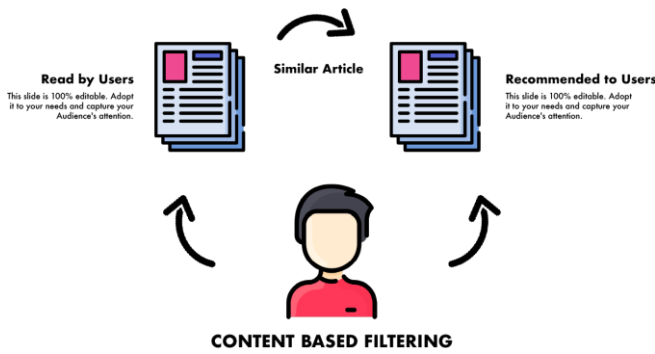


Figure 1: WORKING OF CONTENT-BASED FILTERING

For content-based filtering, the most popular vector space model Term Frequency–Inverse Document Frequency (TF-IDF) is used to select a set of candidate items. TF-IDF score is the product of term frequency (number of times a term appears in the document) and Inverse document frequency (measure of whether a term is rare or common in the collection of documents) [21].

Mathematically, the TF-IDF score for the word t in the document d from the document set D is calculated as follows [21]:

$$TF-IDF(t, d, D) = tf(t, f) \times idf(t, D)$$

Where,

$$tf(t, d) = \log \log (1 + freq(t, d))$$

$$idf(t, D) = \log \log ((N/count)(d \in D : t \in d))$$

TF-IDF algorithms apply to different dimensions of an article by selecting contents and for calculating similarity among them.

2.1.1. CHALLENGES FACED BY CONTENT BASED FILTERING

As content-based filtering decides only using the tags, it faces a few challenges in recommending as:

- o **Limited Content Analysis:** If the content doesn't contain enough information to discriminate the items precisely, the recommendation itself risks being imprecise [22].

- o **No Diversity:** User will never be recommended different items that he might like. Due to which business might not expand [22].

- o **New Users:** User Profile Information collected requires considerable amount of data and features to define an item. Adequate information is thus required to be able to recommend effectively [22].

2.2. COLLABORATIVE FILTERING APPROACH

Collaborative filtering is a technique that can filter out items that a user might like based on reactions by similar users. [22]. It is based upon the historical data and assumption here is that, if the user who have liked or views something in the past is likely to like or view similar content in the future as well. Collaborative filtering can be of two types [22].

A) User Based Collaborative Filtering

In user-based collaborative filtering, recommendations are made based on similar users who share the same rating pattern as of active user.

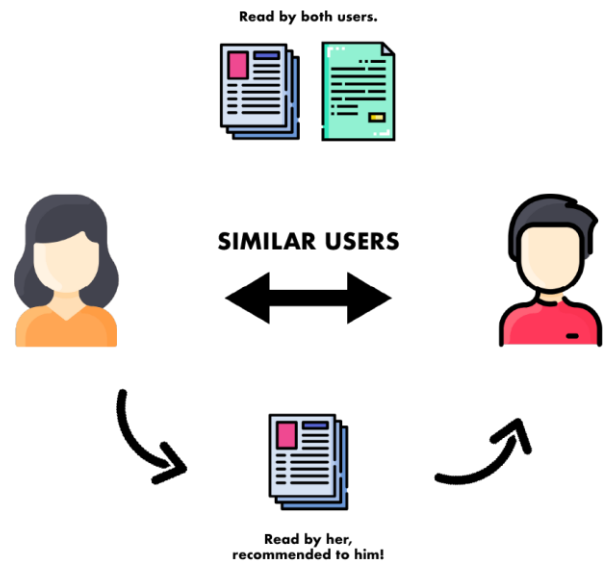


Figure 2: User-Based Collaborative Filtering

B) Item Based Collaborative Filtering

Item based collaborative filtering is a type of recommendation method which looks for similar items

based on the items users have already liked or positively interacted with.

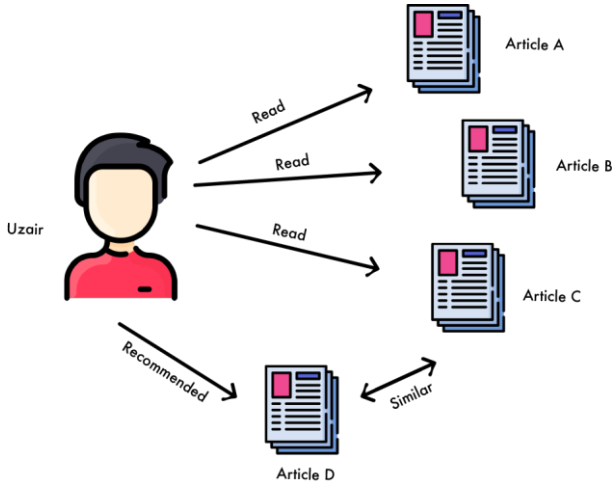


Figure 3: Item-Based Collaborative Filtering

Single Value Decomposition (SVD) is used as a Matrix factorization technique to find the similarities between users as well as similarities between the articles. An advantage of using this approach is that instead of having a high dimensional matrix containing abundant number of missing values, a much smaller matrix in lower-dimensional space is required. The more factors there are, the more specific the model will be. However, the model may result in over fitting, if too many factors are taken into consideration.

2.2.1. CHALLENGES FACED BY CONTENT BASED FILTERING

Some of the challenges faced by the collaborative filtering are:

- **Cold Start:** Cannot effectively handle the fresh items. Since, fresh items or users do not have enough rating to find a relation [7].
- **Scalability:** Collaborative filtering essentially depends on the number of users of a system. As the number of users increases, the complexity of analysing similar users and examining the history of the items used by these users increase exponentially. Scalability is also an issue where online systems need to react immediately to the requirements of the users [7].
- **Shilling Attack:** The abusing use of liking or disliking a product can affect this recommendation model [7].

- **Data Sparsity:** People mostly don't rate the items. Due to which there are missing ratings of a new item. This leads to poor recommendations. [7]

3. METHODOLOGY:

The proposed method was to combine the two most popular recommendation models, i.e., Content-based filtering and Collaborative filtering and develop more accurate hybrid model by eliminating drawbacks of each other. Further, to cope with the issue of cold start and also increase the model accuracy, proposed hybrid recommendation model is combined with the popularity-based recommendation model.

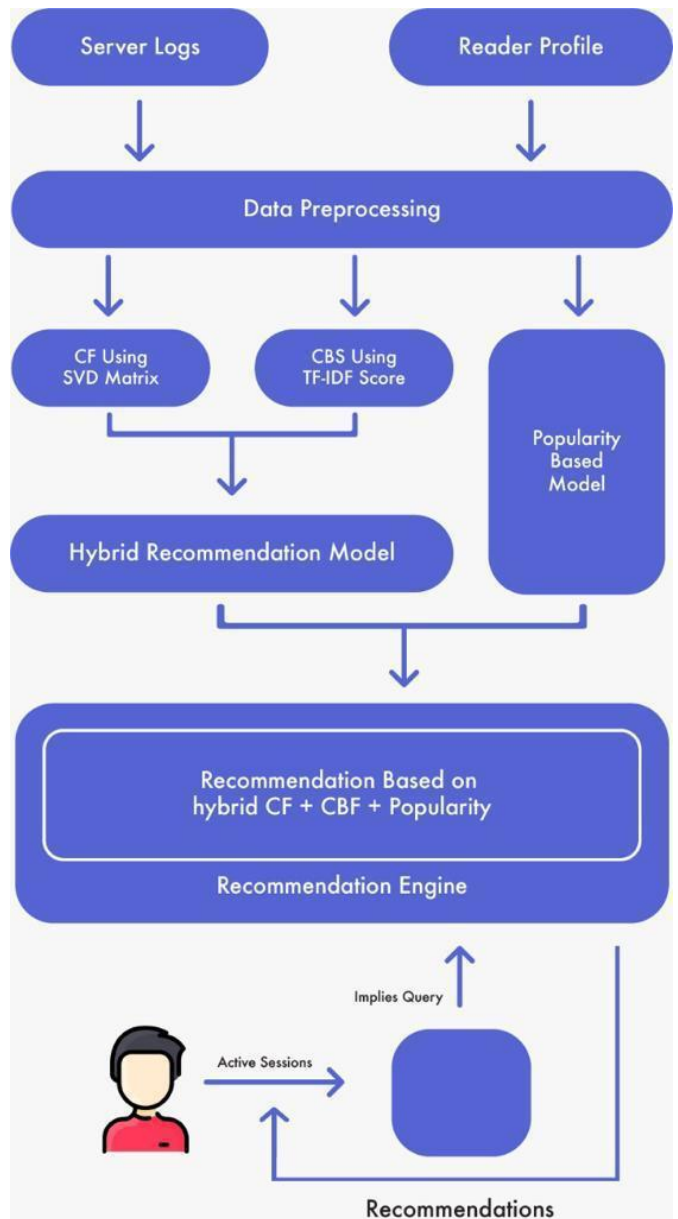


Figure 4: Flowchart of the overall methodology

3.1. EXPERIMENTAL SETUP

In this research, the recommendation models have been developed in Python. To overcome over fitting and to randomize the dataset, cross validation technique called Holdout has been used to validate the results. The data set was split in train data and test data, 20% of the data for testing purpose and 80% for training the model.

3.2. MODEL SELECTION – HYBRID RECOMMENDATION MODEL

Content-based and collaborative filtering can provide some effective recommendations but each of the approaches have some disadvantages. [23] Thus, in order to improve the results, a hybrid model has been used in which the two algorithms content based filtering and collaborative filtering was combined.

It takes the weighted average of Collaborative filtering - Score with Content- Based scores and performs recommendation ranking by resultant scores.

$$\text{Resultant Hybrid Score} = \text{CBF Score} \times \text{SVD Score}$$

This Hybrid model has also been combined with the popularity model to cope with cold start and to increase the accuracy of recommendation even more.

Popularity Based Recommendation Model – It takes the weighted average of Collaborative filtering Score with Content Based scores and performs recommendation ranking by resultant scores. popularity based recommendation system works with the trend. It basically uses the items which are in trend right now. For example, if any item which is usually viewed by every new user, then there are chances that it may suggest that item to the user who just visited.

“Popularity-Based Filtering Model sums up users’ interactions with each article and recommends the article with the most interactions.”

3.3. DATASET

The dataset used to train and test the hybrid recommendation model has been downloaded from kaggle [17] which contains the real samples of 12 months news. Dataset is properly labeled and has two parts. One part contains the **Articles** and their timestamp, contentID, authorID, contentType, URL, title and text. Second part contains **User Interactions** such as eventType, contentID, sessionID, userAgent (Browsers, etc.), userRegion,

userCountry, etc. It contains data of about **73000 logged-in users** and more than **3000 user interactions**.

3.4. DATA PRE-PROCESSING

There are five types of user interactions in the dataset. **VIEW** (The user has opened the article), **LIKE** (The user has liked the article), **BOOKMARKS** (The user has bookmarked the article for easy return in the future), **FOLLOW** (The user chose to be notified on any new comment in the article), **COMMENT** (The user created a comment in the article).

Since, there are different interaction types, specific weights should be associated to the interactions. For instance, “comment” on an article weighs more than a “simple view”. Henceforth, in the first step of pre-processing, weights were associated to each interaction that the user performs on the data based on their importance in accurate recommendations such as: View 1.0, Like 2.0, Bookmark 2.5, Follow 3.0, Comment 4.0.

In the second step, the users and interactions having weights less than five were removed. This is because the users and interaction with such low weight cannot be considered as base for recommendation. The details of the dataset after pre-processing can be seen in table...

Interactions before removal	72312
Interactions after removal	69869
Users before removal	1897
Users after removal	1140

In the final step, the Unique Interactions were calculated using already calculated weighted sum of interactions for each news article of every user.

Total Unique Interactions	39106
---------------------------	-------

3.5. EVALUATION METRICS

To find the efficiency of the proposed model, evaluation has been carried out as it is one of the most important aspects to any machine learning project. Evaluation enables to compare different hyper-parameter choices and algorithms for the models. [24]. Evaluation also helps us identify the generic nature of the proposed model, also known as Generalization. Hence, holdout method of cross validation has been used to generalize our model by tuning it accordingly.

Evaluation technique that has been used in this research is the Top-N accuracy metrics. It evaluates the accuracy by

comparing the recommendations provided to the user with the items the users actually interacted with in the test set [24]. The Top-N accuracy metric that was chosen for the hybrid model is Recall@N which evaluates whether the interacted item is amongst the top N items in the ranked list. [24]

4. RESULTS AND DISCUSSIONS

The evaluation of Popularity Model was performed by the method discussed in section 3, and it is surprising you that a popularity model can perform this well. It achieved the Recall@5 of 0.241, meaning 24% of interacted items in test sets were ranked by popularity model among the top-5 item. Recall@10 was 37% higher as expected.

In the content-based filtering, a Recall@5 of 0.414 was achieved, which means about 42% of interacted items in test set were tanked by content-based filtering among the top-5 items. Recall@10 was higher at 53%.

In the Collaborative filtering, a Recall@5 of 0.333 was achieved, which means about 33% of interacted items in test set were tanked by content-based filtering among the top-5 items. Recall@10 was higher at 47%.

After these results, the developed Hybrid Recommendation model was also evaluated, and the results were higher than all other recommendation models. A Recall@5 of 0.434 was achieved, which means about 44% of the interacted items in the test set were ranked by Hybrid Model among the top-5 items. Recall@10 was 54%. After combining the popularity-based model with the Hybrid model, the Recall@5 increased up to 65% and Recall@10 reaches up to 90%. These results can be thoroughly seen in table 1.

4.1. COMPARISON OF DIFFERENT RECOMMENDATION MODELS

The results of recommendation models working independently were compared with the proposed hybrid model. The results were clearly in favor of the hybrid recommendation model. It has been found that when the proposed Hybrid model was merged with Popularity model, the accuracy was even increased to 90%. Comparison of the results has been shown in Table 1 and represented graphically in Figure 5.

Model Name	Recall@5	Recall@10
Popularity Model	24%	37%
Content Based Filtering	42%	53%
Collaborative Filtering	33%	47%
Hybrid Model	44%	54%
Hybrid + Popularity	65%	90%

Table 1: Tabular Comparison of Recommendation Models

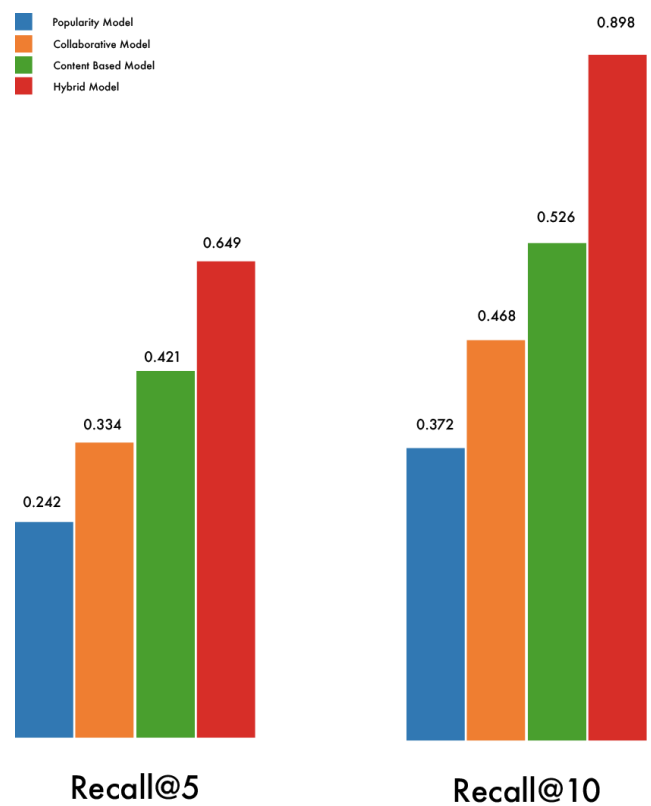


Figure 5: Graphical representation of the accuracy of Different Recommendation Models

5. CONCLUSION

Content-based and collaborative filtering can provide some effective recommendations but each of the approaches have some disadvantages. In this research, the development of a Weighted Hybrid Model was proposed that combines the scores of the three standalone recommender models in a linear fashion. The hybrid recommendation model takes the weighted average of collaborative filtering score and the content-based filtering scores to carry out the recommendation ranking based on

the resultant scores. This Hybrid model has also been combined with the popularity model to cope with cold start and to increase the accuracy of recommendation even more. The performance of the proposed hybrid model is tested against all three standalone models on an online News dataset. Using a Top-N accuracy metric, it is found that the accuracy of the weighted hybrid model is higher than the standalone Content-Based, Collaborative and the Popularity-Based models against the same dataset. An efficiency of 90% was achieved for the proposed Hybrid model comparing to the best performing standalone model having an efficiency of 53%. This research is not limited to news websites only, but it can be extended to other recommendations such as in e-commerce.

REFERENCES

- [1] Chaturvedi, A. K., Peleja, F., & Freire, A. (2017). Recommender system for news articles using supervised learning. *arXiv preprint arXiv:1707.00506*.
- [2] Jonnalagedda N, Gauch S, Labille K, Alfarhood S. 2016. Incorporating popularity in a personalized news recommender system. *PeerJ Computer Science* 2:e63 <https://doi.org/10.7717/peerj-cs.63>
- [3] Ismail, Walaa & Nasr, Mona & Saied, Mohamed, "A HYBRID NEWS RECOMMENDER SYSTEM," Cairo, Egypt., 11th-12th April 2018.
- [4] Steinberger, Ralf & Pouliquen, Bruno & van der Goot, Erik. (2013). An introduction to the Europe Media Monitor family of applications. Technical Note No.I.
- [5] Zhou, Y., Wilkinson, D., Schreiber, R., Pan, R. (2008). Large-Scale Parallel Collaborative Filtering for the Netflix Prize. In: Fleischer, R., Xu, J. (eds) *Algorithmic Aspects in Information and Management. AAIM 2008. Lecture Notes in Computer Science*, vol 5034. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-68880-8_32.
- [6] R. M. Bell and Y. Koren, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights," *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, Omaha, NE, USA, 2007, pp. 43-52, doi: 10.1109/ICDM.2007.90.
- [7] C. Feng, M. Khan, A. U. Rahman and A. Ahmad, "News Recommendation Systems - Accomplishments, Challenges & Future Directions," in *IEEE Access*, vol. 8, pp. 16702-16725, 2020, doi: 10.1109/ACCESS.2020.2967792.
- [8] I. Cantador and P. Castells, "Semantic contextualisation in a news recommender system", *Proc. Workshop Context-Aware Recommender Syst. (CARS)*, pp. 1-6, 2009.
- [9] J Zhou W, Wen J, Qu Q, Zeng J, Cheng T (2018) Shilling attack detection for recommender systems based on credibility of group users and rating time series. *PLoS ONE* 13(5): e0196533. <https://doi.org/10.1371/journal.pone.0196533>
- [10] Li, Lei & Wang, Dingding & Li, Tao & Knox, Daniel & Padmanabhan, Balaji. (2011). SCENE: a scalable two-stage personalized news recommendation system. 125-134. [10.1145/2009916.2009937](https://doi.org/10.1145/2009916.2009937).
- [11] P. Viana and M. Soares, "A hybrid approach for personalized news recommendation in a mobility scenario using long-short user interest," *Int. J. Artif. Intell. Tools*, vol. 26, no. 02, Apr. 2017, Art. no. 1760012
- [12] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*. 2005;17(6):734–749.
- [13] Khattar, D., Kumar, V., Gupta, M., & Varma, V. (2018). Neural Content-Collaborative Filtering for News Recommendation. *NewsIR@ ECIR, 2019*(2018), 45-50
- [14] Jiang, S., & Hong, W. (2014, August). A vertical news recommendation system: CCNS—An example from Chinese campus news reading system. In *2014 9th International Conference on Computer Science & Education* (pp. 1105-1114). IEEE.
- [15] Das, A. S., Datar, M., Garg, A., & Rajaram, S. (2007, May). Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web* (pp. 271-280).
- [16] Liu, J., Dolan, P., & Pedersen, E. R. (2010, February). Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces* (pp. 31-40).
- [17] Zhang, Q., Lu, J., & Jin, Y. (2021). Artificial intelligence in recommender systems. *Complex & Intelligent Systems*, 7(1), 439-457.
- [18] Gabrilovich, Evgeniy & Dumais, Susan & Horvitz, Eric. (2004). Newsjunkie: {P}roviding Personalized Newsfeeds via Analysis of Information Novelty. *Thirteenth International World Wide Web Conference Proceedings, WWW2004*. 482-490. [10.1145/988672.988738](https://doi.org/10.1145/988672.988738).
- [19] Ekstrand, M. D. (2020, October). Lenskit for python: Next-generation software for recommender systems experiments. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 2999-3006).

[20] Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence, 2009*.

[21] Garcia Esparza, S., O'Mahony, M. P., & Smyth, B. (2011, October). A multi-criteria evaluation of a user generated content based recommender system. In *Freyne, J. et al.(eds.). Proceedings of the 3rd ACM RecSys' 10 Workshop on Recommender Systems and the Social Web*.

[22] Chaturvedi, A. K., Peleja, F., & Freire, A. (2017). Recommender system for news articles using supervised learning. *arXiv preprint arXiv:1707.00506*.

[23] Jonnalagedda, N., Gauch, S., Labille, K., & Alfarhood, S. (2016). Incorporating popularity in a personalized news recommender system. *PeerJ Computer Science, 2*, e63.

[24] Deshpande, M., & Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS), 22(1)*, 143-177.

[25] Khattar, D., Kumar, V., Gupta, M., & Varma, V. (2018). Neural Content-Collaborative Filtering for News Recommendation. *NewsIR@ ECIR, 2079(2018)*, 45-50.

NEUROSCAN: Revolutionizing brain tumor detection using Vision-Transformer

Kamran khan

Computer Systems Engineering,
University of Engineering & Technology,
Peshawar, Pakistan
20pwcse1895@uetpeshawar.edu.pk

Najam Aziz

Computer Systems Engineering,
University of Engineering & Technology,
Peshawar, Pakistan
najamkhattak@gmail.com

Afaq Ahmad

Computer Systems Engineering,
University of Engineering & Technology,
Peshawar, Pakistan
20pwcse1876@uetpeshawar.edu.pk

Munib-ur-Rehman

Computer Systems Engineering,
University of Engineering & Technology,
Peshawar, Pakistan
20pwcse1887@uetpeshawar.edu.pk

Yasir Saleem Afridi

Computer Systems Engineering,
University of Engineering & Technology,
Peshawar, Pakistan

Abstract— Brain tumor detection is a pivotal component of neuroimaging, with significant implications for clinical diagnosis and patient care. In this study, we introduce an innovative deep learning approach that leverages the cutting-edge Vision Transformer model, renowned for its ability to capture complex patterns and dependencies in images. Our dataset, consisting of 3000 images evenly split between tumor and non-tumor classes, serves as the foundation for our methodology. Employing Vision Transformer architecture, we processed high-resolution brain scans through patching and self-attention mechanisms. The model is trained through supervised learning to perform binary classification task. Our employed model achieved high of 98.37% in tumor detection. While interpretability analysis was not explicitly performed, the inherent use of attention mechanisms in the Vision Transformer model suggests a focus on important brain regions and enhances its potential for prioritizing crucial information in brain tumor detection.

Keywords: Brain Tumor Detection, Medical Imaging, Classification, Vision Transformers, ViT, Machine Learning, Deep Learning

1. INTRODUCTION

Brain tumors are a major global health concern, and by early identification patient healthcare can be greatly enhanced. Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are commonly used medical imaging techniques for tumor diagnosis in human brain. However, the diagnosis still relies on doctor or radiologist decision, which is error prone and time-consuming process. In order to support medical personal in making fast and precise diagnosis, the increasing amount of brain imaging data need sophisticated computational algorithms.

This research presents the incorporation of a type of artificial neural networks called vision transformer (ViT) for the brain tumor detection. ViT are a viable option for medical image analysis, since they have demonstrated exceptional performance in a variety of computer vision tasks. The purpose of the research is to assess the accuracy of vision transformers while detecting brain tumors.

Vision transformers are a useful tool for detecting brain tumors because of their capacity to extract contextual information and spatial correlations from medical images. In contrast to conventional convolutional neural networks (CNNs), which analyze images using a fixed grid-like structure, vision transformers use a self-attention mechanism

that enables them to take interactions and long-range relationships between various image regions into account.

Additionally, vision transformers have proven to perform remarkably well in a variety of computer vision applications, such as segmentation, object identification, and image classification. They are a desirable alternative for medical image analysis, where reliable and accurate abnormality detection is crucial, due to their capacity to learn from large-scale datasets and generalize well to unseen data.

This research aims to assess the effectiveness of ViT for identifying brain tumors through extensive experimentation and achieve an acceptable accuracy by ViT for the said task. Our research will demonstrate the potential of ViT as a valuable tool for diagnosing brain tumors in clinical setting.

The research paper has been divided into five sections. Section 1 elaborates the Introduction, Section 2 covers the Literature Review, Section 3 covers the methodology, Section.4 discusses results and section 5 conclude the paper.

2. LITERATURE REVIEW

During the previous decade, deep learning algorithm made a great hype by their high performance in various domains ranging from textual analysis to image analysis. These algorithms also extensively used for healthcare data and helped in automating various tasks related to healthcare while achieving comparable accuracy to that of human. The incorporation of deep learning techniques has resulted in a remarkable revolution in the field of brain tumor identification. Conventional approaches in this field mostly depend on manually designed features and rule-based algorithms, which frequently prove inadequate for managing the complex and diverse nature of medical images. In a research study, Kumar et al. [1], has made significant progress in proving that Convolutional Neural Networks (CNNs) are effective at automating the process of identifying brain tumors. Conventional methods have difficulties when it comes to capturing spatial hierarchies and patterns inside medical images. CNNs have proven essential in overcoming these challenges.

Even with CNNs' significant success, it is still necessary to investigate new approaches. Hossain et al. [2] developed the concept of vision transformers and provided a new dimension to improving the detection of brain tumors. Vision transformers provide a comprehensive viewpoint by

removing global dependencies from images, which may enhance the model's capacity to identify complex and subtle patterns suggestive of malignancies. By customizing vision transformers for the classification of medical images, Manzari et al. [3] have made a significant contribution to this investigation. Their research highlights how adaptable these transformer models are at handling the difficulties presented by intricate medical imagery.

The quick advancement of medical imaging technology emphasizes how crucial it is to identify brain tumors with accuracy. Puttagunta et al. [4] offer an extensive overview of deep learning methods for medical images interpretation. Their work describes the changing environment in which precise identification becomes more and more important, in addition to highlighting the revolutionary potential of these tools. Furthermore, the work of Sadad et al. [5] offers an insightful information about brain tumor detection through a wider variety of machine learning algorithms, offering a comprehensive overview of the difficulties faced and prospects in the field.

Ahn et al. [6] in their study, provides the theoretical foundation for comprehending attention mechanisms, which are a crucial aspect of vision transformers. The model can detect global dependencies in images, thanks to this attention mechanism, which is in line with the complex needs of brain tumor diagnosis.

The field of brain tumor diagnosis has experienced a dynamic shift due to the incorporation of state-of-the-art deep learning techniques. This development was prompted by the limits of traditional approaches that relied on manually created features and rule-based algorithms. Recent research has demonstrated the revolutionary potential of convolutional neural networks (CNNs) in automating the complex process of recognizing brain tumors, as demonstrated by the work of Kumar et al. [1]. With their capacity to extract intricate patterns and spatial hierarchies from medical images, CNNs are a key component in the development of diagnostic techniques.

The study of Brownlee et al [7], which focuses on mastering machine learning techniques by providing useful information that can support the continued advancement and deployment of deep learning techniques in the particular field of brain tumor identification. The notion of vision transformers was created by Akinyeluet al. [8] added a new layer to enhance to ViT for brain tumor identification. By eliminating global dependencies from images, vision transformers offer a holistic perspective that may improve the model's ability to recognize intricate and nuanced patterns suggestive of tumors. A noteworthy contribution to this study has been made by Wanget al. [9], who customized vision transformers for the classification of medical images. Their study demonstrates how versatile these transformer models are in managing the challenges posed by complex medical images.

Jiang, et al. [10], in their work provided in-depth review of deep learning techniques for the interpretation of medical imaging data. Their work highlights the revolutionary potential of these techniques and outlines the evolving environment in which exact identification becomes increasingly vital. Moreover, Tummalaet al. [11] provide

informative data regarding the identification of brain tumors using a greater range of machine learning algorithms, providing a thorough summary of the challenges encountered and future prospects in the field.

Asiriet al. [12] utilized attention mechanism to enables the model to identify global dependencies in images, which is consistent with the requirements of complicated task of brain tumor diagnosis.

The work of [14, 15, 16, 17] provides a variety of perspectives and approaches that greatly enhance our knowledge and progress in the field of brain tumor detection using deep learning techniques.

3. METHODOLOGY

This section explains the overall methodology of this research and is presented in Figure.1 by the methodology flow diagram.

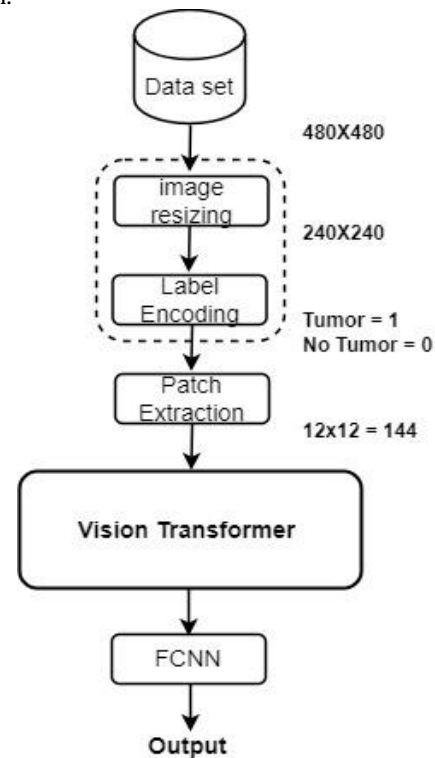


Figure 1. Methodology flow diagram

3.1 Data Description

The dataset that has been utilized for our research study is Br35H Brain Tumor Detection 2020 Dataset and is available on Kaggle. The dataset contains 3000 T1 weighted images of tumorous (yes) class and healthy (no) brain scans. Each class has 1500 images and equal distribution of the Dataset. Sample MRI brain scans of tumor and without tumors are shown in Figures 2(a) and (b), respectively.

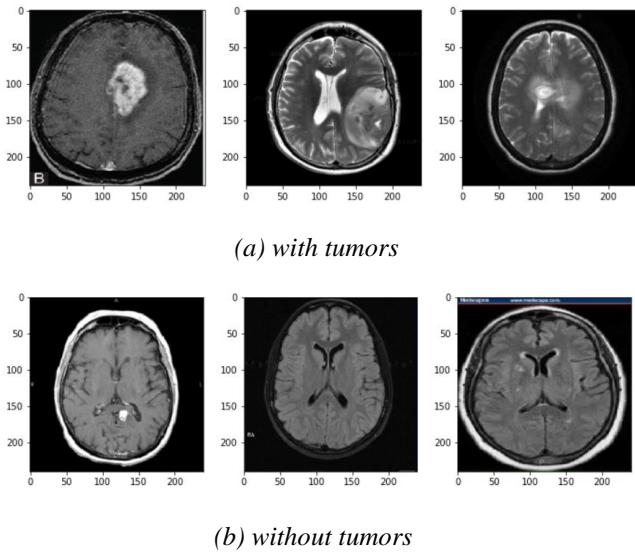


Figure 2. Samples from the dataset

3.2 Preprocessing

Data preprocessing is a vital step in preparing data that is suitable for model training. This enhances data quality, compatibility and ensure optimal performance by the model being trained. Various preprocessing steps that are applied on the utilized dataset are explained as follow;

- *Resizing*: The dataset contains images that are of varying sizes. Hence, resizing is done to bring every image to 240×240 pixels, which was an important first step. This consistency allowed our model to have consistent input dimensions, which made processing easier.
- *Normalization*: In order to scales data to a common range, normalization is done. The process ensures that all the input features have a similar influence on the model, preventing certain features from dominating the learning process due to differences in their original scales. This aid in effective learning and performance improvement of the model. To achieve normalization, we used the following mathematical transformation.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where the resized image is represented by X and the normalized image is represented by X' .

- *Label Encoding*: By label encoding we converted the categorical labels yes and no for brain tumor presence and absence, into numerical values to make sure they are compatible with our model.

3.3. Vision Transformer (ViT)

The Vision Transformer, or ViT, is a model for image classification that employs a Transformer-like architecture over patches of the image. An image is split into fixed-size patches, each of them are then linearly embedded, position embeddings are added, and the resulting sequence of vectors

is fed to a standard Transformer encoder. In order to perform classification, the standard approach of adding an extra learnable “classification token” to the sequence is used [19].

The intricate details of medical images are catered in the vision transformer architecture. In order to process image patches and enable the network to capture both local and global information, the model is composed of a number of transformers block as shown in Figure 3. The model can assess the relative importance of various regions in the input images using the self-attention process.

3.4 Model Architecture

The model architecture we have implemented for this study is depicted in Figure.3. and is explained as follows;

- *Image Patching*: The input to the model is split up into smaller patches as part of the patching process. This separation makes processing more efficient and makes it possible for the model to successfully collect local spatial information.
- *Image Flattening*: After patching, the two-dimensional spatial information is converted into a linear representation by flattening the patches. The data is now ready for additional processing inside the model.
- *Patch Encoder*: Patch Encoder module processes the flattened patches. This module's dense projection layer and positional encoding encode the patches' spatial information and give the locations of the patches within the image context.
- *Positional Encoding*: An essential part of embedding positional information into the patch representations is positional encoding. As a result, the model is able to comprehend the spatial relationships between patches and gather crucial categorization context.
- *Transformer Encoder Block*: The Transformer Encoder Blocks are the central component of the model. These blocks are made up of feedforward neural network layers and multi-head self-attention layers. They facilitate the model's ability to extract high-level characteristics from the encoded patches and to capture global dependencies.
- *Aggregate Representation*: Upon completion of several Transformer Encoder Blocks, the representations obtained from every patch are combined. The contextual data that is acquired from every patch is combined throughout this aggregation phase to provide a complete representation of the full image.
- *Classification*: In the end, a classification head made up of dense layers receives the combined representation. To ascertain whether or not a brain tumor is present in the input image, these layers carry out categorization.

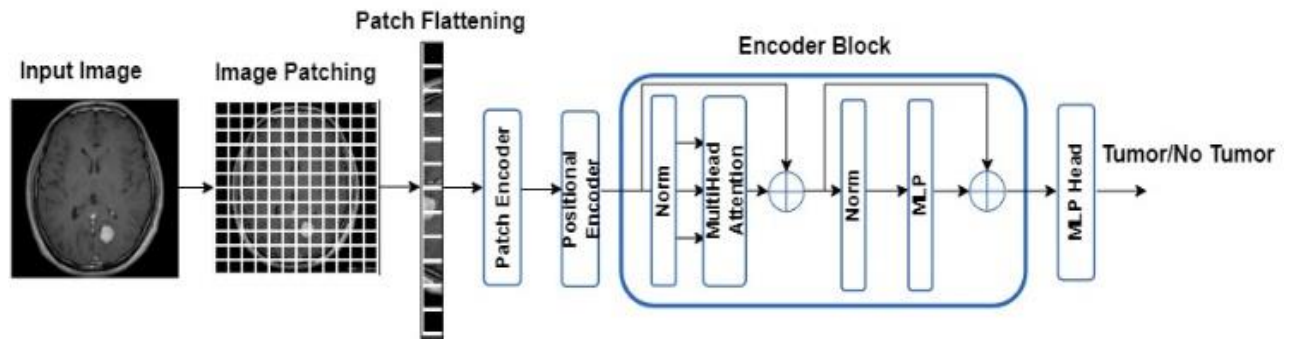


Figure 3 :Model Architecture

3.5 Training Strategy

Here, we describe the training approach that we used to train our brain tumor identification model using the Br35H: Brain Tumor detection 2020 Kaggle dataset in an efficient manner.

Data Splitting: We divided the dataset into separate training and testing sections in order to create a strong training regimen. To ensure there was enough data for the model to learn, we set aside 80% of the dataset for training. The remaining 20% was set aside for testing, allowing for an objective assessment of the performance of the trained model.

Hyperparameters Settings:

- **Learning rate:** We regulate the rate at which the model modifies its parameters during optimization by setting the learning rate to 0.001. The trade-off between parameter stability and convergence speed is balanced by this decision.
- **Weight Decay:** To prevent overfitting inclinations, a weight decay of 0.0001 was used as a regularization strategy. Model generalization is promoted by weight decay, which penalizes large parameter values.
- **Batch size:** The number of samples processed in each training iteration was determined by our training procedure, which included a batch size of 32. This batch size balances gradient stability with computational efficiency.
- **Number of epochs:** The training process took place over fifty epochs, allowing for iterative dataset learning. The model is exposed to the data throughout several epochs, which promotes model convergence and parameter refining.
- **Image Size and Patch Size:** After resizing the input images to 240 by 240 pixels, patches with a size of 20 by 20 pixels were extracted. These dimensions played a crucial role in specifying the input organization and spatial resolution of the feature extraction process of the model.
- **Transformer Architecture:** Eight transformer layers, each incorporating four attention heads, made up our model architecture. Dense layers with units [2048, 1024] were featured in the final classifier, while the

transformer units had size [128, 64]. These architectural decisions were carefully considered in order to maximize the expressiveness of the model and its ability to identify complex patterns in the data.

4. RESULTS and DISCUSSION

The implemented ViT model shows in impressive performance while utilizing the Br35H dataset. The model performance plots are shown in Figure.4. and the classification report is shown in Table.1.

a) Training and Validation loss graph:

The Graph shows how training and validation loss changed over the course of 50 epochs while your brain tumor detection model was being trained. At first, there is a declining trend in both the training and validation losses, which suggests that the model is successfully learning from the data. Both trajectories, however, converge at the 50th epoch, indicating that the model has reached a stage at which more training no longer considerably enhances its performance. This convergence shows that the model is performing at its best and has successfully captured the underlying patterns in the data. The losses then level out or maybe even begin to rise, indicating that the model has taken in all the information it can from the training set. Overall, the convergence after 50 epochs shows that training was successful and efficient by the model.

b) Training and Validation Accuracy graph:

The accuracy of our brain tumor detection model during training and validation is shown in the graph. Both accuracy levels rise gradually, peaking close to the 50th epoch. At this stage, the model obtains an impressive testing accuracy of 98.37%, demonstrating its efficacy in correctly categorizing photos of brain tumors. This great accuracy shows how reliable and appropriate the model is for practical use in clinical settings.

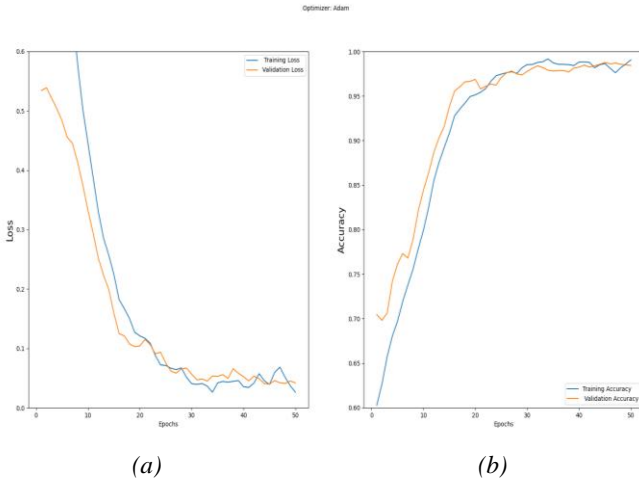


Figure 4. ViT training and validation (a) Loss (b) Accuracy

Table 1. Classification Report

	Precision	Recall	F1-Score	Support
Yes	1.00	0.99	0.99	300
No	0.97	0.99	0.98	300
Accuracy			0.98	600

Explanation of classification Report

- **Precision:** Precision is defined as the percentage of actual positive forecasts among all positive predictions. The precision for the "Yes" class (brain tumor presence) in this instance is 1.00, meaning that all of the cases that were predicted to have brain tumors were accurate. Similarly, the precision for the "No" class (absence of brain tumor) is 0.97, meaning that 97% of cases that were predicted to be brain tumor-free were accurate.
- **Recall:** The percentage of true positive predictions among all actual positive cases is measured by recall, which is sometimes referred to as sensitivity. With a recall of 0.99 for the "Yes" class, the model was able to accurately identify 99% of real brain tumor cases. The recall for the "No" class is likewise 0.99, meaning that 99% of real cases.
- **Support:** In the dataset, Support is the total number of instances of each class. Our test data consist of 300 MRI scans from each class, totaling 600.
- **Accuracy:** Measured as the percentage of correctly classified occurrences over the total number of examples, accuracy assesses the overall correctness of the model's predictions. With an accuracy of 0.98, 98% of the cases in the dataset were properly classified by the model.

5. CONCLUSION

The field of medical image processing is undergoing a fundamental paradigm changes with the integration of vision transformers for brain tumor diagnosis. Vision transformers are extremely useful tools for healthcare practitioners because of

their special qualities, which include their interpretability and ability to understand complex relationships among images. One important function of the attention mechanism, which is a distinguishing characteristic of vision transformers, is to shed light on the particular characteristics that influence the model's judgment. In the medical industry, confidence and comprehension of the diagnostic procedure are greatly enhanced by transparency. The vision transformer model is being improved and optimized on a continuous basis. To evaluate the model's generalizability and robustness, extensive research is being conducted across a wide range of datasets and clinical settings.

Finally, this study emphasizes how important it is to incorporate vision transformers into the field of brain tumor identification. The model's proven performance on a variety of complex medical images represents a significant breakthrough in diagnostic skills. In addition to their exceptional performance, vision transformers can be interpreted, giving medical professionals useful information on the decision-making process. In addition to having the capacity to interpret intricate patterns, these models have the potential to improve accuracy, which makes them revolutionary instruments for medical practitioners. This study represents not only a significant advancement but also a possible paradigm changes in the field of brain tumor detection.

REFERENCES

- [1] Kumar, Sunil, Renu Dhir, and Nisha Chaurasia. "Brain Tumor Detection Analysis Using CNN: A Review." 2021 *International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. IEEE, 2021..
- [2] Hossain, Shahriar, et al. "Vision transformers, ensemble model, and transfer learning leveraging explainable ai for brain tumor detection and classification." *IEEE Journal of Biomedical and Health Informatics* (2023).
- [3] Manzari, Omid Nejati, et al. "MedViT: a robust vision transformer for generalized medical image classification." *Computers in Biology and Medicine* 157 (2023): 106791.
- [4] Puttagunta, Muralikrishna, and S. Ravi. "Medical image analysis based on deep learning approach." *Multimedia tools and applications* 80 (2021): 24365-24398.
- [5] Sadad, Tariq, et al. "Brain tumor detection and multi - classification using advanced deep learning techniques." *Microscopy Research and Technique* 84.6 (2021): 1296-1308.
- [6] Ahn, Kwangjun, et al. "Linear attention is (maybe) all you need (to understand transformer optimization)." arXiv preprint arXiv:2310.01082 (2023).
- [7] Brownlee, Jason. "Data Cleaning, Feature Selection, and Data Transforms in Python." *Machine Learning Mastery* (2020).
- [8] Akinyelu, Andronicus A., et al. "Brain tumor diagnosis using machine learning, convolutional neural networks, capsule neural networks and vision transformers, applied to MRI: a survey." *Journal of imaging* 8.8 (2022): 205.
- [9] Wang, Wenxuan, et al. "Transbts: Multimodal brain tumor segmentation using transformer." *Medical Image Computing and Computer Assisted Intervention—MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part I* 24. Springer International Publishing, 2021.
- [10] Jiang, Yun, et al. "SwinBTS: A method for 3D multimodal brain tumor segmentation using swin transformer." *Brain sciences* 12.6 (2022): 797.

- [11] Tummala, Sudhakar, et al. "Classification of brain tumor from magnetic resonance imaging using vision transformers ensembling." *Current Oncology* 29.10 (2022): 7498-7511.
- [12] Asiri, Abdullah A., et al. "Exploring the Power of Deep Learning: Fine-Tuned Vision Transformer for Accurate and Efficient Brain Tumor Detection in MRI Scans." *Diagnostics* 13.12 (2023): 2094.
- [13] Swati, Z.N.K.; Zhao, Q.; Kabir, M.; Ali, F.; Ali, Z.; Ahmed, S.; Lu, J. Brain tumor classification for MR images using transfer learning and fine-tuning. *Comput. Med. Imaging Graph* 2019, 75, 34–46.
- [14] AlBadawy, E.A.; Saha, A.; Mazurowski, M.A. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Med. Phys.* 2018, 45, 1150–1158.
- [15] Zeineldin, R.A.; Karar, M.; Coburger, J.; Wirtz, C.; Burgert, O. DeepSeg: Deep neural network framework for automatic brain tumor segmentation using magnetic resonance FLAIR images. *Int. J. Comput. Assist. Radiol. Surg.* 2020, 15, 909–920.
- [16] Sharif, M.I.; Li, J.; Amin, J.; Sharif, A. An improved framework for brain tumor analysis using MRI based on YOLOv2 and convolutional neural network. *Complex Intell. Syst.* 2021, 7, 2023–2036
- [17] Asiri, A.A.; Aamir, M.; Shaf, A.; Ali, T.; Zeeshan, M.; Irfan, M.; Alshamrani, K.A.; Alshamrani, H.A.; Alqahtani, F.F.; Alshehri, A.H.D. Block-Wise Neural Network for Brain Tumor Identification in Magnetic Resonance Images. *Comput. Mater. Contin.* 2022, 73, 5735–5753.
- [18] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).
- [19] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Detection of Holes in Point Clouds using Statistical Technique

Zain Ul Abideen

Departemtn of Computer Systems Engineering
University of Engineering and Technology,
Peshwar, Pakistan
zainikhan3434@gmail.com

Hamza Ali

Departemtn of Computer Systems Engineering
University of Engineering and Technology,
Peshwar, Pakistan
hamzaali.dcse@gmail.com

Muhammad Sajjad

Departemtn of Computer Systems Engineering
University of Engineering and Technology,
Peshwar, Pakistan
muhammadsajad2710@gmail.com

Muhammad Abeer Irfan

Departemtn of Computer Systems Engineering
University of Engineering and Technology,
Peshwar, Pakistan
abeer.irfan@uetpeshawar.edu.com

Atif Jan

Department of Electrical Engineering
University of Engineering and Technology,
Peshwar, Pakistan
atifjan@@uetpeshawar.edu.pk

Yasir Saleem

Department of Computer Systems Engineering
University of Engineering and Technology,
Peshwar, Pakistan
yasirsaleem@uetpeshawar.edu.pk

Abstract— A point cloud is a dynamic, three-dimensional geometric representation of data that has different qualities for every point, including geometry, normal vectors, and color. However, holes that often occur during the 3D point cloud collection process provide an immense obstruction to the analysis and reconstruction of point clouds. Thus, detecting these holes is a crucial initial step toward obtaining precise and comprehensive representations of the real surfaces. Although there are several methods available for hole detection and filling, the problem is exacerbated by their shortcomings, which include high computation complexity or limited effectiveness. Our method is based on a sequence of basic but efficient statistical techniques. Our method is based on a sequence of basic but efficient statistical techniques. First, we find the mean distances between each point using the K Nearest Neighbors (KNN) technique. Next, we can categorize normal points and points that belong to holes and borders by using this mean as a threshold. Our method's simplicity and low computational resource needs offer significant advantages over other approaches.

Keywords: Point clouds; Hole detection; Surface reconstruction; Statistical analysis.

I. INTRODUCTION

The evolution of laser scanning technology has democratized the process of digitizing objects, enabling precise and efficient data acquisition at scale. This advancement has catalyzed the emergence of reverse engineering, a process wherein digital models are generated from point cloud data as a valuable tool in various industries. A point cloud is a collection of points of data plotted in 3D space, using a 3D laser scanner. Such as, when scanning a building, every virtual point would correspond to a real spot on the metalwork, wall, window, or any other surface where the laser beam comes into contact. A point cloud can be defined as “A Point Cloud is an unordered set of 3-dimensional points in a frame of reference (Cartesian coordinate system) on the surface of objects.” The point cloud is a set of discrete points obtained by scanning the object's surface with a three-dimensional (3D) scanning device. The point cloud can directly and stereoscopically represent the geometric features of the object's surface. Various additional details can be used along with 3D coordinates to represent them. A few are listed below:

- i. RGB color information associated with each point.
- ii. Normal to surface at each point.
- iii. Information about meshes (vertices & edges).

3D laser scanner technology is the foundation of 3D point cloud data. These electronic devices use visible light and lidar-based technology to gather detailed information about a specific area. The performance of the scanning device in use has a major impact on the final point cloud's clarity and accuracy. Nowadays, there are various sensors available for 3D scanning which are based on various principles some of which are mentioned below:

- i. Structured Light Cameras
- ii. Time-of-flight sensors
- iii. Stereo Vision
- iv. Microsoft Kinect Cameras

Point cloud is of paramount importance in 3D object reconstruction, and object recognition with its high precision and fast processing speed [1][2]. It is also useful in Simultaneous Localization and Mapping (SLAM) and Preservation & restoration. It is possible to scan the complete surface of an object using high-fidelity scanners, but holes may appear in the final integrated model due to occlusions and manoeuvrability of the scanner while scanning. The holes in 3D objects as shown in Figure 1 may also occur due to factors such as occlusion, low surface reflectance coefficients, high grazing angles, missing parts in the original object, limited number of range scans from various viewing directions, laser range scanner functionality, etc.

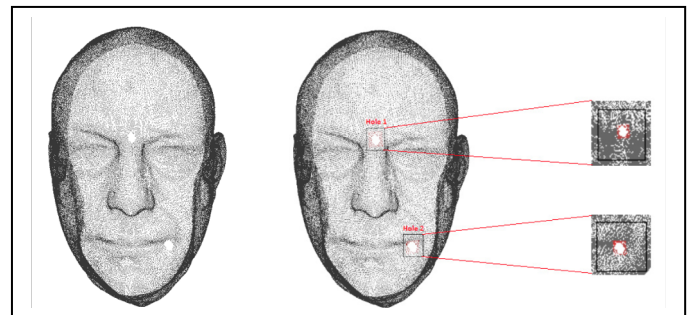


Figure 1: Holes in point clouds [22].

For example, some object portions may be absent because of accessibility or visibility issues, or because of unique physical characteristics of the scanned surface (transparency,

reflectivity, etc.). This produces holes in the dataset, which do not correspond to any holes in the object. If the point cloud's boundary and hole points are not properly processed, it will influence subsequent research, such as point set simplification [3], [4], point cloud registration [5], and hole repair [6]. It is worth mentioning that not all holes appearing in the estimated model are virtual, i.e. due to missing reconstructions of featureless surfaces [8]. A physical hole that is a part of the structure of the object being reconstructed may also be classified as a hole. Unfortunately, just looking at the point cloud, it is almost impossible to differentiate between these two types of holes (real and virtual). These real holes should be left as they are during the filling process. These days, a high-quality reconstruction of objects is an essential demand by many applications. Therefore, surface completion or hole-filling has grown in importance in the process of reconstructing 3D images. However, no hole filling can be applied without the detection of holes in point clouds. So, hole detection in the point cloud is an important component. Usually, 3D image reconstruction methods produce unstructured point clouds, in other words, the object's surface is not explicitly encoded with any connectivity information which makes the problem of the detection of holes on the surface an ill-defined problem [7]. In this paper, we introduce a statistical approach intended to identify holes within 3D point cloud data by using geometrical features in three-dimensional space. Our method relies on the principle of K-nearest neighbors (KNN) to identify neighboring points, followed by the computation of their average Euclidean distance. This average distance serves as a threshold, enabling the detection of irregularities in the point cloud. By comparing the distances between points, we can effectively pinpoint regions where data is missing, indicating the presence of holes.

II. LITERATURE REVIEW

In this section, we study some existing methods for the detection of the hole boundary in the 3D point cloud. As we know the detection of point cloud boundaries and hole points is a vital task. In the state of the art of boundary extracting methods, different methods have been studied and developed to extract the boundary. These methods fall into two categories: (1) The methods executed on the grid; and (2) The methods executed directly on the data points of the point cloud. The former methods triangulate the point cloud and look for the adjacent triangular meshes. If there is no adjacent triangle, the associated triangle is considered as the boundary of the hole [3]. Authors in [16] subdivided the methods on meshes into volume-based methods and surface-based methods. In an early study [8], a seed boundary is selected on the point cloud grid to search for the next boundary according to the half-edge data structure. When the closed loop is reached to complete the boundary detection of holes, the search is finished. Many studies have recently focused on the methods of direct hole detection, which do not need to triangulate the point cloud in advance and can save a lot of time. In [7] the authors computed boundary probability for each point and classified the points in the point set into boundary or interior points through the application of the angle criterion. The coherence of the points is leveraged to extract a boundary loop that represents a simple hole boundary. As the point cloud is usually unstructured, therefore, to enhance the efficiency of neighborhood searches kD-tree, octree, or BSP-tree can be utilized. Along with the angle criterion authors in [7] also presented some other criteria to determine whether a point is a boundary point or not i.e. half disc criterion, shape criterion, and various enhancements based on these criteria. The author in [13], finds the neighborhood of each point in the point cloud and approximates the direction of the normal for

detection of hole boundary in the point cloud. After finding the neighborhood of points, the author detected the candidate boundary points and created boundary polygons from boundary points. Following the detection of boundary points of the hole, an algebraic surface patch is fitted to the neighborhood and sample auxiliary points. Surfaces like spheres, cylinders, and planes are all recreated by this process. The author in [15] extracted the polygonal hole boundary with the Mean Shift approach to find out the vertices that have similar geometry properties with the hole region in the neighborhood of the boundary.

Authors in [14] use the Distance-Centroid technique, which employs maximum squared distance and K-Nearest neighbors to detect the boundary points. The points that do not belong in the hole boundary are filtered out using Statistical Outlier Removal (SoR). To cluster out the different holes, the author used region-growing segmentation. The author in [12] proposed a method based on triangular mesh models which aims to find solid holes in 3D models. The author grouped the interconnected coplanar triangles and extracted the contour of the model using the boundaries of the nearby planes. Then, the author uses the extracted contour to form several disjoint clusters of model vertices and detect holes by analyzing the relationship between the clusters and plane. The author in [17] also uses the triangular mesh for the detection of holes in a point cloud. The author uses boundary edge tracking to automatically identify holes, an edge is a boundary edge if it is part of only one triangle; otherwise, it is an interconnected edge that is part of multiple triangles. Authors in [18] extracted exterior boundary points of the surface S and further classified them into exterior boundary points and interior points. Then looping is done for all points of S , if each of them is not an exterior boundary point, and one of its neighbors (in 4-connectivity) is empty, it is determined as a boundary point of a hole. The boundaries detected by the author are shown in Figure 2. The methods for boundary extraction based on the computation of convex hull

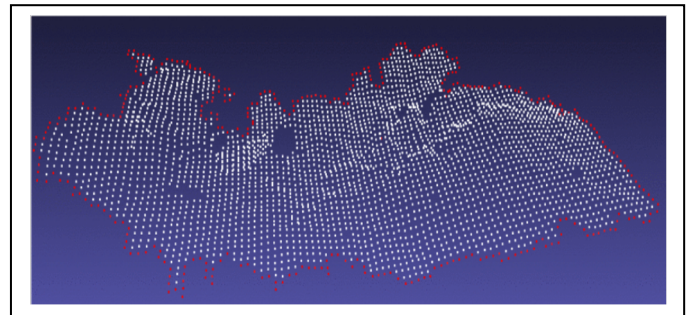


Figure 2: Detection of surficial boundary of the point cloud described in [18].

have also been widely applied. The author in [9] proposed a method that involves modifying the convex hull to detect the boundary of the point cloud. For every point on the boundary p_b , the computation is repeated to calculate the smallest angle between p_b and its neighboring points for finding the next boundary points, but this method is time-consuming. The author in [11] computed the convex hull of a group of points, starting from an initial core point and its neighbors, and referred to it as a cluster. Then, by repeating the procedure and using the convex hulls of the core points as new core points, this cluster is extended repeatedly until each convex hull of a core point is joined with the cluster's main body. When all exterior points have been identified and the final boundary has been reached, clusters will not be combined or expanded further. The author in [10] proposed a method to detect the boundary of a surface of 3D points set. The author triangulated the convex hull C of

the surface first and C is then optimized by computing and removing some outward triangles to obtain the exterior boundary of the surface. The author in [20] introduces a new method for identifying and fixing holes in point clouds. The author first calculated the density and 2D projection points of the point cloud. Then, the author used Euclidean distance to find the boundary points of the holes. It was followed by a preprocessing segmentation algorithm that helped to identify the boundary points more accurately. However, the author failed to detect the boundary of the whole point cloud simultaneously, so the method failed to detect the holes at boundary points.

III. METHODOLOGY

Our proposed methodology consists of two steps first we find K-Nearest neighbors and then we calculate Euclidean distances. These two steps are employed to compare each point within the point cloud. A threshold is settled that facilitates the identification of boundary and hole points. If the distance between points exceeds the predefined threshold, then the point is a potential boundary point, and if it is not then it is a normal point. The threshold has been set as the mean of measured distances between points in the point cloud.

A. Dataset

We have used the Kaggle digit dataset the “3D MNIST Dataset”. The dataset includes 3D point clouds, which are sets of (x, y, z) coordinates produced from about 5,000 images in the original 2D MNIST dataset [21]. The point clouds' maximum dimension range is one, and their mean is zero.

B. K-Nearest Neighbors (KNN)

K-nearest neighbors refer to k () points close to the query point in the spatial distance. K-nearest neighbors search can be described as; given a point set S containing n points, for query point p_i ($p_i \in S$) there is a subset set C containing k points (not including point p_i), $C \in S$, $k < n$ and for any $p_i \in C$, $p_2 \in S - C$, the following equation 1 can be met [19].

$$\text{dist}(p_i, p_1) \leq \text{dist}(p_i, p_2) \quad (1)$$

The k-nearest neighbors in two dimensions are shown in Figure 3, where the points are distributed in the two-dimensional space dispersedly, p is a query point and five points in the circle is the k (k=5) nearest neighbors.

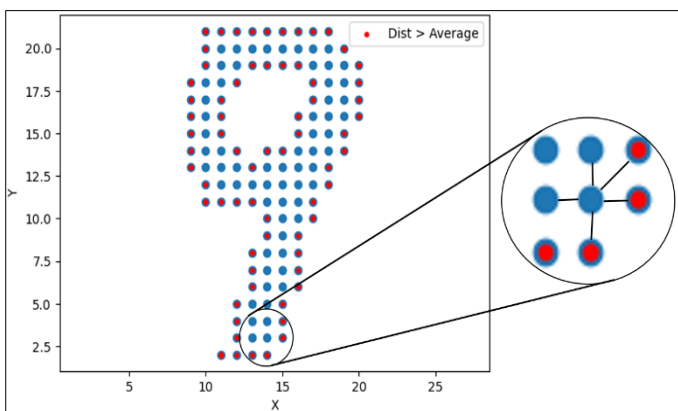


Figure 3: k-nearest neighbor in two-dimension.

When the number of points is small, k-nearest neighbors can be obtained by calculating the query point to all other points' Euclidean distance directly, sorting the distance values, and then taking the first k values as the k-nearest neighbors [19].

C. Euclidean distance

In our approach, we used KNN (K-Nearest Neighbors) for selecting the number of neighbors based on their Euclidean distances in 3D space. Next, we compute the distance between each point of the point cloud, identifying points with distances greater than this threshold as potential hole boundary points or point cloud boundaries. This process not only allows for effective hole detection but also provides insights into the overall structure and boundary of the point cloud. The Euclidean distances for 3-dimensional space are calculated in Equation 2.

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2} \quad (2)$$

Euclidean distance is applied to express the distance between two points on account of its convenience and simplicity to use. Moreover, it offers the advantage of requiring less computational power compared to more complex learning algorithms. We calculated the mean of the point cloud points while considering the mean of the K Nearest Neighbors (KNN). This involves computing distances between the points using KNN and then aggregating the mean distances of the neighbors. Subsequently, the average of these mean distances serves as the threshold, effectively separating points as hole or boundary points and normal points.

IV. RESULTS AND DISCUSSION

As previously discussed, various approaches have been employed for hole detection, each with its own set of advantages and limitations. However, our approach, as detailed in this paper, takes a unique path toward hole detection, primarily relying on statistical techniques such as Euclidean distances. The Euclidean distances between neighboring points are given in Table 1. The distances between 5 points concerning each other are calculated using equation 2 and are recorded in the table. A small value of Euclidean distance between the

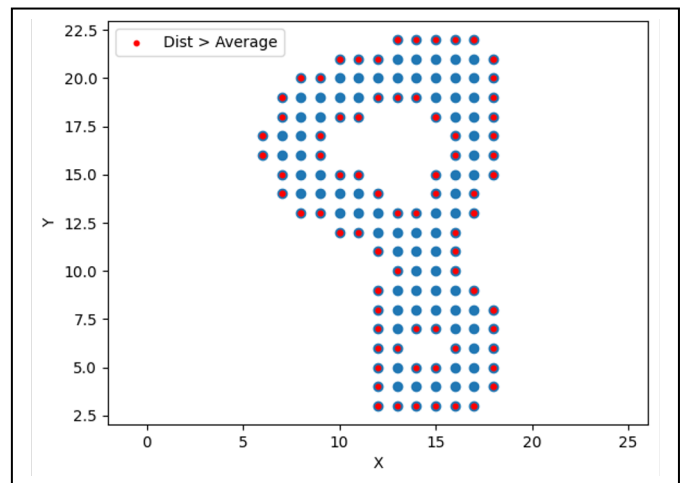


Figure 4: Boundary detection in the Kaggle digit dataset.

points means the points are near to each other. For example, the distance between N1 and N5 is 0.11 and the distance between N1 and N3 is 0.22 meaning that N5 is nearest to N1 as compared to N3. The results obtained from this technique have proven to be highly satisfactory, as illustrated in Figure 5. Through this simple approach, we have achieved effective hole detection and boundary delineation within the point cloud. In Figure 4 our proposed method is applied to the Kaggle digit dataset and the points in the point cloud have been classified. The blue points show normal points while the red points are considered boundary points. In Figure 5a donut-like point cloud is shown.

Before, all the points in the donut are red after looping through our method the points are classified as normal (blue) and boundary points (red) respectively as shown in Figure 5b.

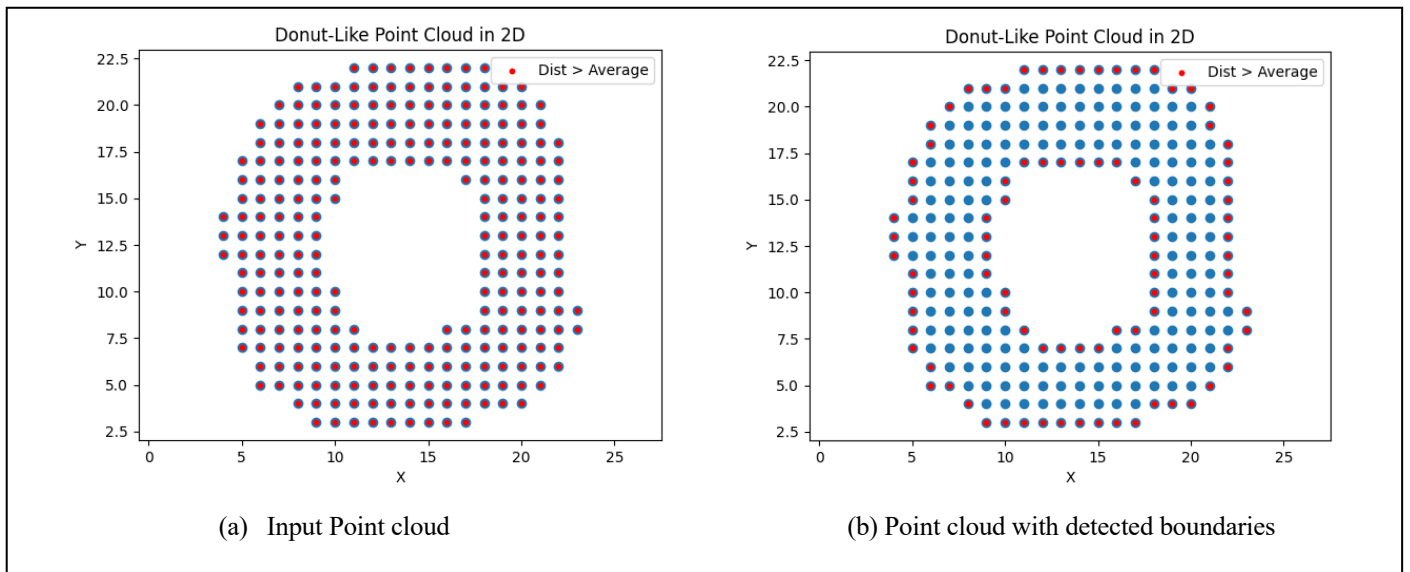


Figure 5: Hole boundary detection in point cloud

V. CONCLUSION

Hole detection in point clouds is a crucial initial step toward obtaining precise and comprehensive representations of the real surfaces. Our technique for the detection of holes is simple but effective and it takes a unique path towards hole detection, primarily relying on statistical techniques such as Euclidean

distances and K Nearest Neighbors (KNN). For now, our method is limited to the detection of hole boundaries and point cloud boundaries in surficial point clouds, but in future work, we go on to modify this statistical technique and make it robust for the accurate detection of holes in complex point clouds.

TABLE I. EUCLIDEAN DISTANCES BETWEEN NEIGHBORING POINTS (K = 5)

5- Nearest Neighbors	N1	N2	N3	N4	N5
N1	0	0.12	0.22	0.18	0.11
N2	0.12	0	0.13	0.17	0.14
N3	0.22	0.13	0	0.15	0.16
N4	0.18	0.17	0.15	0	0.15
N5	0.11	0.14	0.11	0.15	0

VI. REFERENCES

- [1] Y. Guo, F. Sohel, Bennamoun, J. Wan, and M. Lu, "A novel local surface feature for 3d object recognition under clutter and occlusion," *Inf. Sci.*, vol. 293, pp. 196–213, 2015.
- [2] G. Sansoni, M. Trebeschi, and F. Docchio, "State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation," *Sensors (Basel, Switzerland)*, vol. 9, pp. 568–601, 2009.
- [3] H. Song and H.-Y. Feng, "A progressive point cloud simplification algorithm with preserved sharp edge data," *The International Journal of Advanced Manufacturing Technology*, vol. 45, pp. 583–592, 2009.
- [4] H. Han, X. Han, F. Sun, and C. Huang, "Point cloud simplification with preserved edge based on normal vector," *Optik*, vol. 126, pp. 2157–2162, 2015.
- [5] J. Yang, Z. Cao, and Q. Zhang, "A fast and robust local descriptor for 3d point cloud registration," *Inf. Sci.*, vol. 346-347, pp. 163–179, 2016.
- [6] Y. Quinsat and C. Lartigue, "Filling holes in digitized point cloud using a morphing-based approach to preserve volume characteristics," *The International Journal of Advanced Manufacturing Technology*, vol. 81, pp. 411–421, 2015.
- [7] Bendels, G. H., Schnabel, R. and Klein, R., 2006. Detecting holes in point set surfaces.
- [8] Aldeeb, N.H. and Hellwich, O., 2017. Detection and Classification of Holes in Point Clouds. In *VISIGRAPP (6: VISAPP)* (pp. 321-330).
- [9] Sampath, A. and Shan, J., 2007. Building boundary tracing and regularization from airborne LiDAR point clouds. *Photogrammetric Engineering & Remote Sensing*, 73(7), pp.805-812.
- [10] Huang, X., Cheng, X., Zhang, F. and Gong, J., 2008. Side ratio constrain based precise boundary tracing algorithm for discrete point clouds.

- [11] International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 37, pp.349-354.
- [12] Suer, S., Kockara, S. and Mete, M., 2011, December. An improved border detection in dermoscopy images for density based clustering. In *BMC bioinformatics* (Vol. 12, pp. 1-10). BioMed Central
- [13] Wang, Y., Liu, R., Li, F., Endo, S., Baba, T. and Uehara, Y., 2012, August. An effective hole detection method for 3D models. In *2012 Proceedings of the 20th European signal processing conference (EUSIPCO)* (pp. 1940-1944). IEEE.
- [14] Chalmoviansky, Pavel & Jüttler, Bert. (2003). "Filling Holes in Point Clouds". 196-212. 10.1007/978-3-540-39422-8-14.
- [15] M J, Adarsh & Kotturshettar, Sunidhi. (2017). "Automatic Hole Detection for Selective Hole Filling in Point Cloud Data - Results".
- [16] Qiang, H., Shusheng, Z., Xiaoliang, B. and Xin, Z., 2010, October. Hole filling based on local surface approximation. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)* (Vol. 3, pp. V3-242). IEEE.
- [17] Guo, X., Xiao, J. and Wang, Y., 2018. A survey on algorithms of hole filling in 3D surface reconstruction. *The Visual Computer*, 34, pp.93-103.
- [18] Wang, J. and Oliveira, M.M., 2007. Filling holes on locally smooth surfaces reconstructed from point clouds. *Image and Vision Computing*, 25(1), pp.103-113.
- [19] V. S. Nguyen, T. H. Trinh and M. H. Tran, (2015) "Hole Boundary Detection of a Surface of 3D Point Clouds," *International Conference on Advanced Computing and Applications (ACOMP)*, Ho Chi Minh City, 2015, pp 124-129.
- [20] Li, D. and Wang, A., 2017, March. Improved KNN algorithm for scattered point cloud. In *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)* (pp. 1865-1869). IEEE.
- [21] Zhang, C. and Zhou, H., 2023. A method for identifying and repairing holes on the surface of unorganized point cloud. *Measurement*, 210, p.112575.
- [22] LeCun, Y., 1998. The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>.
- [23] Tabib, R.A., Jadhav, Y.V., Tegginkeri, S., Gani, K., Desai, C., Patil, U. and Mudenagudi, U., 2020. Learning-based hole detection in 3D point cloud towards hole filling. *Procedia Computer Science*, 171, pp.475-482.

CUSTOMER REVIEWS ANALYSIS PLATFORM BY CORRELATING SENTIMENT ANALYSIS AND TEXT CLUSTERING

Ehtisham ur Rehman
Dept. of Computer Science,
University of Engineering and Technology
(UET), Peshawar, Pakistan
ehtishamrehman@uetpeshawar.edu.pk

Najam Aziz
Dept. of Computer Systems Engineering,
University of Engineering and Technology
(UET),
Peshawar, Pakistan
najamkhattak@gmail.com

Nasir Ahmad
Dept. of Computer Systems Engineering,
University of Engineering and Technology
(UET),
Peshawar, Pakistan
n.ahmad@uetpeshawar.edu.pk

Abstract— Customer reviews and feedback are of paramount importance in the improvement cycle of any industry, product, or service. Formerly, product ratings were the basis for performance evaluation and key drivers of improvements. However, ratings were unable to depict the complete picture and were not adequate for an in-depth analysis of any product or service. Hence, customer reviews become the ultimate source of providing feedback for a specific detailed analysis as well as contributing to performance metrics. Although, customer reviews provide a very essential measure for performance evaluation, extracting important features and topics from the customer reviews has been challenging due to its unlabeled and variant nature. This paper focuses on extracting topics from customer review data and bringing in use the implicit knowledge for analytics. To extract topics and clusters from review data, unsupervised machine learning algorithms such as K-Means and Latent Dirichlet Allocation (LDA) are used. These topics are then correlated with sentiment analysis - score of positive or negative feedback - of each customer review. The products or services are then categorized with the help of the topics or domain they belong to alongside the sentiments. This provides a valuable analysis such as the score of positive, neutral, and negative feedback for each customer review input to new customers as well as product managers. This research work aims at using the hotel reviews dataset to categorize and rank hotels based on the different services captured in the text from customer reviews. The research work makes use of the hotel reviews dataset for categorizing and ranking hotels based on the different services discussed in the customer's reviews text. Moreover, this paper also provides a visualization of both text clustering algorithms depicting the topics in each cluster for an insightful analysis.

Keywords: Text Clustering, K-means, Latent Dirichlet Allocation Algorithm, Sentiment Analysis, Customer Reviews Feedback.

I. INTRODUCTION

The rise of e-commerce and online shopping platforms has exponentially enhanced the importance of customer reviews and feedback on these platforms. These reviews are not only a source of guidance for new potential buyers but also provide valuable input to the manufacturers or service providers for analyzing customer demands for new product designs as well as improving existing products and services [1]. Customer feedback is gathered in various forms on an online platform, it can be either in the form of ratings on a numeric scale or answering a certain set of questions about a specific product or service [2]. However, the most common is the written reviews from the customers and is most widely used for

analysis among business analysts for assessing the performance due to the different dimensions of the product discussed by customers in their reviews.

With this upsurge in customer reviews and feedback, different statistical and sentimental analysis approaches have been adopted to unfold the hidden insights in the data [3]. Sentimental Analysis focuses on extracting the underlying opinion within the text that can range from positive to negative values thus helping in identifying the sentiment and customer satisfaction about the product or service [4]. On the other hand, statistically, these reviews can help in identifying the most sold products about different quarters of the year as well as people's interest in different products geographically. Thus paving the way for different recommendation systems based on these statistics alongside prediction and machine learning algorithms [5].

Besides sentiment analysis, customer reviews also provide an opportunity to detect the dominant topics or dimensions discussed in these reviews text. These topics can be related to the price, quality, shipping, or any other service related to the product. But due to the unlabeled nature of the review datasets, unsupervised learning becomes the only option to identify dominant topics. Identifying dominant topics in the reviews will enable the manufacturers or service providers to specifically focus on those areas of concern for performance improvement. For this purpose, different unsupervised topic modeling techniques are available to cluster similar data in common groups and identify the most discussed topics in these groups [6].

In this research work, the aforementioned topic modeling techniques are combined with sentiment analysis to not only determine the opinion of the customer but also correlate it with the dominant category the reviewer is referring to. Moreover, the research work also compares the two topic modeling approaches commonly used, LDA and K-Mean, for better categorization of reviews. In addition, the paper evaluates the performance of the aforementioned two clustering algorithms through the evaluation matrix. Furthermore, the paper also provides a visualization of the identified dominant topics in both the clustering algorithms to enable a better comparison of the two techniques. In addition, sentiment analysis and topic modeling become the stepping-stone for conducting statistical analysis in the results Section 4 such as ranking products and services based on different topics alongside the review sentiment score.

The remaining Section of the paper is structured in the following way. Section 2 presents the related literature review as well as the background knowledge required to conduct the research. Section 3 of the paper describes the use case of the European hotel's dataset as well as the methodology applied to conduct the research. Section 4 discusses the results and graphs obtained by using different statistical analysis approaches as well as evaluate the two topic modeling techniques based on their performance matrices. Finally, Section 5 concludes the research work with the help of the results obtained as well as illustrating future research work prospects in this domain.

II. LITERATURE REVIEW

A. Related Work

Due to the utter importance of customer feedback, there has been an increasing focus among industries on analyzing and classifying customer reviews for attaining a competitive advantage over competitors in respective domains. Several research works had focused on extracting the opinion by applying sentiment analysis techniques while others have focused on using topic modeling techniques. Jeong et al. [7] had categorized reviews from Moscow's hotels in five specific attributes. They focus on several attributes such as amenities, experience, location, transactions, and value by applying regression analysis, etc. However, the research work did not extract dominant topics from the customer reviews to provide more in-depth analysis to the hotel owners.

In [8], Siew Hoong et al. have crawled review data of four-star and five-star hotels from Kuala Lumpur. This work focused on finding the most used predominant themes such as location, restaurant, comfort, etc. The topics discussed in the reviews are extracted through SAS text miner using the R language. However, the paper did not relate it to the sentiment described in each review rather it just finds the topic in each review. Authors have applied the Context-Based Keyword Pattern Cluster Analysis (CBKPC) using R-Mini to cluster similar topics together.

Moreover, in [9], Tian et al. analyzed the review dataset from four-star and five-star hotels from four cities of China. This work focused on finding the sentiment score i.e. positive, negative, and neutral as well as finding the customers' interest in categories related to food, staff, services, etc. They use natural language processing to clean text by removing stop words and unwanted words. Then it manually extracted categories by reviewing small parts of the dataset and applied sentiment analysis on the dataset. The authors also presented a correlation between the categories and frequency between the categories. However, the research work only focused on applying sentiment analysis to customer reviews. The research work in [9] is limited by the fact that it manually extracts categories that are not possible for the complete dataset.

Furthermore, in [10] Porntrakoon and Moemeng coupled sentiment analysis with different dimensions in the customer reviews text. The research work had focused on analyzing the sentiment or polarity of the three pre-defined dimensions in the

reviews that are pricing, product, and shipping by using multi-dimensional lexicon and sentiment compensation techniques. Although the framework designed in [10] finds the polarity of three different dimensions of the text of the reviews, it discards any other dimension except the predefined ones, and thus unable to analyze any unseen topic or dimension in the review text.

B. Text Clustering and Topic Modelling Algorithms

Topic modeling has been widely used in numerous domains ranging from text clustering to collaborative filtering, from information retrieval to image analysis [11] and relation extraction to inferring hidden insights from the data [12]. The topic Model Algorithm usually mines out the most dominant topics and semantic words from the text document [13]. Among the extracted topics similar ones are then clustered into the same groups. On the other hand, text clustering has a wide range of applications such as organizing and browsing documents, finding a coherent summary of the text documents collection, and document classification [14].

Although, topic modeling is often studied as a separate research area from text clustering, and it majorly focuses on determining latent topics from the text but it is one of the most widely used methods for probabilistic document clustering [15]. Probabilistic Latent Semantic Indexing (PLSI) and LDA are the two frequently used methods for topic modeling via probabilistic document clustering. The PLSI and LDA differ by the way term-document and topic-document probabilities are modeled. The probabilistic document clustering aims at creating a probabilistic generative model for the entire text document. Each document in probabilistic document clustering belongs to one of the k topics. The topic modeling algorithm usually mines out the most dominant topics and semantic words from the text document [16].

Besides the topic modeling algorithm, Fasheng Liu and Lu Xiong have analyzed and divided text clustering algorithms into different categories: hierarchical clustering, distance-based partitioned clustering, density-based algorithm, self-organizing maps algorithm on basis of several factors such as scalability, dimensionality, dependence on input parameters and ability to deal noise [17].

In this research work, LDA, a topic modeling method from the probabilistic document clustering algorithms, and K-means, a distance-based text clustering algorithm are implemented and evaluated based on their respective evaluation scores.

1) *Clustering*: Clustering techniques are one of the most widely recognized data analyses in the machine learning area and are used to get an instinct about the structure of the data. They can be defined as the task of identifying subgroups in the data such that data points in the same subgroups (cluster) are fundamentally the same as while data points in different clusters are different [18]. Particularly, we attempt to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similar measure, for example, Euclidean-based or relationship-based

separation. The decision of which similarity measure to utilize is application-specific.

2) *K-mean Algorithm*: The K-mean algorithm initially arbitrarily assigns a specified k number of clusters centers in space [18]. Afterward, each sample data point is assigned to these centers based on the nearest Euclidean distance between the data point and the cluster centers [19]. Then iteratively the center is recomputed from the mean of all the samples in the respective cluster and the process of assigning data points to the cluster is repeated until the cluster centers no longer move significantly [20]. The K-mean algorithm implementation cycle is depicted in Figure 1. The implementation cycle starts with data collection, and the much-needed text pre-processing.

The way the K-means algorithm works is as pursue:

- First, determine the number of clusters k.
- Initialize centroids by first rearranging the datasets and then randomly selecting k data points for the centroids without substitution.
- Continue the repeating process until there is no change to the centroids i.e. the assignment of data points to clusters isn't changing.
- Compute the sum of the squared distance between data points and all centroids.
- Refer each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of all data points that belong to each cluster.

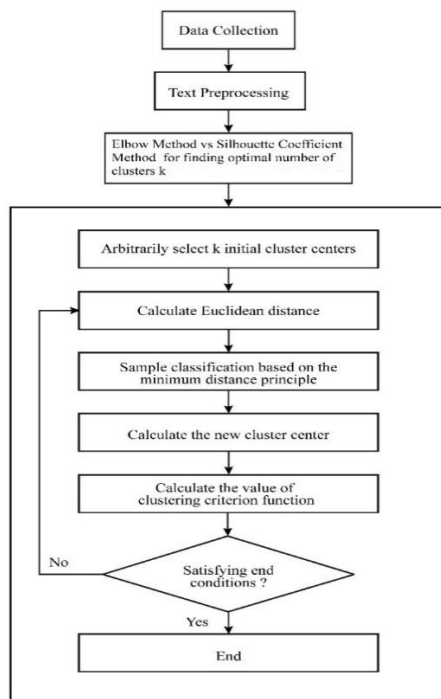


Figure 1: K-mean algorithm implementation cycle

The goal of the k-means algorithm is to reduce the objective function, which is called the squared error function. Equation 1 gives the objective function (J).

$$J = \sum_{j=1}^K \sum_{i=1}^n ||x_i^{(j)} - C_j||^2 \quad (1)$$

The K-mean algorithm distributes the n number of cases into k number of clusters as shown in Figure 2 that are predefined. Whereas Euclidean distance between a case and the centroid. However, to get the appropriate number of K for optimized implementation of the K-mean algorithm silhouette method is used [21].

The K-mean algorithm distributes the n number of cases into k number of clusters as shown in Figure 2 that are predefined. Whereas $||x_i^{(j)} - C_j||^2$ is a Euclidean distance between $x_i^{(j)}$ and the centroid C_j . However, to get the appropriate number of K for optimized implementation of the K-mean algorithm silhouette method is used [21].

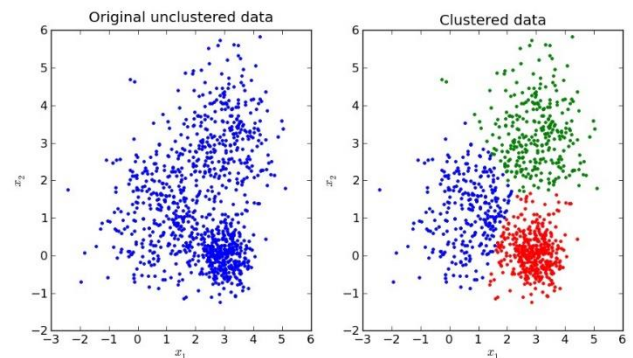


Figure 2: K-mean data distribution in K number of clusters

3) Latent Dirichlet Allocation

In natural language, processing, latent Dirichlet allocation is a generative factual model that enables sets of perceptions to be clarified by unobserved groups that clarify why a few parts of the data/information are comparable.

Topic modeling using LDA (Latent Dirichlet allocation) allows us to extract the hidden topic themes from a large dataset, using LDA a mixture of topics from a large number of documents can be found [22][23]. A defined number of topics is assumed within the document or set of reviews by LDA [24]. LDA enables to process of the frequency of each topic based on the occurrence of it in the dataset [22]. LDA extracts the hidden structure of topics from the data by applying a probabilistic approach. Latent Dirichlet Allocation works on the Bayesian estimation framework to find out the theme topics from the dataset [7]. Latent Dirichlet Allocation is the unsupervised probabilistic model that takes a bag of words i.e. corpus dictionary as input [25]. LDA filter out topics from the document and then find words for topics. The working of LDA is given in Equation 2.

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^N \sum_{z_n=1}^k p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta$$

The main idea of LDA is, the documents are expressed as random mixtures over latent topics, where each topic is categorized by distribution over words [26]. Equation 2 comprises several parameters. The k is the number of topics, and are corpus level parameters and the variables are document-level variables. The variables are word-level variables and are sampled once for each word in each document. LDA involves three levels of evaluation, through which the topic node is sample repeatedly within the document. Under the LDA model, multiple topics associated with the document are established.

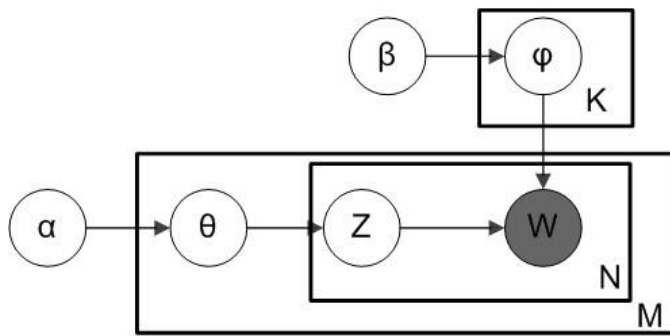


Figure 3: LDA (Latent Dirichlet Allocation)

LDA groups the words in a document based on the frequency and its association with the topics [25][27]. Once the LDA model is trained on data, dominant topics are extracted and categorized. The use case of hotel review data is a model using LDA and the dominant topics are categorized such as hotel services, staff, transactions, location, and restaurant.

C. Sentiment Analysis

The sentiment is an emotion or attitude prompted by the feelings of the customer. Sentiment analysis is also known as opinion mining, which evaluates people's opinions towards any topic, product, or service. Sentiment analysis is an emerging domain in the area of research in natural language processing (NLP) and has gained attention in recent years in text mining and text classification. It is a machine learning approach to analyze and classify the user's comments, emotions, opinions, and attitudes based on the polarity of the text. Sentiment analysis plays an important role in depicting and finding people's opinions in politics, e-businesses, and the general social trend towards an ongoing issue. In politics, sentiment analysis is used for forecasting the outcomes of political trends in the region and predicting poll results [25]. Whereas in business it is widely used to analyze and predict stock market trends. Sentiment Analysis has gained more importance in online applications, e-commerce, and in social media platforms i.e. blogs, Twitter, online website stores,

discussion forums which attracts the customers, organization firms, and stakeholder to do analysis on data from online websites and to extract meaningful information from the data [28] [29]. The main aim of sentiment analysis is to analyze the user opinions and classify that whether it falls in which category i.e. negative sentiment, positive sentiment, and Neutral sentiment.

III. IMPLEMENTATION

This section describes the implementation process of text clustering algorithms, LDA and K-means, alongside sentiment analysis to conduct in-depth analysis. The implementation will be executed in the steps shown in Figure 4. The approach used is generic and is applicable for analysis of any review's dataset in any business domain. This paper analyses the European Hotels review dataset.

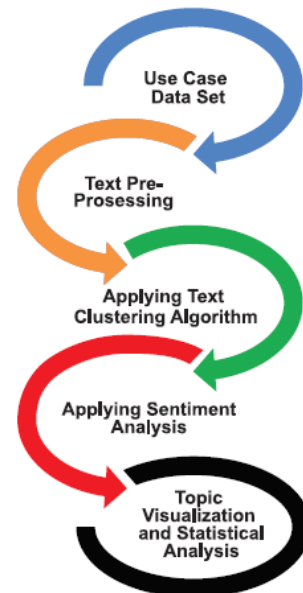


Figure 4: Implementation Execution Steps

A. Use Case Dataset

The dataset contains 515k reviews record of 13 European hotels and has been sourced from an online website of datasets called Kaggle. The dataset contains several attributes as shown in Figure 5 such as hotel address, hotel name, review date, average review score, the total number of reviews, reviewer nationality, negative review, positive review, etc. Each of the customer reviews provides its textual description of their personal opinion or experience and ratings regarding the hotel services acquired by the customer. The review text in the dataset is unlabeled, thus unsupervised text clustering algorithms are applied to categorize the reviews. The review text is mostly unclean and had a lot of unwanted words that are removed in the data preprocessing stage.

Hotel Review Dataset

```
--Hotel_Name: string (nullable = true)
--Hotel_Address: string (nullable = true)
--Review_Date: string (nullable = true)
--Negative_Review: string (nullable = true)
--Positive_Review: string (nullable = true)
--Reviewer_Nationality: string (nullable = true)
--Average_Score: string (nullable = true)
--Reviewer_Score: string (nullable = true)
--Total_Number_of_Reviews: string (nullable = true)
```

Figure 5: Available Attributes Column in the Dataset

B. Text Preprocessing

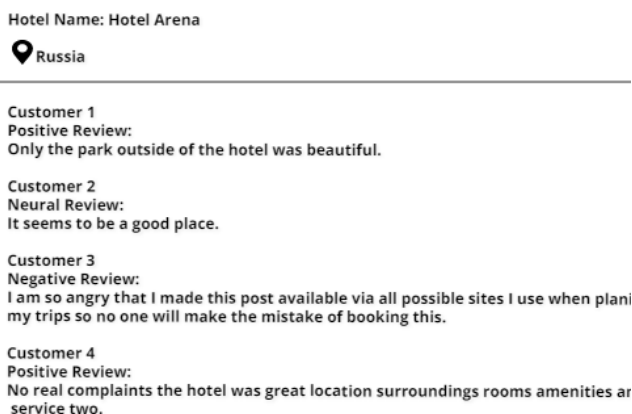
Text Pre-Processing is the conventional method used for natural language processing tasks. It performs the transformation of text into a more digestible form so that the machine learning algorithms can use it for better performance and having magnificent efficiency.

The raw data is pre-processed before applying the machine-learning algorithm. The pre-processing includes cleaning data using natural language techniques, such as tokenization, and removing words that have fewer than 3 characters. Moreover, after removing unwanted words, the remaining words are lemmatized and stemmed.

1) Text Pre-Processing

Tokenization is a process of chopping long sentences or text streams into words and phrases by removing unwanted characters such as empty spaces and punctuations. Every token is made of a word from the first character to the last character [30].

In this research, we take customer reviews i.e. positive, negative, and neural as raw data for tokenization as shown in Figure 6, in which we convert the customer reviews into individual words because the python programing language doesn't understand any distinction between words, they are just a stream of characters which are all same.



Hotel Name: Hotel Arena
Russia

Customer 1
Positive Review:
Only the park outside of the hotel was beautiful.

Customer 2
Neural Review:
It seems to be a good place.

Customer 3
Negative Review:
I am so angry that I made this post available via all possible sites I use when plani my trips so no one will make the mistake of booking this.

Customer 4
Positive Review:
No real complaints the hotel was great location surroundings rooms amenities ar service two.

Figure 6: Different Customers Reviews Data

2) Removal stop words

The review text data consist of excessively common words such as "of", "be", "but", "on", and other common words.

Although these stop words have a very small weight due to their frequent occurrences these words have a very high frequency [30].

We only take the important keywords from the desired topics i.e. comfort, staff, location rooms, etc. as shown in Table 1, and convey the context that is present in the review. Therefore, these "I am", and "In there" are present more frequently in many reviews and these are just connectors, and there doesn't add too much to the context that the machine wants to understand. In this procedure, we get rid of these stop words or punctuation used by the customers in their reviews.

3) Stemming and Lemmatization

Although stemming and lemmatization are used for transforming words to their original base form but both these methods work differently [31]. Stemming tries to achieve it by removing the suffixes and prefixes, but that may sometimes lead to incorrect word forms. So to counter the limitation posed by stemming, lemmatization is done which takes into account the morphological analysis of the words [31]. However, lemmatization needs a dictionary, which it can consult to connect the form of the word to its root. After the text, pre-processing the dictionary is created from pre-processed data. A Corpus dictionary is created from the pre-processed dictionary, which reports the frequency of words appearing in the dictionary showing the number of occurrences of a word in each document.

C. Determine the optimal number of K for K -mean, Silhouette Score, and LDA

1) K -Means

K -means is an unsupervised machine learning algorithm for clustering large datasets. K -means groups the data into different clusters based on Euclidean distance. The optimal number of the cluster for the K -Mean algorithm is evaluated using the Elbow method. Although a manual selection of several clusters k is accurate it comes extensive computation overhead to analyze each k . On the contrary, the elbow method is employed to find the optimal number of clusters for the K -means algorithm by varying k stepwise i.e. starting k at 1 to 10, and calculating the accuracy and computational cost on the training of model [32]. The Elbow method computes the total sum of square error within the cluster in each iteration [33] [34]. After computing the sum of squares, the elbow method plot the curve according to the number of clusters k . The blend knee plot will show the optimal number of clusters k as shown in Figure 7.

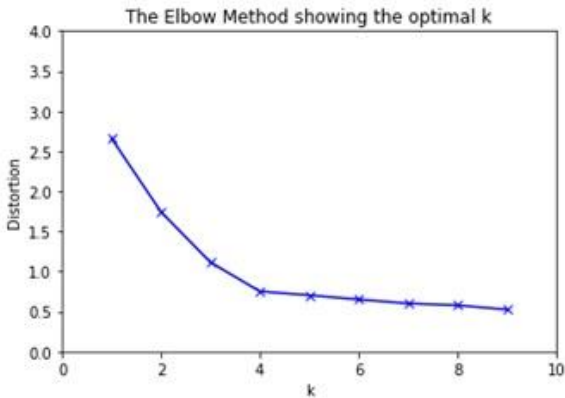


Figure 7: Elbow Method to find optimal K for the K-mean algorithm

2) Silhouette Score

Silhouette is the better measure to choose the number of clusters to be defined from the data [35]. It is determined for each instance and the Equation 3 goes like this:

$$\text{Silhouette Coefficient} = \frac{(a - b)}{\max(a, b)} \quad (3)$$

Where b is the mean intra-cluster distance: mean distance to the other instances in the same cluster. a depicts mean closest cluster distance i.e. mean distance to the instance of the following nearest cluster.

The coefficient varies between -1 and 1. A value near 1 implies that the instance is near to its cluster is a part of the correct cluster. While a value near -1 means that the value is allocated to the inappropriate cluster.

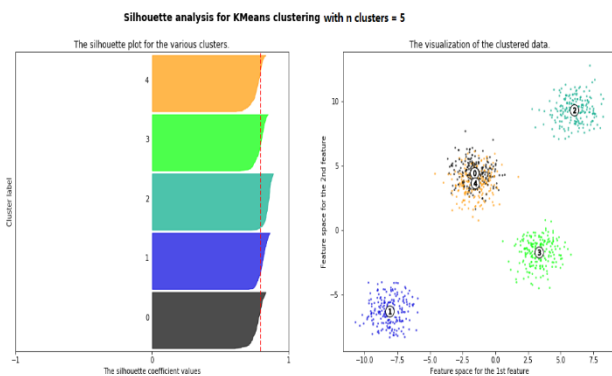


Figure 8: Silhoutter Method

According to strategy, $k=5$ should be chosen for the number of clusters. This method is better as it decides the optimal number of clusters more significant and clearer. But this metric is computation expensive as the coefficient is calculated for every instance. Therefore, the decision regarding the optimal metric to be chosen for the number of cluster decisions is to be made according to the requirements of the product.

In Figure.8, the range of n clusters is 2 to 11, and the thickness of the silhouette plot the cluster size can be visualized. The silhouette plot for cluster 0 when n clusters are equal to 2, is bigger owing to the grouping of the 3 sub-clusters into one big cluster. However, when the n clusters are equal to 5, all the plots have a similar thickness. The average silhouette score with several clusters is given below in Figure 8, in which at cluster 5 the graph bends at 0.7958870 value.

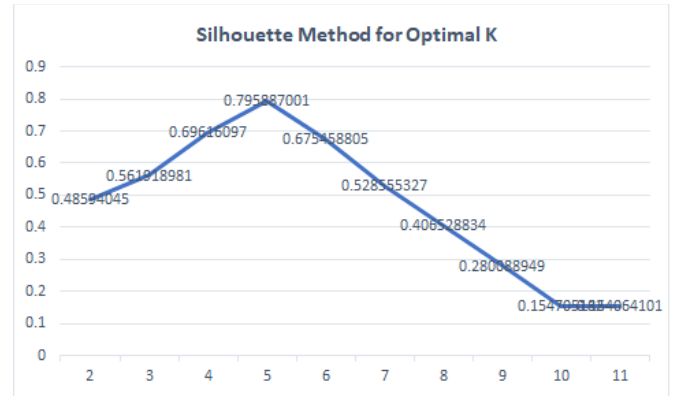


Figure 8: Average Silhouette Score

For implementing the LDA algorithm, the value of optimal k was determined using the log-likelihood and perplexity scores. Higher the log-likelihood score and the smaller the perplexity score, the better is the model [36]. To find the highest value of log-likelihood and smallest value of perplexity, a different value of k with a different set of learning decays was tested. Learning decay rate defines the change in weights during training. The learning rate controls how quickly the model is adapted, and thus for large learning decays the changes are more rapid whereas for smaller learning rates the changes are smaller with each update thus requires more training epochs in comparison to large learning rates. The optimal value of k is determined from six different values of k, which are 5, 10, 15, 20, 25, and 30, by varying three different values of learning decay i.e. 0.5, 0.7, and 0.9. Figure 9 depicts the optimal value for $k=5$ and learning decay = 0.7 when the log-likelihood has the largest value i.e. -19578767.67.

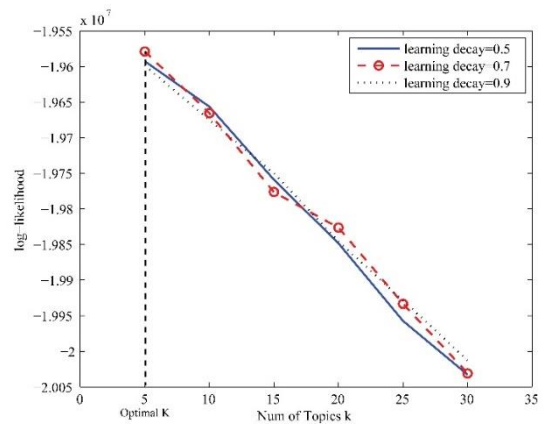


Figure 9: Log-likelihood vs Num of topics graph for finding optimal value of k for LDA algorithm

D. Sentiment Analysis via TextBlob

TextBlob is a python library used for textual processing of raw data and can be used efficiently for sentiment analysis. TextBlob allows us to perform common operations on textual data along with natural language processing tasks such as Noun phase extraction, parsing, n-grams, tokenization, classification, sentiment analysis, and much more. NLP operations are accessed by calling the TextBlob API [37]. In TextBlob sentiment analysis property called sentiment, returns two properties polarity and subjectivity of text. Polarity has its score range which starts from $[-1.0, 1.0]$ float score, with -1 being the extremely negative and 1 being extremely positive. Reviews, whose sentiment score is less than zero are termed negative reviews, while reviews that have sentiment scores greater than 0.25 are labeled as positive reviews. Moreover, reviews that fall in the range of 0 to 0.25 are referred to as neutral reviews. Subjectivity has also a range of float scores from $[0, 1]$ refers to personal opinion emotion or factual information.

IV. RESULTS

This Section illustrates and visualizes the list of topics and keywords extracted from the reviews with the help of machine learning algorithms. The Section also discusses results obtained from performing statistical analysis performed on the use case dataset of European hotels with help of topic categorization and sentiment analysis. Moreover, in this Section, the two topic modeling algorithms are also evaluated based on their respective performance metrics.

A. Extracted Topics and Keywords

The topic-modeling algorithm mentioned in the implementation Section created five different clusters with a different set of keywords. The topics discussed in these five clusters differ from each other in the context inferred from the keywords. The top 10 keywords with the highest frequency in each of the clusters were selected to infer the context and for labeling each of the topics. The five different topics inferred from the keywords are comfort and transaction, staff behavior and services, location and accessibility, room and hotel facilities and decor, and finally, food and restaurants. Table 1 shows the top 10 keywords in each of the topic clusters along with their labels.

Table 1: Most important keywords extracted for each topic

Comfort and Transaction (Topic 1)	Staff Behaviour and Service (Topic 2)	Location and Accessibility (Topic 3)	Room, Hotel Facilities, and Decor (Topic 4)	Food and Restaurants (Topic 5)
comfort	staff	location	room	breakfast
hotel	friendly	position	clean	food
upgrade	service	station	view	restaurant
room	clean	close	specious	coffee

free	helpful	metro	bathroom	buffet
extra	breakfast	city center	modern	service
timing	polite	tram	décor	quality
chec-in/out	professional	transport	pool	delicious
welcome	welcome	park	terrace	price
stay	comfort	access	bed	staff

The distance between each topic and the frequency of each word in the topic can best be visualized with LDAvis [38], a visualization library available in python. LDAvis is a web-based interface specifically designed for LDA algorithm topics visualization. It aims at providing a better understanding and deep inspection of the topics generated from the LDA algorithm by illustrating the meaning of each topic, the prevalence of each topic, and the correlation between different topics. In Error! Reference source not found., a visualization of a European hotel dataset with five different clusters is shown, which is produced through the LDAvis library. The visualization in Figure 10 is divided into two parts, the left panel of the visualization shows the five clusters with five circles plotted in a two-dimensional plane. The distance between the centers represents the inter-topic distance thus showing the correlation between different topics. The area of each cluster circle represents the prevalence of that cluster topic, the larger the area of a cluster circle the more is its prevalence and vice versa [36].

The second part of the visualization in Figure 10 is the right panel with a horizontal bar chart of individual terms. The bar charts represent the most relevant terms and frequency of each term in the selected topic on the left panel thus helping in interpreting the meaning of each topic. The bar charts show the top 30 terms of each topic extracted with help of LDA and then ranked based on estimated frequency and relevance in the selected topic thus providing a clear picture of each topic and its correlation with other topics.

Besides the LDA topic modeling algorithm, the K-means algorithm is also used to cluster the data. Figure 11 shows the different clusters made by the K-means algorithm. The optimal number of k is determined with help of the Silhouette method as described in Implementation Section 3. Although attributes like simplicity and fast convergence make the K-means algorithm a viable option for clustering data, the initial arbitrarily assigned cluster centers significantly hampers its performance [39]. Due to this random selection, the algorithm may converge to locally optimal solutions. Therefore, this paper makes use of the topics extracted via the LDA algorithm for statistical analysis in the next Section due to its more realistic results than k -mean for the topic assignment.

surrounding facilities available such as environment, accessibility from bus stops and airports, location of the nearby markets, historical sites, city center, and restaurants. Among the 81323 reviews, 40774 have sentiment scores greater than 0.25, hence they are termed as positive reviews. Whereas, 31585 reviews are neutral and 8964 people have negative opinions about the surrounding facilities of the hotels.

4) Room and Hotel Facilities and Decor

The Room and hotel Facilities and decor cluster highlight the facilities and interior of the room as well as the overall hotel. The terms discussed in this group include room decor, view from the room, floor and roof design, availability of pool and terrace area in the hotel, spaciousness of rooms and bathroom. There are 104945 (20.35%) reviews that discuss the overall services of the rather than one specific topic or service of the hotel. Among which 31646 are positive reviews, 37358 have sentiment score in the neutral range and 35941 are negative reviews as depicted in Figure 12.

5) Food and Restaurants

This cluster of reviews has 94005 (18.23%) reviews, which majorly discusses the food services available such as breakfast, buffet, and drinks. Moreover, it talks about the quality and different variety of foods and drinks (coffee and tea) served in the restaurants inside the hotels as well as the distance from the famous food outlets nearby the hotel they reside in. Across the total reviews belonging to this cluster, 45914 reviews are positive opinions about the food and restaurants, while 36385 reviews have sentiment scores between 0 and 0.25 thus termed as neutral and 16989 reviews belong to the negative category.

Figure 14 is the illustration of the distribution of the sentiment score of reviews for each topic through Cumulative Distribution Function (CDF) graphs. The CDF plots presented in Figure 14 represent the empirical cumulative distribution function of the sentiment score. The sentiment labels (positive, neutral, and negative) on the y-axis depict the percentage of customer sentiments for each topic. The ranges are defined by the dotted line in Figure 14. CDF plots are useful for comparing the distribution of different ranges of data.

The graphs are marked with the threshold of each category namely; Positive, neutral, and negative sentiment scores. The first five graphs show the distribution of sentiment scores of each topic generated from the topic modeling algorithms. While the last graph is the distribution of combined scores of all the customer reviews. The paper aims to provide a visualization tool for customers, product managers, and business decision-makers to extract topics from any customer review topic data and visualize the corresponding sentiments and topics using the CDF plots.

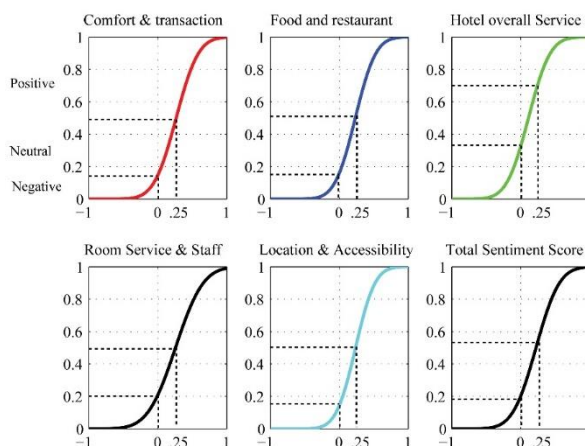


Figure 14: CDF plots of each service's sentiment score distribution

V. CONCLUSION

This paper provided a novel approach to sentiment analysis based on topic extraction. This helps in utilizing implicit knowledge for analytics and useful decision-making. The unsupervised machine learning algorithms such as K-Means and Latent Dirichlet Allocation (LDA) for clustering and topic modeling were employed. The proposed approach has the potential of being a valuable analytic tool for new customers as well as product managers. The research work used the hotel reviews dataset as a use case to evaluate the proposed approach. The hotel review dataset was categorized and ranked hotels based on the different services discussed in the customer reviews text.

In the future, we aim to provide a more general framework by leveraging advanced machine learning tools such as deep learning to the review's dataset. This will help in extracting topics from the review's dataset including hotels, airlines, places to visit, clinics, hospitals, and many more that will potentially assist customers and travelers in useful decision making.

Nomenclatures

$x_i^{(j)}$	Case
C_j	Centroid
θ_d	Document-level variables
z_n and w	Word level variables
J	Objective function
K	Optimized implementation of K-mean algorithm
k	Number of clusters
n	Range of clusters

Greek Symbols

α	Corpus level parameters
β	Corpus level parameters

Abbreviations

JESTE	Journal of Engineering Science and Technology
C	
LDA	Latent Dirichlet Allocation
NLP	Natural Language Processing
CDF	Cumulative Distribution Function
PLSI	Probabilistic Latent Semantic Indexing
CBKP	Context-Based Keyword Pattern Cluster Analysis
C	National Center in Big Data and Cloud Computing
NCBC	

References

- [1] Z. Singla, S. Randhawa, and S. Jain, "Statistical and sentiment analysis of consumer product reviews," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Delhi, Jul. 2017, pp. 1–6. doi: 10.1109/ICCCNT.2017.8203960.
- [2] K. L. S. Kumar, J. Desai, and J. Majumdar, "Opinion mining and sentiment analysis on online customer review," in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIIC)*, Chennai, Dec. 2016, pp. 1–4. doi: 10.1109/ICCIIC.2016.7919584.
- [3] H. Zhang, A. Sekhari, F. Fourli-Kartsouni, Y. Ouzrout, and A. Bouras, "Customer Reviews Analysis Based on Information Extraction Approaches," in *Product Lifecycle Management in the Era of Internet of Things*, vol. 467, A. Bouras, B. Eynard, S. Fougou, and K.-D. Thoben, Eds. Cham: Springer International Publishing, 2016, pp. 227–237. doi: 10.1007/978-3-319-33111-9_21.
- [4] Y. Woldemariam, "Sentiment analysis in a cross-media analysis framework," in *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, Hangzhou, China, Mar. 2016, pp. 1–5. doi: 10.1109/ICBDA.2016.7509790.
- [5] Y. Saito and V. Klyuev, "Classifying User Reviews at Sentence and Review Levels Utilizing Naïve Bayes," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, PyeongChang

Kwangwoon_Do, Korea (South), Feb. 2019, pp. 681–685. doi: 10.23919/ICACT.2019.8702039.

- [6] P. S. Hiremath, S. P. Algur, and S. Shivashankar, "Cluster Analysis of Customer Reviews Extracted from Web Pages," no. 9, p. 8.
- [7] H. Jeong, S. Mankad, N. Gavirneni, and R. Verma, "Text Analytics of Online Customer Reviews," vol. 16, no. 2, p. 19, 2016.
- [8] A. S. H. Lee, Z. Yusoff, and Z. Zainol, "Know your Hotels Well! an Online Review Analysis using Text Analytics," *International Journal of Engineering*, p. 7.
- [9] X. Tian, W. He, R. Tao, and V. Akula, "Mining Online Hotel Reviews: A Case Study from Hotels in China," p. 8.
- [10] P. Porntrakoon and C. Moemeng, "Thai Sentiment Analysis for Consumer's Review in Multiple Dimensions Using Sentiment Compensation Technique (SenseComp)," in *2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Chiang Rai, Thailand, Jul. 2018, pp. 25–28. doi: 10.1109/ECTICon.2018.8619892.
- [11] T. Iwata, T. Hirao, and N. Ueda, "Topic Models for Unsupervised Cluster Matching," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 4, pp. 786–795, Apr. 2018, doi: 10.1109/TKDE.2017.2778720.
- [12] M. Allahyari and K. Kochut, "Discovering Coherent Topics with Entity Topic Models," in *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Omaha, NE, USA, Oct. 2016, pp. 26–33. doi: 10.1109/WI.2016.0015.
- [13] B. Wang, Y. Liu, Z. Liu, M. Li, and M. Qi, "Topic selection in latent dirichlet allocation," in *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Xiamen, China, Aug. 2014, pp. 756–760. doi: 10.1109/FSKD.2014.6980931.
- [14] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, "On feature distributional clustering for text categorization," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '01*, New Orleans, Louisiana, United States, 2001, pp. 146–153. doi: 10.1145/383952.383976.
- [15] C. C. Aggarwal and C. Zhai, Eds., *Mining Text Data*. Boston, MA: Springer US, 2012. doi: 10.1007/978-1-4614-3223-4.
- [16] M. Alhawarat and M. Hegazi, "Revisiting K-Means and Topic Modeling, a Comparison Study to Cluster Arabic Documents," *IEEE Access*, vol. 6, pp. 42740–42749, 2018, doi: 10.1109/ACCESS.2018.2852648.
- [17] Fasheng Liu and Lu Xiong, "Survey on text clustering algorithm," in *2011 IEEE 2nd International Conference on Software Engineering and Service Science*, Beijing, China, Jul. 2011, pp. 901–904. doi: 10.1109/ICSESS.2011.5982485.
- [18] D. A. S. Ramkumar and R. Nethravathy, "TEXT DOCUMENT CLUSTERING USING K-MEANS ALGORITHM," vol. 06, no. 06, p. 5, 2019.
- [19] C. Xiong, Z. Hua, K. Lv, and X. Li, "An Improved K-means Text Clustering Algorithm by Optimizing Initial

- Cluster Centers,” in *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, Macau, China, Nov. 2016, pp. 265–268. doi: 10.1109/CCBD.2016.059.
- [20] S. Lu *et al.*, “Clustering Method of Raw Meal Composition Based on PCA and Kmeans,” in *2018 37th Chinese Control Conference (CCC)*, Wuhan, Jul. 2018, pp. 9007–9010. doi: 10.23919/ChiCC.2018.8482823.
- [21] M. Bertin and I. Atanassova, “K-means and Hierarchical Clustering Method to Improve our Understanding of Citation Contexts,” p. 6.
- [22] Y. Guo, S. J. Barnes, and Q. Jia, “Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation,” *Tourism Management*, vol. 59, pp. 467–483, Apr. 2017, doi: 10.1016/j.tourman.2016.09.009.
- [23] A. Kelaiaia and H. Merouani, “Clustering with Probabilistic Topic Models on Arabic Texts: A Comparative Study of LDA and K-Means,” vol. 13, no. 2, p. 7, 2016.
- [24] J. Büschken and G. M. Allenby, “Sentence-Based Text Analysis for Customer Reviews,” *Marketing Science*, vol. 35, no. 6, pp. 953–975, Nov. 2016, doi: 10.1287/mksc.2016.0993.
- [25] D. M. Blei, “Latent Dirichlet Allocation,” p. 30.
- [26] C. Sievert and K. Shirley, “LDAvis: A method for visualizing and interpreting topics,” in *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*, Baltimore, Maryland, USA, 2014, pp. 63–70. doi: 10.3115/v1/W14-3110.
- [27] D. Kozłowski, V. Semeshenko, and A. Molinari, “Latent Dirichlet allocation model for world trade analysis,” *PLoS ONE*, vol. 16, no. 2, p. e0245393, Feb. 2021, doi: 10.1371/journal.pone.0245393.
- [28] A. Malik, D. Kapoor, and A. P. Singh, “Sentiment Analysis on Political Tweets,” p. 3, 2016.
- [29] W. Wang, “Sentiment analysis of online product reviews with Semi-supervised topic sentiment mixture model,” in *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, Yantai, China, Aug. 2010, pp. 2385–2389. doi: 10.1109/FSKD.2010.5569528.
- [30] B. Saberi and S. Saad, “Sentiment Analysis or Opinion Mining: A Review,” p. 7.
- [31] L. M. Abualigah, A. T. Khader, and M. A. Al-Betar, “Unsupervised feature selection technique based on genetic algorithm for improving the Text Clustering,” in *2016 7th International Conference on Computer Science and Information Technology (CSIT)*, Amman, Jordan, Jul. 2016, pp. 1–6. doi: 10.1109/CSIT.2016.7549453.
- [32] P. Han, S. Shen, D. Wang, and Y. Liu, “The influence of word normalization in English document clustering,” in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, Zhangjiajie, China, May 2012, pp. 116–120. doi: 10.1109/CSAE.2012.6272740.
- [33] P. Bholowalia, “EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN,” *International Journal of Computer Applications*, vol. 105, no. 9, p. 8.
- [34] S. Tripathi, A. Bhardwaj, and P. E, “Approaches to Clustering in Customer Segmentation,” *IJET*, vol. 7, no. 3.12, p. 802, Jul. 2018, doi: 10.14419/ijet.v7i3.12.16505.
- [35] V. Divya and K. N. Devi, “An Efficient Approach to Determine Number of Clusters Using Principal Component Analysis,” in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, Coimbatore, Mar. 2018, pp. 1–6. doi: 10.1109/ICCTCT.2018.8551182.
- [36] H. Jelodar *et al.*, “Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey,” *Multimed Tools Appl*, vol. 78, no. 11, pp. 15169–15211, Jun. 2019, doi: 10.1007/s11042-018-6894-4.
- [37] Z. J. Wang, D. Choi, S. Xu, and D. Yang, “Putting Humans in the Natural Language Processing Loop: A Survey,” *arXiv:2103.04044 [cs]*, Mar. 2021, Accessed: May 22, 2021. [Online]. Available: <http://arxiv.org/abs/2103.04044>
- [38] D. Marutho, S. Hendra Handaka, E. Wijaya, and Muljono, “The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News,” in *2018 International Seminar on Application for Technology of Information and Communication*, Semarang, Sep. 2018, pp. 533–538. doi: 10.1109/ISEMANTIC.2018.8549751.
- [39] H. Wu, L.-Y. Wei, X. Wang, and B. Guo, *Silhouette Texture*. Tomas Akenine-Möller and Wolfgang Heidrich, 2006.

A Deep Learning based Mobile Application for Wheat Disease Diagnosis

Sarmad Riaz

Department of CS & IT, University of
Engineering & Technology, Peshawar
25000, Pakistan
sarmadriaz385@gmail.com

Raja Taimour

Department of CS & IT, University
of Engineering & Technology,
Peshawar 25000, Pakistan
rajataimour794@gmail.com

Mashab Ali Javed

Department of computer Systems
Engineering, Sir Syed CASE institute
of technology Islamabad Pakistan,
Jd.mashab@gmail.com

Amaad Khalil

Department of Computer Systems
Engineering, University of
Engineering & Technology,
Peshawar 25000, Pakistan
amaadkhalil@uetpeshawar.edu.pk

Yasir Saleem Afridi

Department of Computer Systems
Engineering, University of Engineering
& Technology, Peshawar 25000,
Pakistan
yasirsaleem@uetpeshawar.edu.pk

Abid Iqbal

Department of Electrical Engineering
Jalozai Campus, University of
Engineering & Technology, Peshawar
25000, Pakistan
abid.iqbal@uetpeshawar.edu.pk

Abstract—Wheat is one of the major staple crops in Pakistan, playing a crucial role in ensuring food security and contributing to the country's economy. The productivity and quality of wheat crops, however, are vulnerable to several illnesses. The ability to diagnose these diseases quickly and accurately is crucial for taking the appropriate preventative actions, limiting losses, and maintaining food security. In this research paper, we build and test a wheat disease detection system adapted to the conditions in Pakistan. The suggested method uses machine learning based techniques along with image processing algorithms to automatically detect and categorize various wheat diseases based on their symptoms. High-resolution photos of healthy wheat plants and sick plants displaying different diseases were collected from different regions of Pakistan in order to construct an accurate and robust disease detection model. The dataset has been annotated by plant pathologists who provided true labels for use in evaluation and training. To achieve the best results in wheat disease diagnosis, many cutting-edge deep learning architectures were investigated and optimized. These included Convolutional Neural Networks (CNNs) and Transfer Learning models. Multiple models' effectiveness was evaluated using accuracy, precision, and recall, in a series of extensive trials.

Keywords: Wheat diseases, Convolutional Neural Networks (CNNs), Transfer Learning, Tensor Flow

I. INTRODUCTION

Wheat is a vital cereal crop in Pakistan, serving as a staple food for the nation's population and playing a significant role in its agricultural economy. However, the cultivation of wheat faces formidable challenges, with plant diseases being a primary concern. Various diseases, caused by fungal, bacterial, and viral pathogens, can severely affect wheat crops, leading to substantial yield losses and compromising food security in the country. Timely and accurate diagnosis of wheat diseases are crucial for implementing effective strategies for disease management [1]. Traditionally, disease identification has relied on manual visual inspection by experienced agronomists and plant pathologists. While effective, this process is time-consuming, subjective, and may lead to misdiagnosis due to the similarity of symptoms among different diseases.

In recent years, technological advancements in the fields of machine learning, computer vision and image processing have revolutionized the agricultural sector [2]. These

developments offer promising opportunities to automate disease detection and revolutionize the way we monitor and manage plant health. This paper aims to develop a wheat disease detection system tailored specifically for the conditions prevailing in Pakistan. By integrating cutting-edge machine learning algorithms and image processing techniques, the system will automatically identify and classify various wheat diseases based on visual symptoms exhibited by infected plants. To achieve this, a comprehensive dataset comprising high-resolution images of healthy and diseased wheat plants collected from diverse regions in Pakistan.

In many regions of the world, the absence of infrastructure makes it difficult to quickly identify wheat infections, which pose a danger to food security. To that end, we plan to develop a mobile application for diagnosing agricultural diseases. We propose an intelligent and effective application that leverages AI computer vision and machine learning algorithms to identify agricultural diseases. Our dataset is the Plant-Village Dataset (New), which is part of the CNN family. The latest iteration of the plant-village dataset includes 10,000 photos for training and 2,500 for validation. Separated set of 33 test photos used for evaluating the accuracy of the model.

II. LITERATURE REVIEW

Researchers have made great strides in the field of agriculture recently. The detection of plant diseases has been accomplished using a variety of methods. Many studies reveal that they employ various algorithms; among these are Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) [3]. Despite this, researchers are constantly on the lookout for new methods that are both more effective and more interesting to users.

Yusuke Kawasaki and Hiroyuki Uga in [7] analyzed methods for spotting plant diseases using photographs of their leaves. They discussed several methods for removing the afflicted portion of the plant. They also looked at certain feature extraction and clustering techniques for identifying plant diseases and distinguishing between healthy and contaminated leaves. To ensure a successful harvest, accurate recognition and categorization of plant infections using image processing is essential. Methods for removing the highlights of a contaminated leaf, characterizing plant diseases and several techniques for fragmenting the infected portion of the

plant. Disease in plants can be characterized with ease using ANN methods including self-sorting highlight maps, back spread calculations, support vector machines, and so on. Based on these methods, a picture-handling strategy can be used to accurately identify and rank various plant diseases [8]. Results were evaluated using a dataset of 87k RGB photographs of healthy and defected plant leaves divided into groups of 38, out of which 25 were chosen for experimental purposes. This makes use of AI and cameras to detect objects. The proposed approach has a 93% success rate in identifying 20 distinct plant diseases across 5 different species together with back propagation neural network (BPNN) and other digital image processing techniques.

In [5], Monica Jhuria, Ashwani Kumar, and others have discussed techniques for identifying plant illness in photographs of leaves. Two datasets are used in the implementation of support vector machine (SVM). In this case, training datasets are compared to their corresponding datasets stored photographs. After applying a filter, two photographs are compared to one another. After contrasting healthy and unhealthy regions, they arrived with a percentage fraction. An artificial neural network method has been applied for disease identification. And they have a wide variety of algorithms for doing disease detection. Some examples of algorithms used in the artificial neural network method are the back propagation, support vector machine, and major component analysis [4]. The proposed method has accuracy of 91%.

With the help of deep learning and image identification, Ahmed, A. A., and Reddy, G. H. in [9] examined the technological possibility of automating disease detection. Using a publicly available dataset with 54,306 images of healthy and diseased plant leaves, deep convolutional neural network has been trained to classify crop species based on their disease status into 38 categories, including 14 species of crops and 26 types of crop diseases. The model is accurate to within 1% on average. The average accuracy of random guessing on a dataset with 38 class labels is only 2.63%. Overall accuracy on the Plant Village dataset ranged from 85.53% (in case of AlexNet: Training From Scratch: Gray Scale:: 80 – 20) to 99.34% (in case of GoogLeNet :: Transfer Learning :: Color :: 80 - 20), demonstrating the promising potential of the deep neural network architecture.

III. METHODOLOGY

This section explains how to identify plant diseases using photographs of affected leaves. Extracting the image's attributes or valuable information from the image is the goal of image processing, a subfield of signal processing. It displays accurate results by evaluating multiple picture attributes to diagnose illnesses on plant leaves as shown in Figure 1.

Crop diseases pose a significant risk to food security, yet their prompt detection continues to be challenging in numerous regions globally due to inadequate infrastructure. The aim is to develop a mobile application for diagnosing diseases that can be easily accessed by farmers. The implementation through a mobile application aims to enhance access for the average farmer. Our solution includes identifying potential causes of diseases and offering corresponding treatments. Utilizing mobile phone

applications will assist farmers in improving their production levels and income.

Plant and crop diseases can be broken down into four categories: oomycetes, hyphomycetes, bacteria, and viruses. Visual examination of leaf color patterns and crown architecture remains the gold standard in conventional field scouting for crop diseases. Examining plant leaves for disease symptoms and diagnosing plant diseases based on experience require a significant amount of time, effort, and expertise when done using the naked eye. In addition, the wide range of plants means that illnesses can manifest themselves in a wide range of ways across various crops, adding a layer of complication to the process of categorizing plant diseases. Meanwhile, a lot of research has been done using machine learning to categorize plant diseases. First, the background is removed, or the infected part is segmented using preprocessing techniques; second, distinguishing features are extracted for analysis; and third, classification or clustering algorithms are used for feature classification [6].

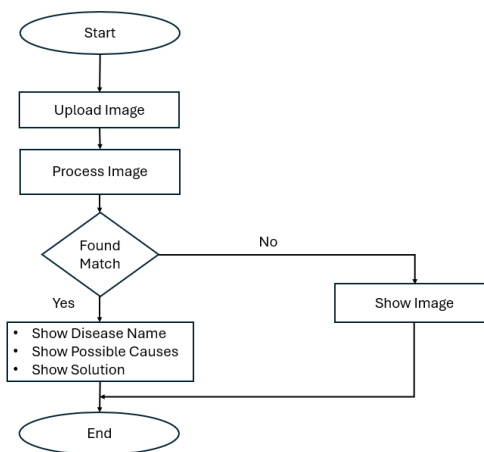


Figure 1. Flow chart of application

Most of the algorithms developed for previous machine learning techniques did not meet the needs of real-world applications, even though many novel algorithms have been established in this field. Agricultural sector that uses machine learning to improve crop yields are increasingly moving towards Deep Learning techniques, and in particular CNNs. Because of its versatility in detection and classification, such as weed detection, crop pests' categorization tasks, or identification of crop illnesses, deep learning approaches are increasingly used in agriculture production. One advantage of using a Deep Learning model is that it eliminates the need for a segmentation operation when extracting features from a task. The object's retrieved features are successfully mined from the raw data.

A. Convolutional Neural Network (CNN)

Major demands on CNN -based, when it comes to the classification of plant diseases, deep learning is unmatched in terms of both scale and variety of datasets [15]. The majority of leaf disease classification systems use CNN. Other types of DL networks, such as deconvolutional networks and fully convolutional networks (FCNs) are more commonly utilized for picture segmentation and medical diagnosis than for classifying diseases in plant leaves [16].

The image's local correlation is used by the convolutional layer in order to extract relevant features. The image's upper left corner is marked with a kernel. Multiplying the pixel values by their matching kernel values, summing the products, and finally adding the bias. The kernel is shifted by one pixel and the filtering process is repeated until the entire image has been processed. The pooling layer makes the model robust to translations, rotations, and scaling by randomly choosing features from the feature map of the higher layer. Maximum or average pooling is the most popular option. In maximum pooling, the input image is divided into numerous rectangular areas with the size of the filter determining which regions receive the maximum value. When regions are pooled together, the result is an average of all of them. In many implementations, convolutional layers follow a pooling layer and vice versa. For classification or detection tasks, the classifier integrates and converts multidimensional information into one-dimensional features at the fully connected layer, where each neuron is connected to the neuron above it [11].

- **VGG19**

VGG stands for Visual Geometry Group. The VGG network is specially crafted for tasks related to image classification. VGG19, a part of this network, is comprised of 19 layers, with 16 being convolutional layers and 3 fully connected layers. The convolutional layers are tasked with extracting features from input images, whereas the fully connected layers handle the classification of these features into various categories or classes.

Table 1: Dataset

Dataset	Images
Leaf rust	2000
Loose Smut	1800
Crown and Root Rot	1700
Healthy	2200

- **MobileNetV2**

MobileNetV2 is an attempt to design a convolutional neural network that can function well on mobile devices. It is predicated on a backwards residual structure, with the bottleneck levels connecting via residual nodes. Lightweight depth-wise convolutions are used in the intermediate expansion layer to filter features and introduce non-linearity. MobileNetV2's overall architecture consists of a 32-filter fully convolutional first layer, followed by 19-filter residual bottleneck layers [13].

B. Plant-Village Dataset

- The dataset as shown in Table 1 has been acquired from Hazara University Mansehra, KPK, Pakistan. It has 10k total images.
- It has 4 classes having diseased and healthy leaves.
- The dataset divided into 80/20 ratio (8000/2000) for training and validation respectively.

C. Data Augmentation

It's common knowledge that deep learning works best with a lot of information. Little data may not be sufficient for model training. For this purpose employing data augmentation to create new examples for the training phase. Common methods of data enhancement include geometric modifications including mirroring, cropping, rotation, and translation as shown in Figure 2.



Figure 2. Data Augmentation

D. Data Generation

In some cases, access to reliable information is limited. In this scenario, creating fake data for use in training detection model. The usage of synthetic data creation has grown in the field of machine learning because of its inexpensive cost. Generative Adversarial Networks (GANs) methods that can be used to generate fake data. To construct fictional instances from a dataset with the same properties as the original set, GAN uses a generative modeling technique called generative adversarial networks [11].

E. Model Selection

AlexNet, VGGNet, GoogLeNet, ResNet, MobileNet, and EfficientNet are only a few of the CNN-based classification models produced in DL-related research for use in classification tasks [14]. After placing first in the ImageNet competition in 2015, Microsoft Lab unveiled the ResNet network. Using shortcut connections and residual blocks, the network was able to address the problem of gradient reduction. In 2016, ResNet networks gained more attention in the field of Deep Learning.

In 2017, Google's engineering teams unveiled the MobileNet network for use in mobile gadgets and embedded software. In 2019, those same Google groups unveiled the EfficientNet network. The network used an easy compound coefficient to implement the strategy of scaling the depth, resolutions, or width. When it comes to plant and crop diseases, Deep networks are not required for classification but are the ideal solution because to their high performance. AlexNet and VGG16 are thought to be suitable for the actual accuracy performance needed in agricultural production, in addition to Deep networks [5]. This process of training and deployment can be divided into the following three steps. First step is data preprocessing and preparation. Second step is Model building, training and evaluation and final step is model inference and deployment as shown in Figure 3.

F. Preparing the Data and Preprocessing

Deep Learning models prioritize data preparation and preprocessing data. Strong accurate and trained input data bounds to precise results. Original datasets require training, validation and testing set processes, the general statistical percentages for such are 70:20:10, 80:10:10 and 60:20:20 [17]. In our model system, the model has been fed with public plant datasets by using Kaggle platform, as it contains 87k images.

A good DL model architecture is required prior to training. Better accuracy and faster classification can be achieved with a well-designed model. CNNs, RNNs, and GANs are the three most common forms of DL networks nowadays. When it comes to the task of detecting and classifying plant diseases, CNN is by far the most used feature extraction network.

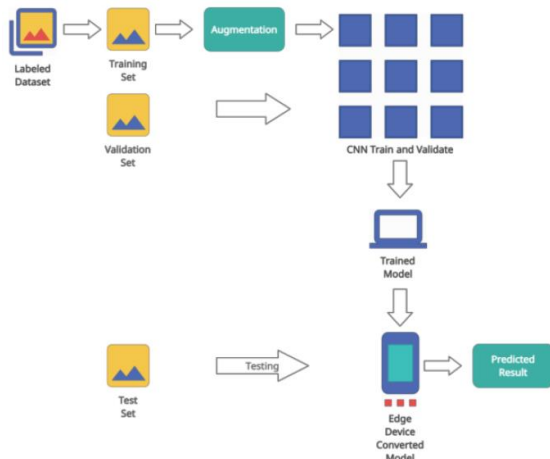


Figure 3. Steps in model training

Once the framework of the model has been built, hyper parameters can be adjusted for use in training and testing. The grid search technique can be used to iteratively explore several parameter configurations in pursuit of the optimal one. During neural network, training data are stored in the first layer, and back-propagation is used to adjust the weight of each neuron based on whether the output matches the label. This cycle is repeated until a new skill can be taught using the available data. The model's efficacy was measured with metrics like accuracy, precision, recall, and F1 score. These indexes can't be discussed in isolation; rather, they need to be introduced alongside the more general concept of a confusion matrix. In binary classification, the confusion matrix displays the expected yes/no answers [10].

G. Evaluation Measures

The classification models, involving both detection and classification of plant diseases, were implemented with deep learning. The statistical evaluation classified the samples of images into following statuses: True-Positive (TP), which determines the perfectly identified image samples being infected, False-Positive (FP), which determines the wrong classified image samples being infected, True-Negative (TN), which determines the correct classified image samples

being healthy and False-Positive (FP), which determines the wrong identified image samples being healthy.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

In plant disease evaluation, classification and accuracy are considered essential for the purpose. From equation 1 High value of accuracy and precision tends to be regarded better for the performance. When the value of F1 is less, the trained model tends to perform much better. The capability of the trained model is applied to new data when the training and evaluation processes are finished.

H. Transfer learning

Transfer learning comes in the classification of the machine learning technique domain. The certain technique adopts the learning capabilities from the recent tasks to the proceeding tasks. With new databases, few layers of pre-trained networks of the model are retrained by reducing the need for masses of datasets, inclining the model towards better performance. Research by Mukti et al reports the utilization of transfer learning model by using ResNet50, by implementing the approach of recognizing plant diseases, as it gives satisfactory results. The report contains a dataset of 87,867 image samples where 80% of the dataset is used for training set and the remaining 20% is used for validating set process. The report concludes with an accuracy of 99.80% from the model implemented practically.

I. Tensor Flow

TensorFlow is an open-source machine-learning framework used by researchers and developers. It offers a rich ecosystem of tools and resources for creating and deploying ML apps. Users can choose the level of abstraction they need, with the Keras API simplifying model building. Eager execution allows for fast iteration and debugging. The Distribution Strategy API enables distributed training without modifying model design, making it suitable for large ML jobs. TensorFlow also supports creating complex models efficiently with features like the Keras Functional API. Supplementary libraries like TensorFlow Probability and BERT can be used alongside TensorFlow for various tasks.

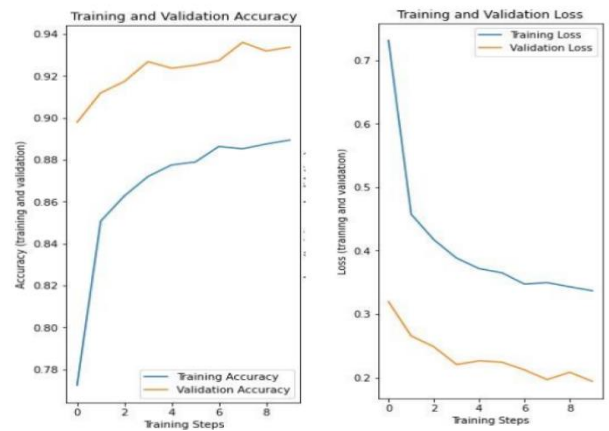


Figure 4. Model training and validation.

IV. RESULTS

The proposed system results are gathered from trained model before its deployment and after its deployment in mobile based application. We used a technique called cross-validation, in which we divided their data into a training set and a validation set. The model is "trained" using the training set, while its performance is "validated" using the validation set. The figure 4 depict loss and accuracy during training and validation, respectively.

The validation accuracy is higher than the training accuracy. On the training dataset, the model's loss will nearly always be smaller than on the validation dataset. Thus, a discrepancy between the train and validation loss learning curves is to be anticipated. The void between these two ideals is known as the "generalization gap." A validation loss that is smaller than the training loss may also be used to detect it. It suggests the validation dataset may be more predictable by the model than the training dataset.

Accuracy is used as a measure of style in the context of typography. Accuracy refers to the proportion of correctly predicted events that our version anticipated. The formal definition of precision is as follows: the proportion of correct predictions to total forecasts is the standard by which accuracy is measured. We see convergence in our model. We obtained a validation accuracy of 93%+ in just 10 epochs as shown in Figure 4.

```
Epoch 1/50
56/56 [=====] - 76s 1s/step - loss: 1.9670 - accuracy: 0.5963 - val_loss: 0.9167 - val_accuracy: 0.7188
Epoch 2/50
56/56 [=====] - 47s 842ms/step - loss: 0.9288 - accuracy: 0.6926 - val_loss: 0.6480 - val_accuracy: 0.7587
Epoch 3/50
56/56 [=====] - 50s 895ms/step - loss: 0.7430 - accuracy: 0.7321 - val_loss: 0.6070 - val_accuracy: 0.7743
Epoch 4/50
56/56 [=====] - 49s 876ms/step - loss: 0.6785 - accuracy: 0.7539 - val_loss: 0.5885 - val_accuracy: 0.7951
Epoch 5/50
56/56 [=====] - 47s 845ms/step - loss: 0.6091 - accuracy: 0.7811 - val_loss: 0.5149 - val_accuracy: 0.8090
Epoch 6/50
56/56 [=====] - 50s 899ms/step - loss: 0.5869 - accuracy: 0.7794 - val_loss: 0.4834 - val_accuracy: 0.8316
Epoch 7/50
```

Figure 4. Accuracy values

To make our model communicate with App we must convert into the TensorFlow lite version, tflite is made for mobile Versions [12]. So, you can build or create a mobile app and make the app communicate with the model. The following steps are being done:

- Saved model converted into TFLite.

```
model.save('mobilenet.h5')
```

- Model converted to TFLite which is then used to develop a mobile application.

```
# convert the model to TFLite
import tensorflow as tf
from tensorflow import lite
from tensorflow.keras.models import load_model
converter = lite.TFLiteConverter.from_keras_model(model)
tfmodel = converter.convert()
open("model.tflite", "wb").write(tfmodel)
```

The prototype application uses camera or device media to get an image of the crop as shown in Figure 5. Preview the images and send it to API, for disease detection are shown in figure 6 and figure 7. Results page showing detected disease as given in Figure 7 and 8 for diseases such as Root Rot and Leaf Rust. The healthy wheat plant results are given in Figure 9.

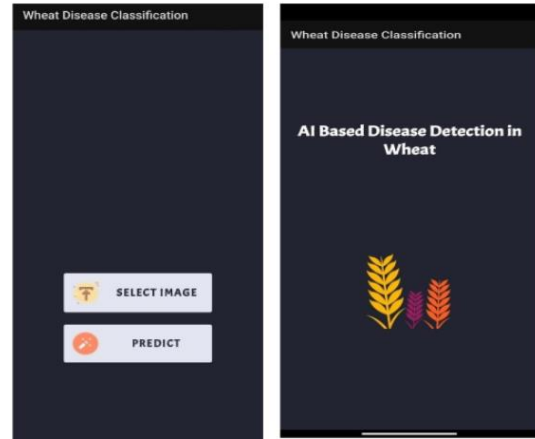


Figure 5. Prototype of mobile application

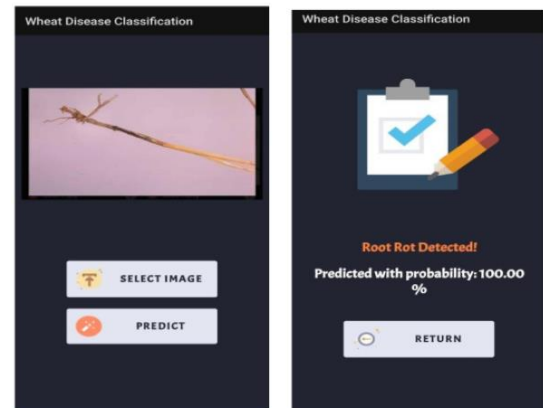


Figure 6. Root Rot disease

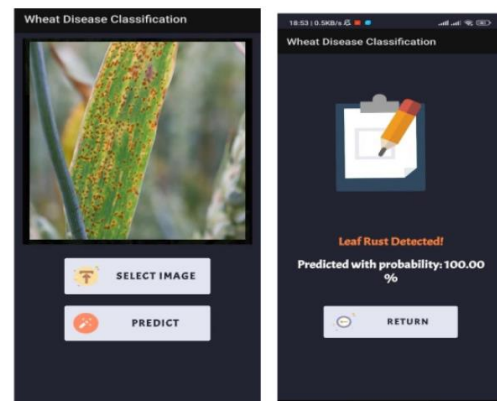


Figure 7. Leaf Rust disease

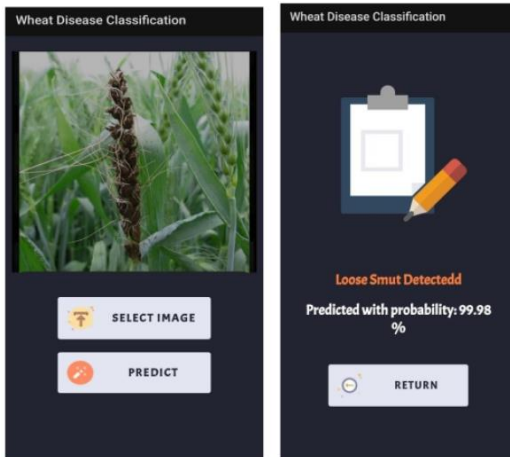


Figure 8. Loose smut disease

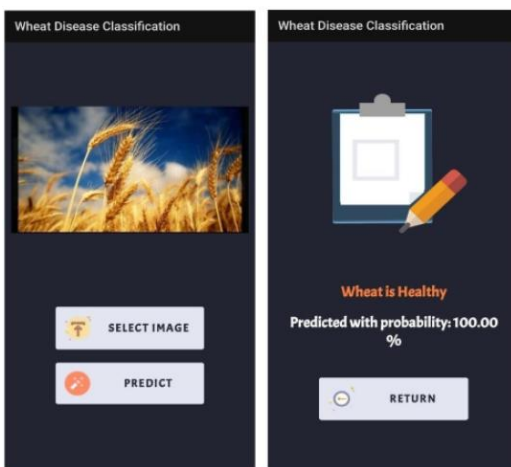


Figure 9: Healthy leaves

V. CONCLUSION

Wheat diseases are serious hazards to food security and must be addressed immediately to prevent total crop failure. Many times, however, farmers are unable to tell the difference between diseases that present with identical symptoms. Because of this, fertilizer applications may be under- or over-applied. If diseases are misdiagnosed. Professionals in the field are required to identify wheat crop illnesses. Who can prevent the loss of the entire harvest? To mitigate this loss and better instruct farmers via video, in proposed solution we apply Convolutional Neural Network (CNN) multi-layer ANN algorithms hereinafter referred to as Deep Learning Algorithms. Overarching goal was to enhance the efficacy of agricultural methods. Currently, machine vision-based plant disease and pest detection equipment is being widely used in agriculture, replacing the more laborious and time-consuming practice of identifying these problems by eye. We trained and validated our strategy using a transfer learning method. We also conducted an evaluation of the design using a dataset of 14 plant types and 38 class labels. We propose an intelligent and effective application software that leverages AI computer vision and machine learning algorithms to identify agricultural diseases. We have faith that this study's findings will be seen as supplementary to those already published,

opening the door to important studies of transfer learning methodologies for plant and disease identification.

REFERENCES

- [1] S. S. Hari, M. Sivakumar, P. Renuga, S. Karthikeyan and S. Suriya, "Detection of Plant Disease by Leaf Image Using Convolutional Neural Network," 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019, pp. 1-5, doi: 10.1109/ViTECoN.2019.8899748.
- [2] L. Ale, A. Sheta, L. Li, Y. Wang, and N. Zhang, "Deep Learning Based Plant Disease Detection for Smart Agriculture," 2019 IBEE Globecom Workshops (GC Wkshps), 2019, pp. 1-6, doi: 10.1109/GCWkshps45667.2019.9024439.
- [3] Boulent J, Foucher S, Théau J, St-Charles PL. Convolutional neural networks for the automatic identification of plant diseases. *Front Plant Sci.* 2019;10:941.
- [4] Yusuke Kawasaki, Hiroyuki Uga, Satoshi Kagiwada, and Hitoshi Iyatomi. Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In *International Symposium on Visual Computing*, pages 638-645. Springer, 2015.
- [5] Agarwal, M.; Gupta, S.; Biswas, K.K. A new conv2d model with modified relu activation function for identification of disease type and severity in cucumber plant. *Sustain. Comput. Inform. Syst.* 2021, 30, 100473.
- [6] Sankaran, S.; Mishra, A.; Ehsani, R.; Davis, C. A review of advanced techniques for detecting plant diseases. *Comput. Electron. Agric.* 2010, 72, 1-13.
- [7] Türko glu, M.; Hanbay, D. Plant disease and pest detection using deep learning-based features. *Turk. J. Electr. Eng. Comput. Sci.* 2019, 27, 1636-1651.
- [8] Lin, K., Gong, L., Huang, Y., Liu, C. and Pan, J., 2019. Deep learning-based segmentation and quantification of cucumber powdery mildew using convolutional neural network. *Frontiers in plant science*, 10, p.155.
- [9] Ahmed, A. A., & Reddy, G. H. (2021). A mobile-based system for detecting plant leaf diseases using deep learning. *AgriEngineering*, 3(3), 478-493.
- [10] Malathi, V., & Gopinath, M. P. (2021). Classification of pest detection in paddy crop based on transfer learning approach. *Acta Agriculturae Scandinavica, Section B-Soil & Plant Science*, 71(7), 552-559.
- [11] S. P. Mohanty, D. P. Hughes and M. Salathé, "Using deep learning for image-based plant disease detection", *Frontiers Plant Sci.*, vol. 7, pp. 1419, Sep. 2016
- [12] R. Gour, *The essential guide to Learn Tensorflow mobile and TENSORFLOW LITE*, April 2019.
- [13] Malathi, V., & Gopinath, M. P. (2021). Classification of pest detection in paddy crop based on transfer learning approach. *Acta Agriculturae Scandinavica, Section B-Soil & Plant Science*, 71(7), 552-559.
- [14] J. F. Yeh, K. M. Lin, C. Y. Lin and J.C. Kang, "Intelligent Mango Fruit Grade Classification Using AlexNet-SPP with Mask R-CNN based Segmentation Algorithm", *IEEE International Conference on Consumer Electronics-Taiwan*, 2021.
- [15] G. Fenu and F. M. Mallocci, "An application of machine learning technique in forecasting 'crop disease'", *Proceedings of the 2019 3rd International Conference on Big Data Research*, pp. 76-82, 2019, November.
- [16] F. Yi and R. Ramasamy, *Current and prospective methods for plant disease detection*, April 2021
- [17] Vishal Mani Tiwari and Tarun Gupta, "Plant Leaf Disease Analysis using Image Processing Technique with Modified SVM-CS Classifier", *ResearchGate* 2017.
- [18] Pranjali B. Padol and Anjali A. Yadav, "SVM classifier based grapeleaf disease detection", *Conference on advances insignal processing*, 2016.

Home Automation Using Internet of Things and Machine Learning

Laila Amin

dept. of Computer Systems Engineering
University of Engineering and Technology Peshawar
Peshawar, Pakistan
lailaamin.cse@uetpeshawar.edu.pk

Nasru Minallah

dept. of Computer Systems Engineering
University of Engineering and Technology Peshawar
Peshawar, Pakistan
n.minallah@uetpeshawar.edu.pk

Abstract— This paper proposes an energy-efficient home automation system leveraging the Internet of Things (IoT) and machine learning. The system, implemented in Python on a Raspberry Pi, enables remote control of appliances (lights, televisions, air conditioners) via a web interface accessible from any local network device. Machine learning is introduced in the second phase, utilizing linear regression to automate appliance management based on historical data stored in a database. This work demonstrates the feasibility of IoT and machine learning for cost-effective and efficient home automation, laying the groundwork for future development of database-driven smart homes with advanced machine learning algorithms.

Keywords: Home Automation, Internet of Things, Machine Learning.

I. INTRODUCTION

Home automation systems have gained significant traction in recent years, offering various functionalities and specifications [1-8]. Existing literature explores a range of approaches, including ZigBee-Arduino control for small appliances [1], low-cost Arduino UNO-based systems with web server functionality and environmental sensors [2], X10 controller and IEEE 1394 AV framework integration for remote access [3], Java-based structures for PC server control [4], PIC microcontroller-driven systems with web interfaces for slave control points [5], Bluetooth-over-internet appliance control [6], and RF module-Arduino UNO-WiFi-cloud-mobile application systems with environmental sensing for fan and light control [7]. Additionally, research has been conducted on minimizing energy usage through multiband antennas, energy-efficient sensors, and thermal management [8]. However, limitations exist in many proposed systems. Some prioritize simplicity, focusing on basic controls for fans and lights, while others target complex devices like intelligent doors. Issues arise with user-friendliness, basic functionality, cost, and energy consumption, leading to a gap between user requirements and system capabilities.

This paper addresses these limitations by proposing a cost-effective, user-friendly Internet of Things (IoT) home automation system. The system offers internet, PC, and mobile application control for home appliances, promoting energy efficiency. We leverage a Raspberry Pi as a bridge device, translating user interaction through a web page into signals for end-device control. The system utilizes Wi-Fi technology for local network communication and, with an internet-connected server, allows remote access via a web browser. This paper details the software and hardware implementation, focusing on the initial control of three key



Figure 1: Possibilities with Smart Security [9]

appliances: fan, air conditioner, and light. The proposed system promises to be cost-effective, reliable, and easy to implement, aiming to bridge the gap between user needs and existing home automation solutions..

II. ELECTRONICS AND THEIR APPLICATION IN AUTOMATION

1) *Raspberry Pi*: This work utilizes a Raspberry Pi 3B+ as the central hub for communication and processing. The Raspberry Pi offers connectivity through one RJ45 connector, four USB ports, and an integrated Wi-Fi adapter. This allows for interaction with various communicating terminals within the smart home environment.. The Raspberry Pi serves two primary functions, It facilitates communication between all connected appliances and sensors. And it stores historical data in its memory for machine learning purposes. A Flask server running on the Raspberry Pi manages the database and facilitates model training.

2) *Passive Infrared sensor*: A Passive Infrared (PIR) sensor is employed to detect infrared radiation, enabling the system to turn appliances on or off based on user presence. Ideally, upon entering a room, lights should illuminate based on a pre-set schedule retrieved from the database. Conversely, if a user leaves the room unexpectedly, the PIR sensor detects the absence of human movement and triggers the Raspberry Pi to turn off appliances, promoting energy efficiency. The sensor transmits a binary value ("1" for presence, "0" for absence) to a designated pin on the Raspberry Pi, which is also displayed on the web interface for user convenience. This eliminates the need for intrusive camera installations for presence detection.

3) *Relays*: Relays are crucial for switching connected devices on and off. The control signal received by the relay determines whether a specific pin on the IoT device is

activated or deactivated. This enables remote control of appliances within the smart home system. This project utilizes 5-volt, 10-ampere relay modules for the aforementioned functionality.

4) *Flask Server*: Flask, a web framework, provides the necessary libraries for building web applications. In this project, a Flask server is implemented to host all connected devices. It facilitates sending "ON" and "OFF" signals to these devices. The server is hosted on the Raspberry Pi with a local IP address, restricting access to devices within the same local subnet for security purposes. This necessitates the smartphone controlling the system to be connected to the same subnet. Alternatively, a VPN connection could be established for remote control from outside the local network

III. IMPLEMENTATION OF HOME AUTOMATION SYSTEM

A. Prototype Development

The prototype for the machine learning-based home automation system using a Raspberry Pi is connected to three electrical devices (fan, television, and light source) the connections to the Raspberry Pi are through a breadboard and jumper cables. A Flask server running on the Pi provides a web interface for remote control (on/off) of these devices. In parallel, a Python script maintains a database logging device activation times and associated GPIO pins. This data serves as the foundation for a routine learning algorithm, enabling the system to predict and automate device behavior after a seven-day training period. The web interface remains accessible from laptops or smartphones within the same local network for manual control or system monitoring. The Implementation is completed in two phases

1) *Phase 1*: controlling all the appliances in a smart home environment through any internet enabled device like a Smartphone or Laptop etc. The connectivity is done by having a raspberry pi as a bridge device to translate signals generated by human interaction through web page to the end terminal devices.

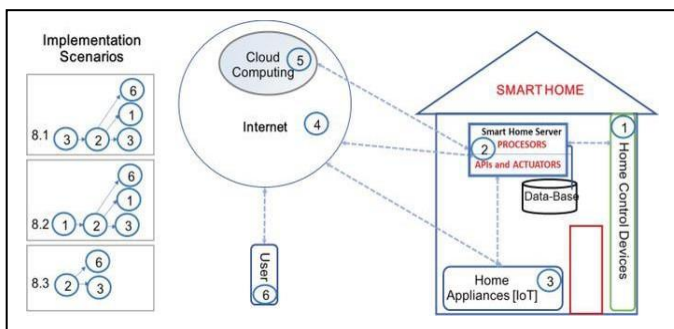


Figure 2: Smart home implementation chart

2) *Phase 2*: Phase 1 is enhanced in phase 2 by automating human interaction by in the introduction of databases that are maintained through machine learning algorithm

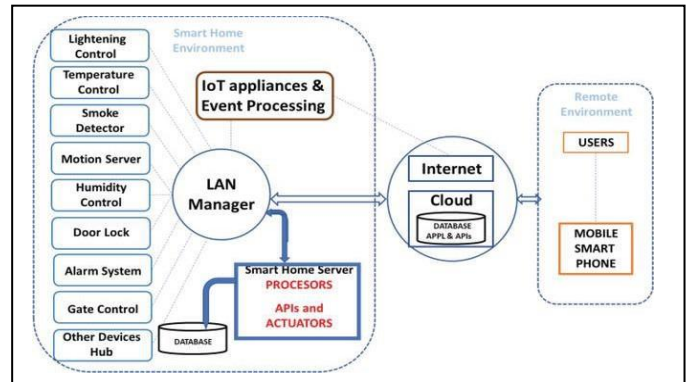


Figure 3: Detailed implementation with introduction of machine learning

B. Setup for Smart Home Project Implementation

When the device (Raspberry PI) is switched on. The most important element is, its connection to the Local Area Network. In this way it can communicate with the existing communicating devices. After the device is allocated an IP either through DHCP or assigning a static IP. The web GUI of the project can be accessed. This GUI enables the user to control the connected appliances. In the web GUI the user can select the mode of operation to be Manual or Automatic. The first mode enables the project to use data from the databases for the corresponding behavior of the project while the second mode needs the interaction from user to turn on /off the connected devices. Once the interaction is complete the existing database is updated and the flow ends.

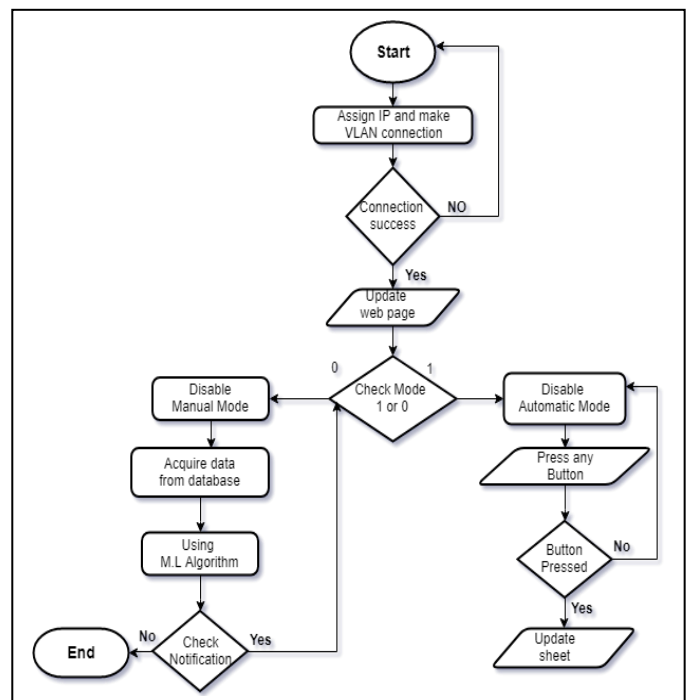


Figure 4: Flow Chart for Implementation of the Smart Home

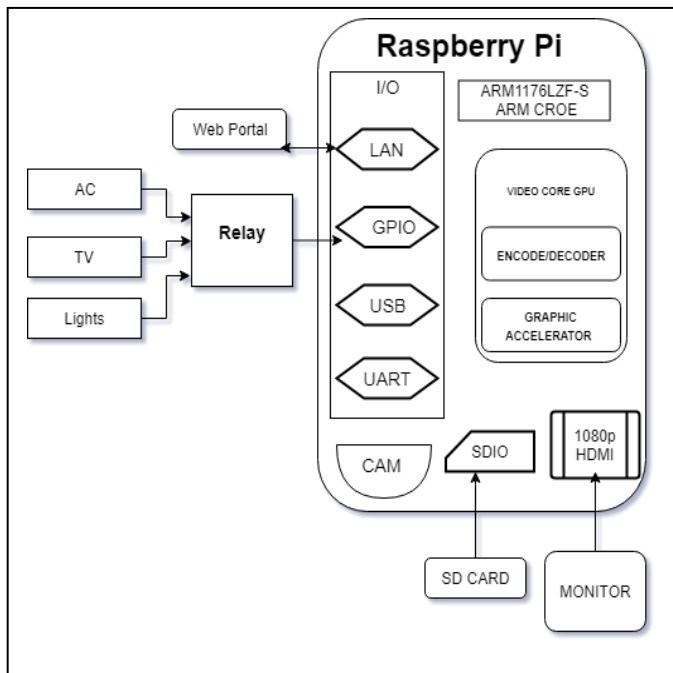


Figure 5: Procedure of Setup

The development of a machine learning-based home automation system using a Raspberry Pi 3B+ leverages Ubuntu 16.04 for visualization and Python 3.5 for core functionality. Code development is facilitated by Spyder 3.3.0. A user-friendly web interface is implemented using HTML to enable remote control and facilitate future application integration. Python scripts interact with the Raspberry Pi's GPIO pins to control connected appliances (fan, television, light source) via IoT protocols. A basic data logging system is established, utilizing a Python script to record device activation times and associated timestamps into a text file (.txt). This data serves as the foundation for a future machine learning algorithm, enabling the system to learn user behavior and automate device operation.

IV. RESULTS

The data of sensor are sent to the web browser for monitoring of the system after the successful connection to the server. The webpage will appear when we enter the IP address in the web browser. The web server provides important information about the status of the home appliances that are connected to the internet remotely.

Home automation makes it possible to automate tasks related to security, well-being and comfort through a smart system installed in a home or building. In other words, it integrates technology into the design of a space. One of the main advantages of home automation systems is energy efficiency.

V. CONCLUSION

This paper contributes to the growing field of smart home automation by leveraging the Internet of Things (IoT) and machine learning. We present a Raspberry Pi-based system that interfaces with various electrical appliances, enabling their remote control and automated operation. In the initial phase, the system provides a user-friendly interface for manual control. Subsequently, a machine learning

component is introduced to analyze historical usage patterns and automate appliance switching based on learned schedules. This approach not only promotes energy efficiency but also enhances user convenience by reducing manual intervention and offering increased flexibility in managing time and resources.

VI. FUTURE WORK

This work's inherent scalability allows for the integration of "smarter" appliances, controllable either through smartphones or autonomously based on historical data. This opens doors to numerous domestic applications, including smart cameras, self-activating cleaning robots, and intelligent washing machines. By leveraging machine learning algorithms, household chores can be automated through these appliances, enhancing user convenience and promoting significant cost savings via efficient resource management.

ACKNOWLEDGMENTS

Special thanks to Dr. Nasru Minuallah, who acted as my project supervisor and examiner respectively, for his ideas, patience, supports and guidelines to come up with this project. I would also like to thank my parents for their support.

REFERENCES

- [1] J. Holler, J. Tsiatsis, V., Mulligan, C., Karnouskos, S., Avesand, S. and Boyle, D., 2014. Internet of Things. Academic Press.
- [2] Kopetz, H., 2011. Internet of things. In Real-time systems (pp. 307-323). Springer, Boston, MA.
- [3] Image from URL: https://www.google.com/search?q=smart+security+iot&client=firefox-b-d&source=lnms&tbm=isch&sa=X&ved=0ahUKEwixKCK2o_kAhUFaFAKHTt6Dj8Q_AUIESgB&biw=1920&bih=944#imgrc=w hZs6WmUrfZrnMI.
- [4] Upton, E. and Halfacree, G., 2014. Raspberry Pi user guide. John Wiley & Sons.
- [5] Jain, S., Vaibhav, A. and Goyal, L., 2014, February. Raspberry Pi based interactive home automation system through E-mail. In 2014 International Conference on Reliability Optimization and Information Technology (ICROIT) (pp. 277-280). IEEE.
- [6] Guinard, D. and Trifa, V., 2016. Building the web of things: with examples in node.js and raspberry pi. Manning Publications Co..
- [7] Maksimović, M., Vujović, V., Davidović, N., Milošević, V. and Perišić, B., 2014. Raspberry Pi as Internet of things hardware: performances and constraints. design issues, 3(8).
- [8] Montgomery, D.C., Peck, E.A. and Vining, G.G., 2012. Introduction to linear regression analysis (Vol. 821). John Wiley & Sons.
- [9] Müller, A.C. and Guido, S., 2016. Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".

DEEP LEARNING-BASED SKIN LESION SEGMENTATION AND CLASSIFICATION

Ayesha Rani

Computer System Engineering
University of Engineering and
Technology
Peshawar, Pakistan
ayesharani6788@gmail.com

Haseeb Ullah

Computer System Engineering
University of Engineering and
Technology
Peshawar, Pakistan
haseebullahbj@gmail.com

Hafiza Atika Shahab

Computer System Engineering
University of Engineering and
Technology
Peshawar, Pakistan
atikashahab4455@gmail.com

Ammad Khalil

Computer System Engineering
University of Engineering and
Technology
Peshawar, Pakistan
amaadkhalil@uetpeshawar.edu.pk

M. Abeer Irfan

Computer System Engineering
University of Engineering and Technology
Peshawar, Pakistan
abeer.irfan@uetpeshawar.edu.pk

Yaser Ali Shah

Department of Computer Science,
COMSATS University Islamabad, Attock
Campus, Attock 43600, Pakistan
Yaser@cuiatk.edu.pk

Abstract-- By using deep learning to automate skin lesion segmentation, this work aims to improve the classification of melanoma. By properly segmenting lesions and utilizing the U-Net algorithm's preprocessing capabilities, our research aims to improve the accuracy of skin cancer diagnosis. During preprocessing, raw dermoscopic pictures from the HAM10000 dataset are enhanced and normalized early. Next, the U-Net model is used to accurately segment lesions. Advanced deep learning approaches are applied after segmentation segmented images are subjected to classification, such as Convolutional Neural Networks (CNN) and Vision Transformer (ViT) models. This all-encompassing method not only improves dermatological image analysis's dependability and effectiveness, but it also shows promise for enhancing clinical outcomes in the diagnosis and management of different forms of skin cancer. Our work is a significant step toward the creation of more reliable techniques in this important area, opening the door for improvements in patient care and healthcare diagnostics.

Keywords:

Melanoma, Skin cancer, lesion segmentation, deep learning, lesion classification, image processing, vision transformer.

I. INTRODUCTION

Skin cancer originates in skin cells due to the uncontrollable growth of skin cells. Skin cancer is primarily caused by UV radiation damage to the cell's DNA, which is passed on by exposure to sunlight or artificial UV radiation sources. Melanoma, squamous cell carcinoma (SCC), and basal cell carcinoma (BCC) are the most common forms of skin cancer.

According to [1], each year the number of cases of melanoma identified, increased by 53% between 2008 and 2018. If the skin cancer is detected later, the survival rate is less than 14%. Over the next decade, there will likely be an increase in the death rate. On the other hand, if skin cancer is diagnosed in the early stages, the survival rate will increase to 97%.

The increasing global prevalence of skin cancer highlights the urgent need for efficient techniques for both detection and categorization. With the advancement of

technology, computer-aided diagnostic systems started to be used in the identification of skin cancer. Inspired by the critical need for detecting skin cancer in an early stage, scientists began to employ machine learning methods in unconventional ways [2]. Deep learning, a subset of AI (artificial intelligence), has been integrated into skin cancer detection in recent years, raising the bar for accuracy and efficiency in skin cancer diagnosis through automated, data-driven insights [3].

For the diagnosis of skin cancer segmentation is the most critical step for extracting features and isolating the skin lesion. The conventional method for classifying skin lesion images involves segmenting and pre-processing the image. After extracting characteristics from the region of interest, the lesion is classified using different classifiers [4]. The segmentation approach reduces the image processing steps. Seeja et al. [5] suggests that deep learning can also improve diagnostic efficiency by simplifying the process of interpreting images by using a model to extract representative features from lesions by combining the input image with a segmentation mask.

In this paper, we present a novel approach for the segmentation of skin cancer utilizing the deep convolutional neural network based on the U-Net algorithm in preprocessing. This work focuses on the segmentation and classification of different seven classes of skin cancer by employing the HAM10000 Dataset for training and evaluation. The experimental results showcase the effectiveness of our proposed method achieving the notable segmentation accuracy of seven classes of skin cancer. In the pre-processing step, we utilized the U-Net architecture for segmentation. For classification, we employed ViT and CNN architectures.

II. RELATED WORK

Rafael Luz Ara'ujo et.al [6] proposed a segmentation method for melanoma skin lesions using modified U-net along with post-processing techniques. The research was conducted on two datasets, PH2 and DermIS involving acquisition and

segmentation with a highly effective U-net network, and for further improvement, they performed post-processing techniques that disconnected extra regions, filled holes, and removed loose regions. In the PH2 dataset, they were able to acquire a dice coefficient of 0.933, and in the DermIS dataset, 0.872.

For the fast and accurate detection and segmentation of melanoma skin lesions, M. Taghizadeh et al. [7] proposed a method using SegNet and Yolov3 Based on Deep Transfer Learning. He suggests a two-phase procedure for melanoma detection. F-YoloV3 is applied for melanoma localization and F-SegNet for segmentation.

Bisla et al. [8] suggested a deep learning model for both classification and segmentation for the diagnosis of skin lesions. The segmentation technique was to mask away portions of the image that weren't needed. He utilized the U-net architecture for segmentation. To improve the accuracy of the automated diagnosis of melanoma through deep learning image segmentation technique, Aleksandra Dzieniszeewska et al [4] combined the segmentation mask and skin lesion images using Gaussian blur. They employed the deeplabV3 and U-net segmentation networks for the segmentation process. On the combined ISC dataset, they obtained an accuracy of 84.85%. The classification performance of melanoma was enhanced by Seeja R D, and Suresh A [5] using deep learning-based automatic skin lesion segmentation. The process unfolds in three stages: Segmentation using U-Net, Feature Extraction (color, texture, shape) using HOG, LBP, Edge Histogram, and Gabor methods. , and Classification using SVM, Random Forest, K-NN and naïve Bayes. The SVM classifier yields the best result on the ISBI 2016 dataset based on F1-score and accuracy. For image segmentation, they achieved a Dice co-efficiency value of 77.59% and the SVM classifier produced 85.19 % accuracy. It is observed that classification with segmentation achieves much better accuracy, sensitivity, and specificity of the model compared to classification unsegmented images.

Mehwish Zafar et al. [9] proposed a segmentation model in which features are extracted through a pre-trained MobilebetV2 model. This model acts as a base of Deeplabv3+ for boundary extraction. Using the ISIC 2016, 2017, 2018, and PH2 datasets, the suggested segmentation strategy is assessed based on Mean Accuracy, Global Accuracy, BF Score, Weighted IoU, and Mean IoU. These metrics yield global accuracy values of 0.97481, 0.97297, 0.98642, and 0.95914, respectively.

Nojus Dimša et al [10] suggested automatic segmentation of skin lesion using deep learning, which explores the crucial field of melanoma diagnosis and highlights the important significance of early detection.

MultiResUNet outperforms U-Net++ by a tiny amount (0.86%), all things considered, these U-Net variations show promise for improving the traditional U-Net model in skin lesion segmentation. Still, multi-class segmentation in skin lesions is a difficult field of study due to its complexity.

Researchers proposed a deep learning-based approach, comprising fuzzy k-means clustering (FKM) and region based convolutional neural network (RCNN), to classify skin melanoma at an early phase [11]. The PH2, ISBI-2016, and ISIC-2017 datasets were utilized to evaluate the efficacy of the methodology that was offered. It beat existing state-of-the-art approaches, according to the data, with an average accuracy of 95.40%, 93.1%, and 95.6%.

Mohammad Ali Kadampur et al [1] proposed a skin cancer detection method using Deep Learning Studio (DLS). Their approach involved data preparation, model construction, tuning, and deployment as a REST API. The DLS model achieved an exceptional AUC of 99.7% in skin cancer detection, demonstrating its effectiveness and ease of use.

Ameri A [12] presented a robust deep-convolutional neural network utilizing AlexNet to classify skin lesions as benign or malignant .This model shows significant potential for assisting dermatologists in skin cancer detection with a classification accuracy of 84%, sensitivity of 81%, and specificity of 88%.

Using the Vision Transformer architecture, [13] research presents a unique method for classifying skin cancer that achieves an impressive 96.15% accuracy on the HAM10000 dataset. The work outperforms conventional deep learning techniques by utilizing pre-trained models like ViT patch-32, which has promising potential for dermatological diagnostics. The model's performance is further improved by using Segment Anything Model for lesion segmentation, proving its usefulness in computer-aided skin cancer diagnosis. This study demonstrates the noteworthy advancements in deep learning applications for the interpretation of medical images, especially in the field of dermatology, which enhance patient outcomes by enabling prompt and precise diagnosis.

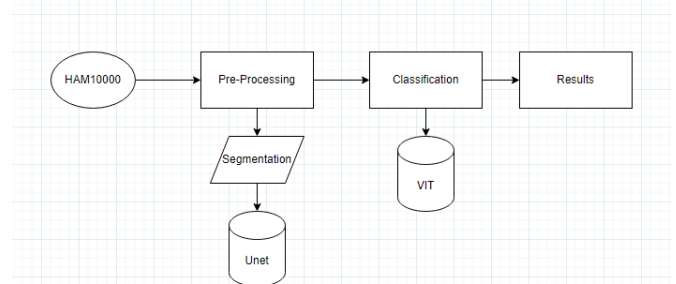


Figure. 1. Flow chart of proposed method.

III. MATERIALS AND METHODS

In our approach, we utilize deep learning techniques for automatic skin lesion segmentation using U-Net algorithm as a preprocessing step to improve the classification of melanoma. Convolutional neural network architecture U-Net performs exceptionally well in biomedical image segmentation applications [4] [5] [14]. We employ the HAM10000 Skin Cancer dataset, which consists of various classifications of skin lesions, including melanocytic nevi, melanoma, basal cell carcinoma, actinic keratoses, vascular lesions, and dermatofibroma. The methodology involves the following steps:

DATASET

We use the HAM10000 dataset, which consists of 10,015 dermoscopic pictures with extensive clinical metadata, in our study. This collection includes a variety of skin lesion representations, including benign nevi, malignant melanoma, and other dermatological diseases [14]. The HAM10000 dataset makes it easier to train and assess machine learning models for automated skin lesion recognition and segmentation because of its diverse population sources and clinical contexts. *A.*

A. Process:

Some separate subtasks that are suited for various input skin image types comprise the classification process:

- **Unaltered Lesion Classification:** In this subtask, skin lesions are categorized without any segmentation or preprocessing. It acts as a reference point for comparing segmented lesion classifications.
- **U-Net Segmented Lesion Classification:** Lesions are automatically separated from input images by utilizing the U-Net segmentation model. The U-Net architecture is trained to precisely identify skin lesions and is well-known for its efficacy in biomedical image segmentation.

B. Pre-Processing

Our preprocessing stage is used for segmentation tasks by preparing the raw dermoscopic pictures. By means of the given service, we analyze the data to remove biases, the adjustment of the intensity levels and also improve the image quality. The delineation procedure is performed in a next step to draw a boarder around a skin lesion. We will before segregation and classification to do certain preprocessing tasks so as to make accurate and consistent process.

1) Data Cleaning

To ensure data integrity and dependability for our research, we used data cleaning techniques during the preprocessing step to correct mistakes, inconsistencies, and missing values in the dataset.

2) Distribution of 7 Different Classes:

Subsequently, we Analyze the dataset's distribution of the seven distinct groups of skin lesions to identify any class imbalances.

3) Addressing Class Imbalance:

To balance the distribution we decide to employ both upsampling and down-sampling methods. A more equal distribution of the classes resulted in down-sampling, in which randomly selected samples of samples from the majority class were used to trim the size of the minority class. Moreover, we ensured that each class of the minority samples was replicated using such methods as up-sampling techniques to the level of attaining the balance of the dataset and a proper representation of each class.

4) Analysis of Spatial and Demographic Factors:

We performed an analysis to understand the distribution of skin lesions in several localized fields (e.g., arms, legs, torso) as well as the demographic features (gender, age) of skin lesioned individuals. The goal of the comprehensive study was to find geographic patterns or variances in the distribution of lesions and demographic trends or correlations, offering significant data for the classification of skin lesions.

C. Segmentation

A crucial preprocessing step in our work is segmentation, which is used to precisely identify skin lesions from dermoscopic pictures. Each of the numerous subtasks that make up the segmentation approach contributes to the process's improvement and optimization.

1) Model Definition and Training

Our approach relies heavily on the U-Net segmentation model to precisely identify skin lesions from dermoscopic images. The U-Net architecture, which is well-known for its effectiveness in biomedical image segmentation, is described inside a function that enables the customization of parameters such as number of epochs. By iteratively modifying its parameters throughout the training phase and utilizing the rich contextual data that its architecture captures, the U-Net model gains the ability to accurately recognize skin lesions.

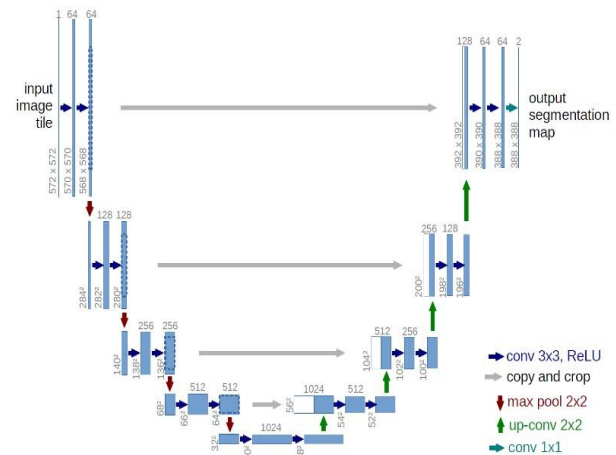


Figure.2. Illustration of U-Net Architecture [4]

2) Data Loading and Preparation

To provide consistency and structure in data handling, a function is defined to load the dataset in a sorted order. The dataset is divided into training and test sets to evaluate model performance impartially and prevent biases.

3) Evaluation Metrics:

As evaluation metrics, intersection over union (IoU), Jaccard Index and dice coefficient are used to measure how accurate and similar segmented lesions are to ground truth annotations. Jaccard Index, also known as the Jaccard similarity coefficient, is calculated as the ratio of the size of the intersection between the two sets to the size of their union. It is essentially the same as IoU, but sometimes computed slightly differently. A statistic used to assess the

similarity and diversity of sample sets is the Jaccard index, sometimes referred to as the Jaccard similarity coefficient and Intersection over Union. Like accuracy, the Dice score penalizes for false positives that the algorithm detects in addition to counting how many positives you find.

$$Dice = 2 * \frac{tp}{tp + fp} + (tp + fn)$$

4) Visualization and Post-processing:

To evaluate the implemented model for skin lesion segmentation, predictions are made on unknown data which have not been trained upon. We apply extra postprocessing approaches, for example, to improve lesion boundary feature visibility and then to adjust segmentation mask projections afterward the segmentation process. The proposed method is also suitable for smoothing segmentation masks and increasing visual interpretation accuracy with imprecise lesion boundaries. Thus, these after-processing procedures make the skin cancer detection method based upon the accurate segmentation results both reliable and aesthetically appealing.

5) Application of Masks

To precisely define skin lesions for subsequent classification tasks, we employed the segmentation masks produced by the U-Net model in the last segmentation stage. Precise lesion borders were provided by the segmentation masks, which made feature extraction and classification easier. The application of segmentation masks generated by the U-Net model for accurate delineation of skin lesions is illustrated in Figure 3.

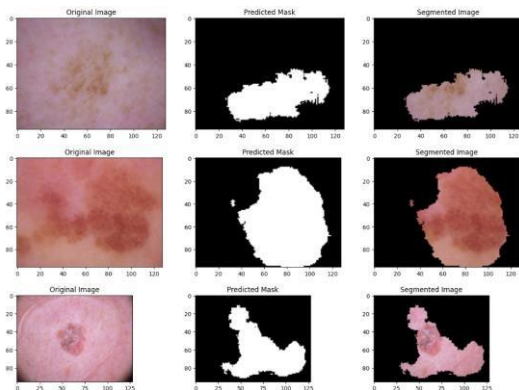


Fig. 3. Predicted Mask and Segmented Images

D. Classification using VIT:

Several separate segments were produced when we trained a model to segment VIT images. The VIT model was then trained to distinguish between melanoma and melanocytic nevi using these segments. Accuracy measurements displayed in tables and figures helped to clarify the findings. The segmentation-based classification strategy is shown to be reliable and successful by this thorough analysis. Significant progress toward improving our understanding and identification of skin lesions has been accomplished through this research, likely leading to improvements in the results of dermatological healthcare.

To train our classification model, segmented images were sent to a Vision Transformer (ViT) model. This method took advantage of ViT models' ability to efficiently handle image segmentation tasks. Our model was trained with the segmented representations to discriminate between two classes: melanoma and melanocytic nevi. This approach demonstrates a possible path toward increasing binary classification accuracy in medical image analysis by leveraging the advantages of both transformer-based models and segmentation. From this model the training accuracy is 0.73 and validation accuracy is 0.73.

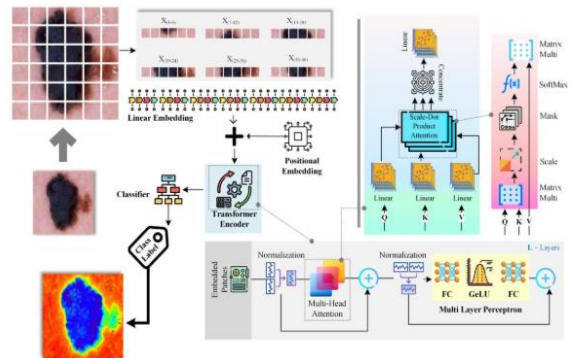


Figure.4. (Vision Transformer-based Skin Cancer Classification Model) [13]

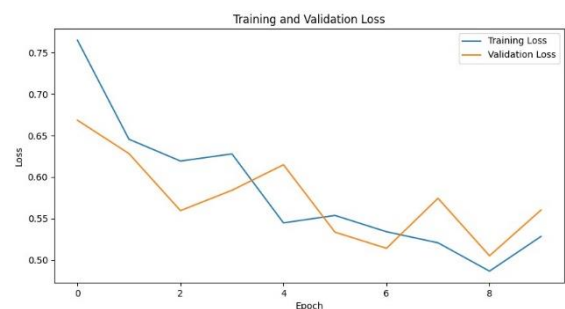


Figure.5. (Training & validation loss)



Figure.6. (Training & validation Accuracy)

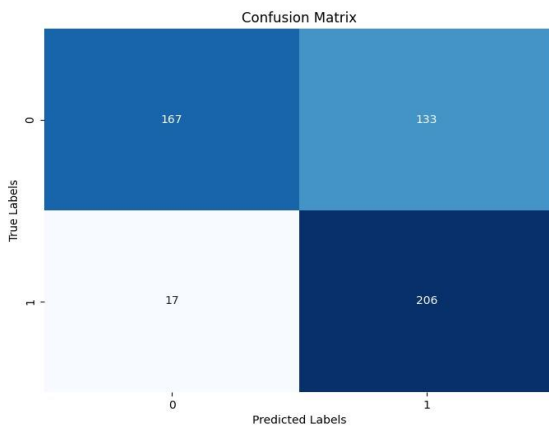


Figure.7. Confusion Matrix

The confusion matrix above Thirteen of the actual class one 167 images were wrongly classified as class 0 and 133 were accurately predicted as class 1 and 206 images that were truly class 0 were accurately classified, whereas 17 images were mistakenly classed as class 1.

E. Classification using CNN:

Because of dataset imbalances, we trained a CNN model for a 7-class classification challenge using the segmented images. This all-inclusive strategy sought to address the data's unpredictability more successfully. A comprehensive table and figure below give the results of a thorough analysis of the resulting categories, together with pertinent metrics. By providing insights into how CNN models can handle unbalanced datasets and improve classification accuracy across several classes, this approach makes a substantial contribution to the area of medical image analysis.

It uses pre-trained weights from ImageNet to initialize MobileNet, leaving out the fully connected layers. It reduces geographic dimensions by adding a layer of global average pooling. We adds a completely connected layer with ReLU activation and 256 units at the end. It includes softmax activation and an output layer with 7 units (for a classification problem with 7 classes).

builds a new model with the custom classification layers and the input from the MobileNet. Essentially, it adds unique layers for classification to modify the robust MobileNet architecture for a particular picture classification application.

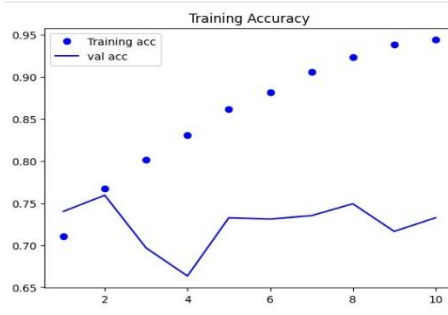


Figure.8. Training and Validation Accuracy

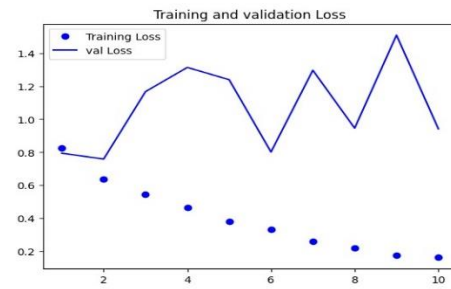


Figure.9. Training and Validation Loss

IV. RESULTS

The VIT model demonstrated a high training accuracy of 0.94, indicating its effectiveness in learning from the training data. However, its validation and testing accuracies were at 0.73. The VIT model still achieved a notable testing accuracy, indicating its capability to generalize well to unseen data.

TABLE 1 -CLASSIFICATION USING VIT

Training Accuracy	0.94
Validation Accuracy	0.73
Testing Accuracy	0.73

On the other hand, the CNN model showed a comparable training accuracy of 0.95 to the VIT model, implying its ability to learn the training data effectively. However, its validation and testing accuracies were lower at 0.73 respectively.

TABLE 2 -CLASSIFICATION USING CNN

Training Accuracy	0.94
Validation Accuracy	0.73
Testing Accuracy	0.73

V. CONCLUSION

In conclusion, this work presents a method for skin lesion segmentation and classification using state-of-the-art deep learning techniques. Using Convolutional Neural Networks (CNN) and Vision Transformer (VIT) models for classification and the U-Net technique for segmentation, we have shown encouraging results in correctly classifying different kinds of skin lesions. The metrics acquired for segmentation accuracy verify the efficacy of our method in medical picture analysis. Our approach has a lot of potential to boost clinical outcomes by increasing the precision and dependability of skin cancer diagnosis. better work will concentrate on enlarging the classification task to support a wider variety of lesion classes, and continuous efforts will be made to better hone and optimize our methodology. This work represents a major advancement in the use of deep learning to dermatological analysis and emphasizes the need for ongoing research and development in this area.

VI. REFERENCES

- [1] S. A. R. Mohammad Ali Kadampur, "Skin cancer detection: Applying a deep learning based model driven architecture in the cloud for classifying dermal cell images," *Informatics in Medicine Unlocked (IMU)*, p. 7, 2020.
- [2] M. A. a. H. K. M. a. D. R. a. E. M. M. Kassem, "Machine Learning and Deep Learning Methods for Skin Lesion Classification and Diagnosis: A Systematic Review," *Diagnostics*, vol. 11, 2021.
- [3] R. & A. A. Ahmadi Mehr, "Skin Cancer Detection Based on Deep Learning.," *Journal of biomedical physics & engineering*, vol. 12(6), pp. 559-568, 2022.
- [4] A. a. G. P. a. P. R. Dzieniszewska, "Skin Lesion Classification Based on Segmented Image," pp. 1-6, 2023.
- [5] S. A. Seeja R D1*, "Deep Learning Based Skin Lesion Segmentation and," p. 7, 2019.
- [6] R. L. a. R. R. d. A. L. a. R. J. J. P. C. a. S. R. R. V. e. Araújo, "Automatic Segmentation of Melanoma Skin Cancer Using Deep Learning," in *2020 IEEE International Conference on E-health Networking, Application & Services (HEALTHCOM)*, 2021, pp. 1-6.
- [7] M. a. M. K. Taghizadeh, "The Fast and Accurate Approach to Detection and Segmentation of Melanoma Skin Cancer using Fine-tuned Yolov3 and SegNet Based on Deep Transfer Learning," *arXiv preprint arXiv:2210.05167*, 2022.
- [8] L. a. K. J. a. A. E. a. K. A. a. F. M. a. F. D. Bi, "Dermoscopic image segmentation via multistage fully convolutional networks," *IEEE Transactions on Biomedical Engineering*, pp. 2065--2074, 2017.
- [9] M. a. A. J. a. S. M. a. A. M. A. a. M. G. A. a. K. S. Zafar, "DeepLabv3+-based segmentation and best features selection using slime mould algorithm for multi-class skin lesion classification," *Mathematics*, p. 364, 2023.
- [10] N. D. N. Dimsa, "Melanoma multi class segmentation using using different U-Net type," 2021.
- [11] { . A. a. A. R. a. M. I. a. T. S. Marriam Nawaz and Zahid Mehmood and Tahira Nazir and Naqvi, "Skin cancer detection from dermoscopic images using deep learning and fuzzy k-means clustering," *Microscopy Research and Technique*, vol. 85, pp. 339--351, 2022.
- [12] A. A., "A Deep Learning Approach to Skin Cancer Detection in Dermoscopy Images," *Journal of Biomedical Physics & engineering*, vol. 10, pp. 801-806, 2020.
- [13] G. M. S. I. M. M. A.-A. K. A. K. S. I. & S. M. K. U. Himel, "Skin Cancer Segmentation and Classification Using Vision Transformer for Automatic Analysis in Dermoscopy-Based Noninvasive Digital System.," *International journal of biomedical imaging*, 2024.
- [14] P. a. R. C. a. K. H. Tschandl, "The HAM10000 dataset, a large collection of multi-source dermoscopic images of common pigmented skin lesions," *Scientific data*, pp. 1-9, 2018.
- [15] O. a. F. P. Ronneberger, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Springer*, pp. 234-251, 2015.

Stock Market Analysis and Prediction

Safiullah Muhammad
 Department of Computer Systems
 Engineering
 University of Engineering and
 Thechnology Peshawar
 Peshawar, Pakistan
qazisafiullah.7@gmail.com

Madiha Sher
 Department of Computer Systems
 Engineering
 University of Engineering and
 Thechnology Peshawar
 Peshawar, Pakistan
madiha@uetpeshawar.edu.pk

Kashan Muhammad
 Department of Computer Systems
 Engineering
 University of Engineering and
 Thechnology Peshawar
 Peshawar, Pakistan
20pwcse1892@uetpeshawar.edu.pk

Adeel Rehman
 Department of Computer Systems Engineering
 University of Engineering and Thechnology Peshawar
 Peshawar, Pakistan
itsadeel456@gmail.com

Yasir Saleem Afridi
 Department of Computer Systems Engineering
 University of Engineering and Thechnology Peshawar
 Peshawar, Pakistan
yasirsaleem@uetpeshawar.edu.pk

Abstract— This paper is about an innovative application which is designed to fill the gap between traditional stock market analysis and cutting-edge predictive modeling. Real-time information is important for financial decision-making, this application aims to provide investors, traders, and financial analysts with a powerful toolset for informed decision-making and help them to invest in stocks intelligently. The significance of this application lies in its ability to democratize access to sophisticated stock analysis and prediction tools. By doing real-time data analysis, having user-friendly interfaces, and advanced predictive modeling, the software provides a good solution for users ranging from novice investors to seasoned financial professionals. The application not only addresses the challenges associated with fragmented data and delayed analysis but also opens avenues for continuous monitoring and optimization of predictive models in response to dynamic market conditions. The Analysis part allows users to analyze the stock data which will be very helpful for them to gain insights from the stock data. The Prediction part introduces advanced machine learning techniques to develop predictive models capable of forecasting stock prices and trends. These models are seamlessly integrated into the application developed in the Analysis Phase, providing users with real-time predictions and valuable insights. The predictive capabilities empower users to anticipate market movements and make proactive decisions, enhancing their overall investment strategies. Long Short-Term Memory (LSTM) network models have been demonstrated to perform well in stock price prediction by prior research, and most people regard them as one of the most accurate prediction methods, particularly when used for longer prediction ranges. After performing pre-processing steps like data normalization, we employed an LSTM network model in this application. Through training and testing, we determined the ideal settings for the optimizer, dropout, batch size, epochs, and other parameters. The outcome of comparing the LSTM network model with eXtreme gradient boosting (XGBoost), linear regression, last value, and moving average indicates that while it performs well for long-term forecasting, it is not suitable for short-term forecasting.

Keywords: "Stock Market; Predictive Modeling; Deep Learning; LSTM, Data Analysis; Financial Technology".

I. INTRODUCTION

In the ever-evolving landscape of financial markets, the ability to comprehend and forecast stock trends is a critical facet for investors, analysts, and financial institutions. As markets become increasingly dynamic and

interconnected, traditional methods of analysis are often challenged to provide timely and accurate insights. This study sets out to exploit the capabilities of deep learning, namely Long Short-Term Memory (LSTM) networks, in order to tackle the intricacies, present in stock market data.

II. BACKGROUND:

The development of technology and the growing complexity of international financial markets have had a profound impact on the field of stock market analysis. Conventional approaches to stock data analysis frequently struggle to deliver thorough and timely insights, which puts investors and financial experts at a disadvantage when it comes to making wise judgements. Because of the stock market's volatility, a more complex strategy combining predictive modelling and real-time data analysis is required.

The goal of this project is to close these gaps and offer a solution that gives customers access to a thorough and intuitive platform for analyzing and predicting market data. More precise trend analysis and prediction may be possible thanks to the advances in machine learning and data science, which have created new opportunities for the meaningful pattern extraction from large datasets.

A. Navigating the Evolving Landscape of Stock Market Dynamics with Advanced Analysis and Prediction Tools:

Technology and the complexity of today's financial markets have caused a revolutionary change in the field of stock market analysis. The needs of modern investors and financial professionals are not adequately satisfied by traditional techniques of analyzing stock data, which frequently leads to delayed insights and less-than-ideal decision-making. Because the stock market is so volatile, a more advanced strategy that seamlessly combines predictive modelling and real-time data analysis is required.

B. Contextualizing the Challenges:

Technology and the complexity of today's financial markets have caused a revolutionary change in the field of stock market analysis. The needs of modern investors and financial professionals are not adequately satisfied by traditional techniques of analyzing stock data, which frequently leads to delayed insights and less-than-ideal

decision-making. Because the stock market is so volatile, a more advanced strategy that seamlessly combines predictive modelling and real-time data analysis is required.

C. The Django Advantage:

Our solution is built around the robust web framework Django, which offers a solid base for creating an integrated utility. Our project aims to overcome the drawbacks of conventional methods by utilizing Django's capabilities in conjunction with sophisticated data analysis and machine learning packages. This combination not only solves the problems caused by incomplete data and slow analysis, but it also brings predictive modelling, which allows users to foresee market movements and take well-informed decisions instantly.

D. User-Centric Design:

This study is important because of its user-centered design, which targets a wide range of readers, from seasoned financial experts to novice market participants. Our project places a high priority on a user-friendly interface that incorporates real-time data analysis and predictive modelling using data obtained from APIs or other sources.

LITERATURE REVIEW

E. Overview of Literature Review:

This research is situated at the nexus of financial analysis and technology innovation. Through the application of web development, deep learning, and data analysis, the project aims to close the gap between state-of-the-art predictive modelling and conventional stock market analysis.

The background of the research is further reinforced by the growing use of technology in financial decision-making. Investors are actively looking for tools that allow them to make preemptive decisions using predictive analytics, in addition to providing a retrospective picture of market movements. In order to satisfy this need, this application offers a solution that integrates data collecting, analysis, and predictive modelling into a single, cohesive programme.

F. Theoretical Foundations:

By utilizing the Hidden Markov Model (HMM) to estimate stock prices for the four distinct airlines depicted in Figure 1, Hassan and Nath made a significant contribution to the field of stock market forecasting [1]. Notably, the opening, closing, highest, and lowest prices—four crucial components of stock prices—were captured by the authors by narrowing down the model's states. Their method stands out for not relying on expert knowledge, which makes it possible to build a prediction model without the need for specific domain knowledge. Nonetheless, it is imperative to recognize certain constraints identified in their research, namely its limitation to the airline sector and assessment on a rather limited dataset. Although their application's specialization offers insightful information about the aviation industry, its generalizability may be restricted to more general stock market circumstances. Additionally, as the authors limited

the time frame for training and testing datasets to a maximum of two years, it is imperative to take the evaluation period into account. This period provides a useful benchmark for our assessment and enables a comparison with alternative methods in the field of stock market forecasting.

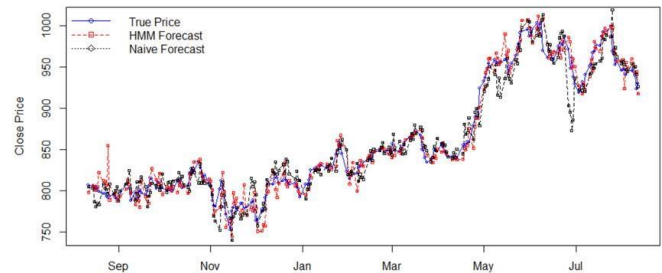


Figure 1: Hidden Markov Model for stocks prediction

G. Related Works:

More recently, a paradigm shift in stock market prediction has been brought about by the rise of deep learning. Recurrent neural networks (RNNs) and long short-term memory (LSTMs) are two deep learning models that have proven to be exceptionally adept at identifying complex patterns in sequential data [2]. The financial research community has taken a keen interest in these models due to their capacity to automatically learn hierarchical representations of data without the need for manual feature engineering.

Ya Gao, Rong Wang, and Enmin Zou's study, which was carried out at the Central University of Finance and Economics in Beijing, China's School of Public Finance and Taxation, the School of Computer Science and Technology at Xidian University in Xi'an, China, and the School of Electronics and Information at Xi'an Jiaotong University in Xi'an, China, [3], uses a variety of technical indicators, including financial data and indicators of investor sentiment. The authors utilize sophisticated dimension reduction methods, utilizing deep learning techniques like PCA and LASSO, to reduce the multiplicity of factors that impact returned stock values.

In addition, a thorough analysis of the effectiveness of two well-known deep learning architectures—LSTM and Gated Recurrent Unit (GRU)—in the context of stock market forecasting is included in the study. This comparative investigation is carried out using several parameter values in order to fully assess the prediction abilities of both GRU and LSTM. The results of this study demonstrate that LSTM and GRU are equally efficient in predicting stock values, underscoring the versatility and potency of both deep learning techniques in the field of financial forecasting [4].

By using an optimized artificial neural network (ANN) model, Qiu and Song [5] developed a predictive approach for predicting the direction of the Japanese stock market. The authors' study combined artificial neural network-based models with genetic algorithms (GAs), naming the resulting

framework a hybrid GA-ANN model. This novel method increases the forecasting accuracy of stock market movements by combining the learning capabilities of artificial neural networks with the evolutionary optimization capabilities of genetic algorithms.

The study by V.V. Kranthi Sai Reddy emphasizes how important stock trading is on a worldwide level [6]. The endeavour to forecast future prices of different financial instruments traded on currency exchanges is known as share market prediction. According to the author, technical and fundamental analysis are typically combined to create stock projections in modern finance. These studies form the basis of stockbrokers' basic methods.

In this paper, machine learning algorithms based on Python are used to realise the predictive capabilities of the system. The suggested strategy promotes the use of machine learning techniques, in which the model is trained on stock data that is already available in order to gain intelligence and then apply this knowledge to produce precise forecasts. The study uses a machine learning system called the Support Vector Machine to operationalize this method. Specifically, three different marketplaces are used to anticipate the prices of large and minor equities using this approach. The forecasting model covers a range of time periods, including daily and minute-by-minute price fluctuations [7].

Prof. S.P. Pimpalkar together with co-authors Jenish Karia, Muskaan Khan, Satyamandand, and Tushar Mukherjee are credited with a combined effort on the scholarly publication titled "Stock Market Forecasts Using Machine Learning" [8]. With a focus on using a wide range of attributes as input for building predictive models, the study presents a novel method to stock market prediction. Predicting whether the market value will show a positive or negative trend is these models' main goal.

The study utilizes a variety of machine learning approaches, including Regression, Support Vector Machine, and Recurrent Neural Network methods. This unique combination of machine learning techniques highlights the writers' dedication to investigating several strategies, each designed to tackle a particular facet of the challenging challenge of stock market forecasting. Regression models are employed to ascertain linear associations in the data, whilst Support Vector Machine and Recurrent Neural Network models aid in identifying non-linear patterns and temporal dependencies [9].

RNN and LSTM models were used by McNally et al. [10] to forecast Bitcoin values. They used a feature engineering method called Boruta algorithm, which is similar to random forest classifier. The authors used Bayesian optimization to adjust the LSTM parameters in addition to feature selection. The study's primary emphasis was a Bitcoin dataset that ran from August 19, 2013, to July 19, 2016. Several optimization techniques were used to improve the performance of deep learning models; nonetheless, overfitting proved to be a significant obstacle. There are similarities between the study problem McNally et al. addressed and stock market price prediction, especially when it comes to handling noise and

hidden elements in the price data. The study question was formulated by the authors as a time sequence problem. Their work is notable for the careful consideration they provide to the feature engineering and optimization procedures. We might be able to duplicate the techniques used in these areas in our own data preprocessing stage.

H. Comparison with Related Works:

Their meticulous attention to detail in feature engineering and optimization processes makes their work stand out. It's possible that we could replicate these areas' methods in our own data preprocessing phase.

HMMs became popular in the early 2000s for stock price forecasting, as Hassan and Nath [12] showed. HMMs are useful in certain situations, but they have trouble identifying intricate, non-linear correlations in data. On the other hand, our approach makes use of LSTMs' deep learning capabilities to automatically identify complex patterns in sequential stock data without the need for manual feature engineering.

In contrast to traditional machine learning models, the paper by Weng et al. [13] explores short-term stock price prediction using ensemble methods, incorporating neural network regression ensembles, Random Forests, AdaBoost, and support vector regression ensembles. While ensemble methods are robust, they might lack the capacity to capture nuanced temporal patterns present in stock data. Our LSTM-based approach, being a deep learning model, inherently excels at capturing temporal dependencies and has the p Furthermore, the LSTM model we propose extends beyond the limitations faced by Overall, our methodology, grounded in LSTM networks, strives to overcome the limitations and enhance the predictive capabilities observed in existing works. The deep learning paradigm, particularly with LSTM networks, proves to be a promising avenue for advancing the accuracy and adaptability of stock market predictions.

III. METHODOLOGY

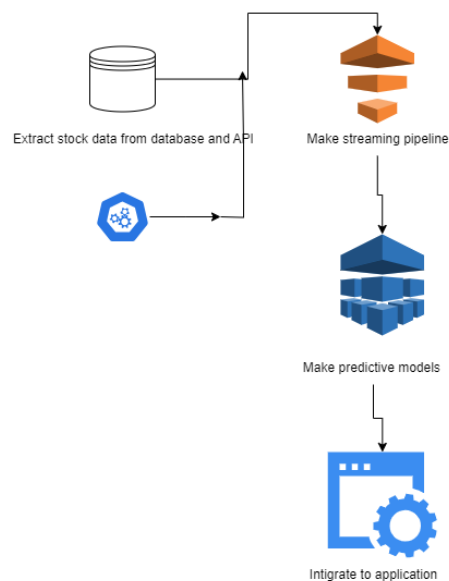


Figure 2: Flow chart of implementation

As shown in Figure 2, we got our data using APIs and then made streaming pipelines for analysis of data and then made predictions from that data.

1) Analysis Phase:

a) Data Collection:

In the initial stage, we gathered stock data from various sources, fetching data from APIs and implemented web scraping techniques. This step ensures a comprehensive and diverse dataset for analysis and prediction. The utilization of APIs ensures a structured and organized retrieval of real-time market data, while web scraping allows us to extract valuable insights from a broader spectrum of sources.

b) Data Transformation and Streaming:

Once collected, the raw data is transformed to cleanse and preprocess collected data from different sources so that it can be used for analysis and predictive purposes. Mechanism is developed in such a manner so that stock related information keep on updating on real time in the application and investors can have informed and intelligent decisions. Streaming pipeline like ETL is created to transfer real time data using Apache Kafka and to get good stock data ready for analysis.

c) Interactive Application:

For the facilitation of the users, we develop an interactive application so that users can have seamless view of real time stock data.. This would allow users to visualize and analyze key metrics, trends, and patterns in the real time stocks.

2) Prediction Phase:

a) Predictive Model Development:

In this phase, we focused on building a robust predictive model using large datasets. Leveraging the power of deep learning, particularly LSTM networks, we constructed a model capable of capturing complex temporal dependencies and patterns in stock data.

Many factors affect stock prices, and these factors change over time. Since stock values are continuously changing due to market sentiment, economic events, and other external causes, long short-term memory (LSTM) models are excellent in learning and adapting to the temporal dynamics of sequential data.

Our prediction model is the result of a painstaking process that includes complex algorithms, architecture fine-tuning, and model parameter calibration. We hope to use deep learning to not only forecast stock movements but also to understand the intricate dynamics and underlying complexity of the financial system.

b) Model Integration:

Our application architecture effortlessly incorporates the established prediction model. Thanks to this connectivity, users may use the same platform to forecast future stock movements in addition to analyzing previous data.

c) Model Evaluation and Optimization :

We carry out extensive assessments to guarantee our forecasting model's precision and dependability. Performance is evaluated using a variety of metrics, such as mean absolute error (MAE), recall, and precision. We also investigate optimization strategies, modifying model parameters and architecture to improve prediction performance.

d) Deployment:

Deployment of the application on platforms that align with security standards, making it accessible to users and stakeholders.

e) Continuous Monitoring and Improvement:

Establishment of mechanisms for continuous monitoring of the application's performance, with the ability to introduce updates and improvements based on user feedback and evolving market conditions.

f) Optimization and Scalability:

Optimization of the application and predictive models for performance, scalability, and efficiency, providing a smooth user experience.

g) Testing and Quality Assurance:

Rigorous testing to identify and rectify bugs or issues, ensuring the reliability and accuracy of the application and predictive models.

IV. RESULT AND DISCUSSION

Four additional methods were employed in this experiment to generate predictions in this study, which were then contrasted with the outcomes of applying the LSTM neural network model. The forecast outcomes are displayed in Figures 1 through 7:

Figure 3 displays the outcome of predicting using Last Value. The most economical forecasting model is this one. We can see that the number from the day before (green cross) is all that is needed to predict each day (red cross). A common benchmark for assessing more intricate models is Last Value.

Figure 4 displays the outcome of the prediction process using linear regression. We can draw the conclusion that this approach performs a poor job of capturing swings in stock prices.

Figures 5, 6, and 7 display the prediction results obtained by employing Moving Average, XGBoost, and LSTM, in that order. The iterative process of turning weak learners into strong learners involves the use of the XGBoost algorithm. The projected value in the moving average approach is equal to the average of the first N values. The figures show that there is little variation in the prediction outcomes of the three methods; however, the LSTM and XGBoost prediction results, respectively, are marginally better than the Moving Average prediction results, with comparatively small errors between the test and predicted values.

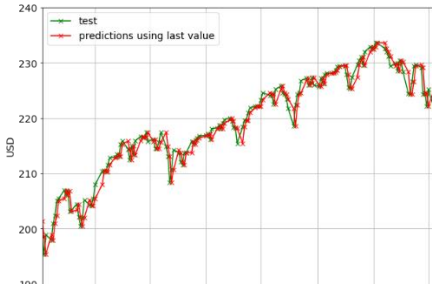


Figure 3: Forecast results of Last Value

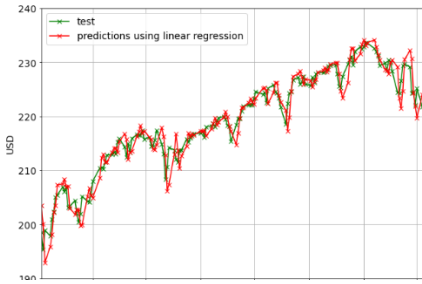


Figure 4: Forecast results of Linear Regression

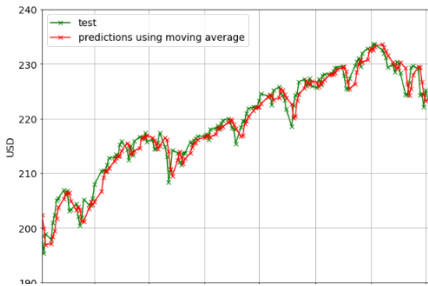


Figure 5: Forecast results of Moving Average

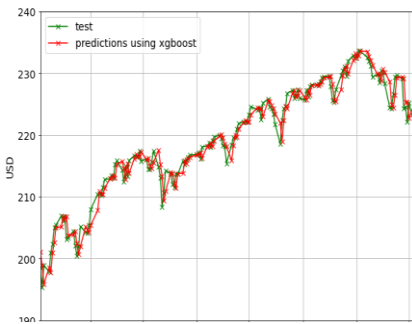


Figure 6: Forecast results of XGBoost



Figure 7: Forecast results of LSTM

With the exception of the poor linear regression prediction results, the predictions of the other four models are hard to assess from direct observation; these are combined into a single graph for comparison, as seen in Figure 8.

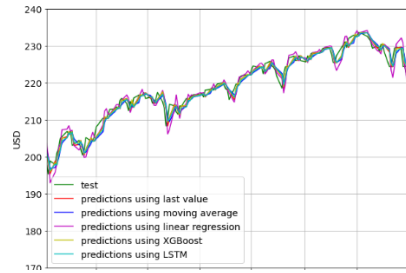


Figure 8: Forecast results of 5 methods

The prediction outcomes are assessed in the paper using mean absolute percentage error (MAPE) and root mean square error (RMSE).

RMSE can accurately reflect the precision of a measurement since it is sensitive to both extremely large and very small errors in a sequence of measurements. Equation (1)'s RMSE formula is displayed below, with y_i representing the neural network's output and y denoting the true value [14].

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - y)^2}{n}} \quad (1)$$

The prediction techniques' performance is evaluated using MAPE. In the field of machine learning, MAPE is also used to quantify the accuracy of predicting techniques by presenting the result as a percentage. The following represents the MAPE formula for equation (2), where y_i denotes the neural network's output value and y denotes the true value.[14]

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y - y_i}{y} \right| \times 100 \quad (2)$$

The RMSE and MAPE values of the five methods are shown in Table 1.

Table 1. RMSE of five methods

Method	RMSE	MAPE(%)
Last Value	1.69	0.59
Moving Average	1.89	0.69
Linear Regression	2.27	0.78
XGBoost	1.65	0.59
LSTM	1.75	0.63

Table 1 presents a comparison of the RMSE and MAPE of the five approaches, indicating that XGBoost and Last Value

yield the best prediction outcomes with lower RMSE and MAPE. Compared to the other more involved approaches, Last Value produces superior prediction results, particularly surpassing the performance of the LSTM neural network model. The fact that the prediction range is only one may account for this finding. The other approaches might work better than the Last Value if the prediction range is longer.

There's also the problem of data size. Being a neural network, the LSTM has to be trained on a lot of data, just like any other neural network. With XGBoost, the most accurate forecasts are produced. Second-order Taylor expansions are applied to the loss function by the XGBoost. The second-order derivative is introduced in this method in order to improve accuracy and enable modification of the loss function. This virtually eliminates the time lag issue, which is a frequent LSTM downside.

V. CONCLUSION

The stock market is seeing a sharp increase in trading, and analysts and investors are keen to discover a way to accurately forecast future patterns in the market. Numerous research conducted in recent years have demonstrated the effectiveness of LSTM neural network models in stock market prediction. Additionally, when compared to other machine learning algorithms, LSTM neural network models outperform other algorithms when used for longer prediction horizons.

In this paper, we find that the LSTM neural network model is not a flawless prediction approach by comparing its performance in creating short-term forecast range predictions with the performance of the other four algorithms. It must be acknowledged, nonetheless, that in long-term forecast range prediction, the LSTM neural network model performs better at capturing trends and seasonality. As a result, more researchers will be motivated to employ cutting-edge approaches to discover novel forecasting strategies that can be used in a wider range of scenarios. This will benefit analysts, investors, and anybody else with an interest in the stock market by giving them solid insight into the market's future.

VI. ACKNOWLEDGMENTS

I would like to acknowledge and give my warmest thanks to my university teachers who made this work possible. Their teachings, guidance and advice carried me through all the stages of writing my project.

VII. REFERENCES

- [1]. Hassan, M. R., & Nath, B. (2005, September). Stock market forecasting using hidden Markov model: a new approach. In *5th international conference on intelligent systems design and applications (ISDA'05)* (pp. 192-196). IEEE.
- [2]. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research*, 270(2), 654-669.
- [3]. Martinez, J., Qian, B., Wang, S., & Zou, H. F. (2006). *Local Public Finance in China: Revenues of Local Governments* (No. 551). China Economics and Management Academy, Central University of Finance and Economics.
- [4]. Gao, Y., Wang, R., & Zhou, E. (2021). Stock prediction based on optimized LSTM and GRU models. *Scientific Programming*, 2021, 1-8.
- [5]. Qiu, M., & Song, Y. (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS one*, 11(5), e0155133.
- [6]. "Stock Market Prediction Using Machine Learning" V Kranthi Sai Reddy, Researchgate
- [7]. Patel, A., Patel, D., & Yadav, S. (2021, May). Prediction of stock market using artificial intelligence. In *Proceedings of the 4th international conference on advances in science & technology (ICAST2021)*.
- [8]. Sachin Sampat Patil, Prof. Kailash Patidar, Asst. Prof. Megha Jain, "A Survey on Stock Market Prediction Using SVM ", CTET 2016.
- [9]. McNally, S., Roche, J., & Caton, S. (2018, March). Predicting the price of bitcoin using machine learning. In *2018 26th euromicro international conference on parallel, distributed and network-based processing (PDP)* (pp. 339-343). IEEE.
- [10]. Weng, B., Lu, L., Wang, X., Megahed, F. M., & Martinez, W. (2018). Predicting short-term stock prices using ensemble methods and online data sources. *Expert Systems with Applications*, 112, 258-273.
- [11]. Sachin Sampat Patil, Prof. Kailash Patidar, Asst. Prof. Megha Jain, "A Survey on Stock Market Prediction Using SVM ", CTET 2016.
- [12]. Nguyen, N. (2017). An analysis and implementation of the hidden Markov model to technology stock prediction. *Risks*, 5(4), 62.
- [13]. Liu, F., Weng, C., & Yu, H. (2019). Advancing clinical research through natural language processing on electronic health records: traditional machine learning meets deep learning. *Clinical Research Informatics*, 357-378.
- [14]. Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E., & S, S. (2020). Deep learning for stock market prediction. *Entropy*, 22(8),

Land Subsidence in Quetta Valley: A Study using PyGMTSAR Library and SBAS Technique

Hadia Khan

National Center in Big Data and Cloud
Computing (NCBC), University of Engineering
and Technology (UET), Peshawar.
Department of Computer System Engineering, UET
Peshawar
hadiakhan@uetpeshawar.edu.pk

Dr. Nasru Minallah

National Center in Big Data and Cloud
Computing (NCBC), University of
Engineering and Technology (UET),
Peshawar.
Department of Computer System
Engineering, UET Peshawar
n.minallah@uetpeshawar.edu.pk

Waleed Khan

National Center in Big Data and Cloud
Computing (NCBC), University of
Engineering and Technology (UET),
Peshawar.
Khanwaleed@uetpeshawar.edu.pk

Abstract— This paper presents the analysis of land subsidence as a result of excessive groundwater abstraction for irrigation and domestic supply in Quetta valley, Pakistan. The study reveals that the rate of land subsidence in the area has been steadily increasing over the past few decades, leading to significant damage to infrastructure and posing a threat to the safety of residents. Our research harnesses the power of the PYGMTSAR library, a cutting-edge resource in synthetic aperture radar (SAR) data analysis, to explore the intricacies of surface deformation using the Small Baseline Subset (SBAS) technique. This analysis will provide valuable insights for implementing effective strategies to mitigate the adverse effects of land subsidence in the Quetta valley. Observation of surface deformation using the Small Baseline Subset (SBAS) Interferometric Synthetic Aperture Radar (InSAR) technique on Sentinel-1 datasets from 2020 to 2023 reveals subsidence up to of 0.13m to 0.15m and for the year 2023 with 10 datasets land subsidence upto 0.06m and -0.394m.

Keywords: Land Subsidence; PYGMTSAR; SBAS, InSAR.

I. INTRODUCTION

The Quetta Valley is located in western Pakistan, in a 560 km-long and 150 km-wide complex belt of north–south-oriented mountain ranges and intervening valleys [1]. This paper presents the analysis of land subsidence as a result of excessive groundwater abstraction for irrigation and domestic supply in Quetta valley, Pakistan. Land subsidence, the gradual sinking or lowering of the Earth's surface, is a pressing issue in many urban areas worldwide, including Quetta, the provincial capital of Balochistan, Pakistan. Quetta, situated in a tectonically active region, faces significant challenges related to land subsidence due to a variety of factors, including rapid urbanization, groundwater extraction, and natural geological processes. The adverse effects of land subsidence in Quetta are exacerbated by its dense population, extensive infrastructure, and reliance on groundwater for agricultural and domestic purposes.

The primary objective of this study is to analyze land subsidence in Quetta Valley using an advanced InSAR time-series technique, namely the Small Baseline Subset (SBAS) approach, and to validate the research outcomes with field observations. To achieve the aforementioned goals, the specific objectives of this research are as follows: to generate a land subsidence map in Quetta Valley during a certain observation period using the PyGMTSAR library in Python; to explore the possibility of correlating the land subsidence pattern with land use and land cover changes in the research

area; and to identify potential areas that are affected by land subsidence and carry out in-depth field investigations in those places. It is expected that the research outcomes could provide valuable references for the authority to formulate development plans and land management strategies in Quetta Valley.

There are two leading causes of subsidence: surface processes, mostly anthropogenic, and tectonics. Most of the subsidence in populated areas is caused by anthropogenic processes, such as groundwater extraction [2] so numerous studies on subsidence have been conducted primarily in developed countries. However, subsidence in populated centers in developing countries has not been significantly studied due to a lack of data. Figure 1 below shows the generated global map of land subsidence which covers historically documented and new subsiding areas. We estimate that more than 6.3 million square kilometers of the global land is influenced by significant subsidence rates. That includes 231,000 square kilometres of urban and dense settlement areas and a population of nearly 2 billion [3]. Overall, South Asia is modeled to have the greatest extent of land under threat of subsidence (2.2% of its total area experiencing subsidence rates greater than 50 mm/y) as well as the largest number of people to be 5/7 affected by it (20 million). Other countries with subsidence rates exceeding 50 mm/y include the Philippines, Iran, Costa Rica, Indonesia and Uzbekistan.

To address the complexities of land subsidence in Quetta, advanced remote sensing techniques offer valuable insights and tools for monitoring and analysis. Among these techniques, the Small Baseline Subset (SBAS) method, integrated with the Persistent Scatterer Interferometry (PSI) approach, stands out as a robust and effective tool for detecting and quantifying ground movements with high precision and spatial resolution.

In this paper, we focus on the application of the SBAS technique using data from the PyGMTSAR (Python-based Generic Mapping Tools Synthetic Aperture Radar) platform to investigate land subsidence in Quetta. SBAS, Small Baseline Subset, is a powerful technique in interferometric synthetic aperture radar (InSAR) analysis used to monitor and measure ground deformation over time. By leveraging Sentinel-1 synthetic aperture radar (SAR) data, PyGMTSAR enables the generation of precise ground deformation maps and time-series analyses, offering valuable insights into the spatial and temporal patterns of land subsidence in the region.

Through the integration of PyGMTSAR and SBAS, we aim to provide a comprehensive understanding of the dynamics of land subsidence in Quetta, including its causes, rates, and spatial distribution. Such insights are crucial for informing urban planning, infrastructure development, and groundwater management strategies to mitigate the adverse impacts of land subsidence and ensure the sustainable development of Quetta and its surrounding areas.

[4] integrated DInSAR and SBAS techniques to determine mining-related deformations in the Rydułtowy Mine in Poland. The research findings indicated that the SBAS technique could be reliable for monitoring residual subsidence in mining areas. This study provided a specific application of the SBAS method in monitoring ground deformations related to mining activities, offering insights into the potential use of SBAS for monitoring subsidence in regions with industrial activities. The study by [5] assessed the spatial and temporal evolution of surface subsidence in Lanzhou New District using Persistent Scatterer Interferometric Synthetic Aperture Radar (PSInSAR) and the Small Baseline Subset (SBAS) InSAR. This study demonstrates the applicability of

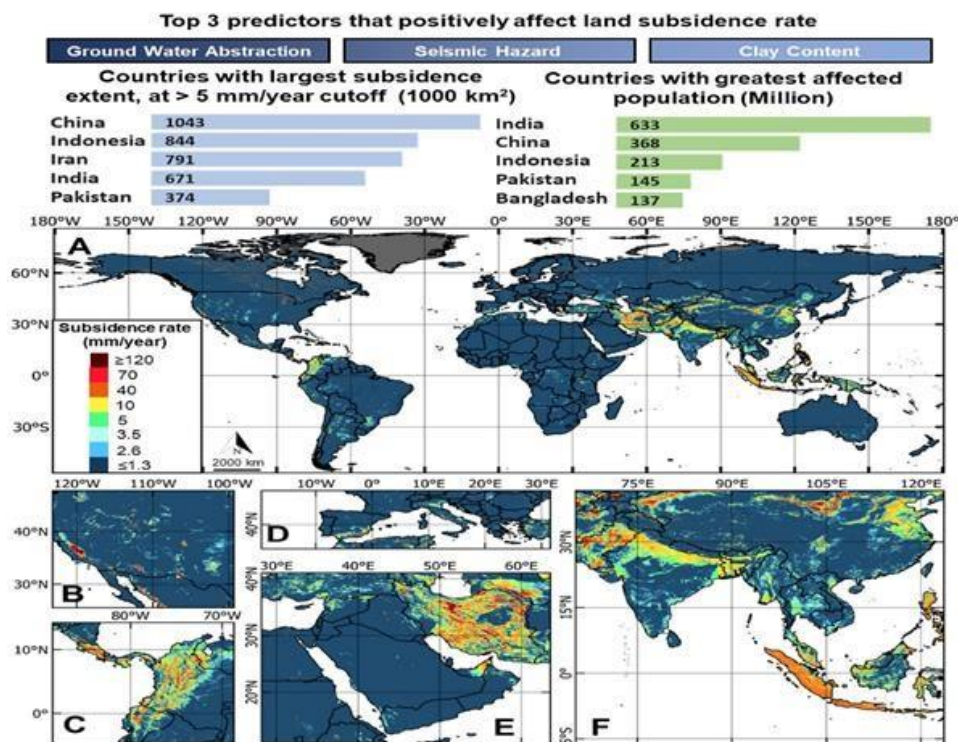


Figure 1 Global prediction of land subsidence, with relevant feature importance and zonal statistics at the top of the figure. Modeled subsidence rates for the entire globe (a), zoomed-in maps of land subsidence for North America (b), South America (c), Europe and South, East, and South-East Asia (f). [Credit: Davydzenka et al, 2024]

II. RELATED WORK

Land subsidence is a critical geohazard that can lead to infrastructure damage, groundwater depletion, and environmental degradation. To monitor and mitigate the impacts of land subsidence, various remote sensing techniques have been utilized. One of the most commonly used techniques is the Small Baseline Subset (SBAS) technique, which has been employed in different geographic locations to detect and monitor land subsidence. [6] used the evidential belief function model in GIS to map land subsidence susceptibility at Kinta Valley in Malaysia. Their study focused on the application of GIS modeling, indicating the significance of spatial analysis in understanding land subsidence.

[7] utilized the Intermittent SBAS (ISBAS) technique to study land subsidence in Mexico City. The findings highlighted the effectiveness of the SBAS technique in detecting subsidence and its potential application in urban areas.

SBAS in monitoring land subsidence, providing valuable insights into the diurnal evolution and annual variability of boundary-layer height and its correlation to other meteorological variables.

[8] conducted a study on the growing city and rapid land use transition in the Pokhara Valley, Nepal, assessing multiple hazards and risks. Although this study did not directly focus on land subsidence, it provides insights into the complex interactions between urbanization, land use transition, and environmental hazards, which are essential considerations in understanding the drivers and impacts of land subsidence.

Furthermore, it is worth noting that there is a knowledge gap in the existing literature regarding the specific application of SBAS in monitoring and assessing land subsidence in Quetta Valley. While several studies have utilized SBAS in different regions, there is a lack of comprehensive research focusing on Quetta Valley. Future research directions should aim to address this gap by applying SBAS to monitor and analyze land subsidence in Quetta Valley, providing valuable insights into the drivers, spatial patterns, and temporal evolution of land subsidence in this particular region. Additionally, further research should explore the potential impacts of land subsidence on infrastructure, agriculture, and the environment in Quetta Valley, contributing to effective mitigation and adaptation strategies.

III. TECHNIQUE

A. Interferometric Synthetic Aperture Radar (InSAR)

InSAR provides high-resolution geodetic measurements of surface deformations. A typical resolution for a phase measurement is about one-fifteenth of a phase cycle [9]. The interferometric stacking technique, Small Baseline Subset (SBAS), was used in this study to calculate surface deformation. The SBAS technique is based on a combination of differential interferograms produced by data pairs to derive surface deformation velocities by inversion of temporal phase profiles, effectively reducing atmospheric effects, topographic uncertainties, and orbital errors [10].

B. Dataset

Four descending Synthetic Aperture Radar images of Sentinel-1 SLC scenes C-Band (~5.6 cm wavelength) covering, are obtained for the Quetta Valley from 2020–2023 obtained from the Alaska Satellite Facility (ASF).

Similarly a second dataset consisting of 10 scenes as shown in the following Table 1 for a period of 2023, with area of interest being Quetta Valley having co-ordinates with Polygon [67.4045, 29.9351], [67.3548, 30.429], [66.6316, 30.4671], [66.6702, 29.9638], [67.4045, 29.9351] acquired through Alaska Satellite Facility.

design provides the users great flexibility and customization when they need to build up their InSAR processing scripts.

Table 1 : Scenes obtained from the Alaska Satellite Facility (ASF) for the study

Scene	Date	Scene	Date
1	2023-01-26	6	2023-07-13
2	2023-02-19	7	2023-08-30
3	2023-03-27	8	2023-09-23
4	2023-04-20	9	2023-11-22
5	2023-05-26	10	2023-12-28

C. PyGMTSAR (Python GMTSAR)

The PyGMTSAR library is an open-source, Python-based software package for InSAR data processing and subsidence analysis. GMTSAR is a well-known and established software for advanced InSAR processing. However, compared to the relatively complex command line-based data processing with GMTSAR, PyGMTSAR provides a user-friendly and flexible interface for the processing of InSAR data. It allows the users of all levels, from beginners to advanced researchers, to carry out various InSAR processing steps, such as data import, data selection, and data processing, with a few lines of Python scripts. And it also provides extensive functionalities for the generation of various subsidence products, like time-series and velocity maps. The compatibility and flexibility with the Python ecosystem is an obvious advantage of the PyGMTSAR library. At the data import stage, different types of InSAR data, including the general InSAR stacked images, Amplitude Change Detection (ACD) images, and coherence images, etc., could be imported and rapidly visualized on the generated multi-channel multilooked localization images. The functionality of the PyGMTSAR library is achieved through a modular design, which encapsulates different types of InSAR processing functions into various modules. This design provides the users great flexibility and customization when they need to build up their InSAR processing scripts.

For each module, corresponding processes and commands will be automatically generated using user-friendly GUI or regular command line environment. And interactive configuration windows will smartly guide the users to complete the required module configurations. The users could simply create and parameterize a dictionary, then the InSAR processing object could be built and module configurations will be done using the created dictionary in a clear, simple, and non-redundant way. At last, all the InSAR processing commands are executed in a batch manner through an automatically generated executable bash script, so that massive InSAR processing tasks could be handled efficiently. PyGMTSAR (Python GMTSAR provides Sentinel-1 Satellite Interferometry for everyone. The goal of the project is easy and fast satellite interferometry (InSAR) processing for Sentinel-1 radar scenes everywhere as on localhost as on cloud environments like Google Colab and in Docker images [11].

The SBAS procedure within PyGMTSAR encompasses several essential steps, including co-registration, interferogram generation, flattening, and Goldstein filtering. These steps are performed with optimized parameters to achieve accurate results. Furthermore, a complex multi-look operation is employed to produce a ground sampling distance of 80m. PyGMTSAR utilizes a predefined linear displacement model to jointly estimate the DEM error and the low-pass displacement parameters, ensuring robust and reliable analysis outcomes.

It focuses on the essential steps of InSAR processing that it executes. From data acquisition to reframing, image co-registration, interferogram formation, geocoding, phase

unwrapping, phase detrending, displacement map creation, displacement projection, time-series analysis, trend analysis, and data exporting, each step is thoroughly explored.

IV. RESULTS AND DISCUSSION

To visualize ground displacement, the SBAS technique implemented through the PyGMTSAR library was utilized to compute average ground velocity and net displacement at various dates. The outputs for surface deformation are depicted in Figure 2 1, Figure 2 2, Figure 2 3, illustrating the

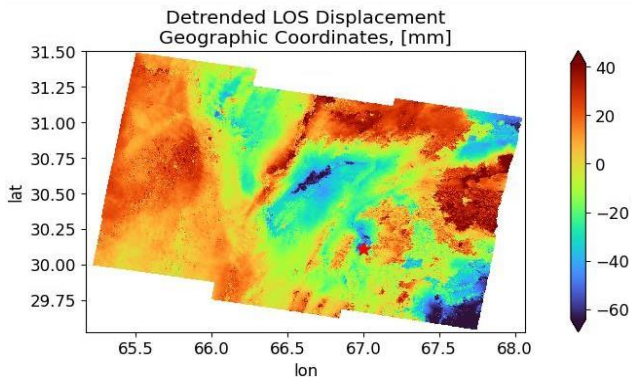


Figure 2 1 Detrended LOS Displacement

results of InSAR data analysis for the city of Quetta Valley and surrounding population centers. The total displacement from a reference elevation, as measured in the interferograms, enables visualization of areas experiencing down-lift or subsidence.

Figure 2 LOS Displacement focuses on potential locations where land subsidence ranging from 0.13m to 0.15m was recorded over the period from 2020 to 2023. These findings provide crucial insights into the extent and magnitude of land subsidence in the study area during the specified timeframe. The identified subsidence zones serve as key indicators for assessing the vulnerability of infrastructure and the potential impacts on local communities.

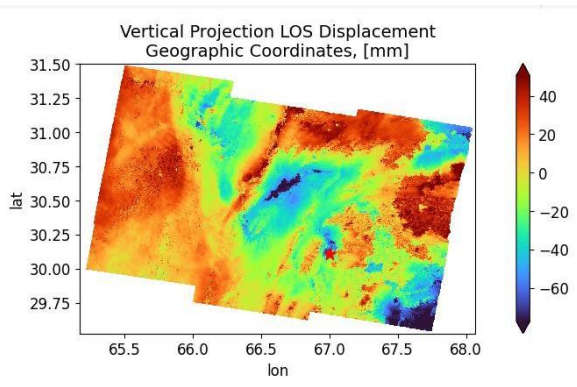


Figure 2 2 Vertical Projection LOS Displacement

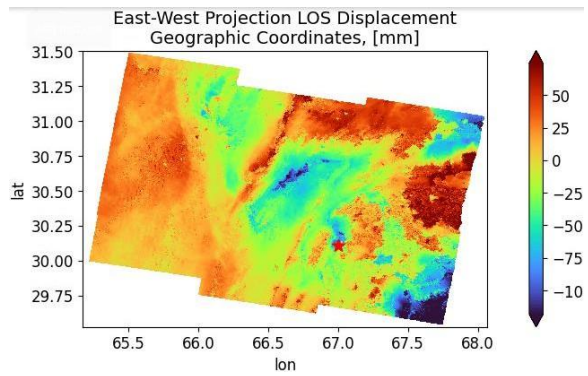


Figure 2 3 East-West LOS Displacement.

Figure 2 LOS Displacement

Similarly, in our second analysis of dataset, we examined that Figure 3 shows the correlation between land subsidence of our specific area of interest of Quetta, observing a positive correlation. This indicates that regions exhibiting higher values tend to experience more pronounced subsidence.

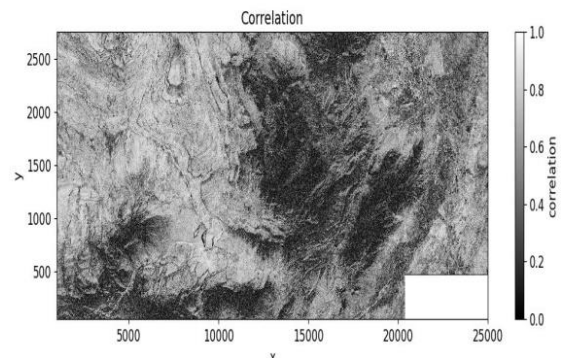


Figure 3 Correlation

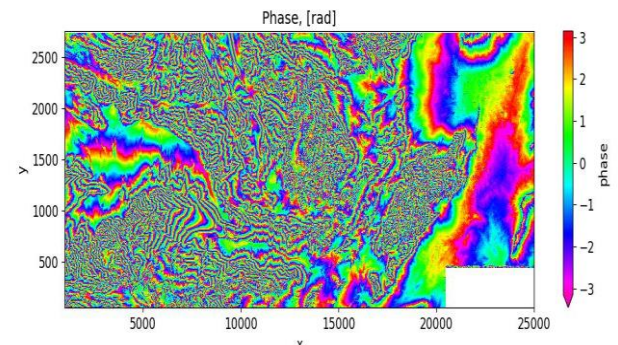


Figure 4 Interferogram

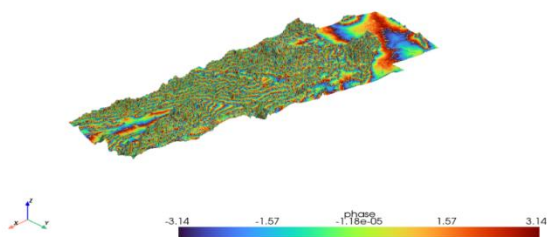


Figure 5 3D Interferogram

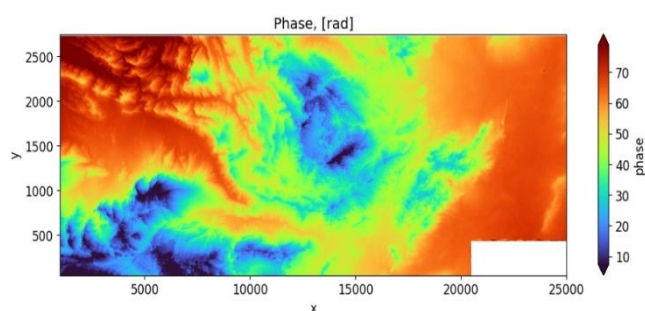


Figure 6 Phase in radians

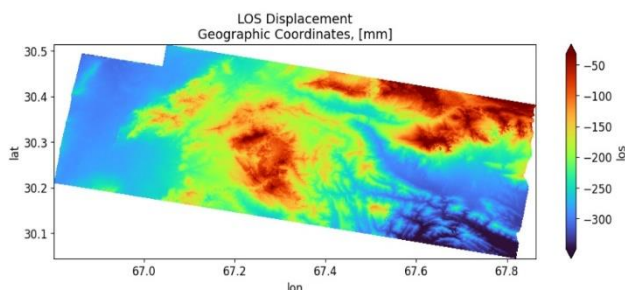


Figure 7 LOS Displacement

Figure 4, Figure 5, Figure 6, Figure 7 above shows the Interferogram generated, 3D interferogram for Area of interest, Phase, and LOS Displacement for the Quetta Valley focusing on potential locations where land subsidence values of 0.06m and -0.394m was recorded over the period of 2023.

CONCLUSION:

In conclusion, PyGMTSAR library using the SBAS method has provided detailed insights into land subsidence dynamics. Through careful analysis of SAR data, we demonstrated the effectiveness of these tools for accurate detection and monitoring subsidence phenomena. This study emphasizes the significance of advanced geospatial analysis methodologies in informing effective subsidence mitigation strategies.

REFERENCES

- [1] A. H. Kazmi and M. Jan, "Geology and Tectonics of Pakistan," Jan. 1997.
- [2] D. L. Galloway, D. R. Jones, and S. E. Ingebritsen, "Land subsidence in the United States," Report 1182, 1999. doi: 10.3133/cir1182.
- [3] T. Davydzienka, P. Tahmasebi, and N. Shokri, "Unveiling the Global Extent of Land Subsidence: The Sinking Crisis," *Geophys. Res. Lett.*, vol. 51, no. 4, p. e2023GL104497, Feb. 2024, doi: 10.1029/2023GL104497.
- [4] K. Pawluszek-Filipiak and A. Borkowski, "Integration of DInSAR and SBAS Techniques to Determine Mining-Related Deformations Using Sentinel-1 Data: The Case Study of Rydułtowy Mine in Poland," *Remote Sens.*, vol. 12, no. 2, p. 242, Jan. 2020, doi: 10.3390/rs12020242.
- [5] L. Bianco, I. V. Djalalova, C. W. King, and J. M. Wilczak, "Diurnal Evolution and Annual Variability of Boundary-Layer Height and Its Correlation to Other Meteorological Variables in California's Central Valley," *Bound. -Layer Meteorol.*, vol. 140, no. 3, pp. 491–511, Sep. 2011, doi: 10.1007/s10546-011-9622-4.
- [6] B. Pradhan, M. H. Abokharima, M. N. Jebur, and M. S. Tehrani, "Land subsidence susceptibility mapping at Kinta Valley (Malaysia) using the evidential belief function model in GIS," *Nat. Hazards*, vol. 73, no. 2, pp. 1019–1042, Sep. 2014, doi: 10.1007/s11069-014-1128-1.
- [7] A. Sowter, Moh. Bin Che Amat, F. Cigna, S. Marsh, A. Athab, and L. Alshammari, "Mexico City land subsidence in 2014–2015 with Sentinel-1 IW TOPS: Results using the Intermittent SBAS (ISBAS) technique," *Int. J. Appl. Earth Obs. Geoinformation*, vol. 52, pp. 230–242, Oct. 2016, doi: 10.1016/j.jag.2016.06.015.
- [8] B. Rimal, H. Baral, N. Stork, K. Paudyal, and S. Rijal, "Growing City and Rapid Land Use Transition: Assessing Multiple Hazards and Risks in the Pokhara Valley, Nepal," *Land*, vol. 4, no. 4, pp. 957–978, Oct. 2015, doi: 10.3390/land4040957.
- [9] H. A. Zebker, P. A. Rosen, R. M. Goldstein, A. Gabriel, and C. L. Werner, "On the derivation of coseismic displacement fields using differential radar interferometry: The Landers earthquake," *J. Geophys. Res. Solid Earth*, vol. 99, no. B10, pp. 19617–19634, Oct. 1994, doi: 10.1029/94JB01179.
- [10] R. Lanari, O. Mora, M. Manunta, J. J. Mallorqui, P. Berardino, and E. Sansosti, "A small-baseline approach for investigating deformations on full-resolution differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 7, pp. 1377–1386, Jul. 2004, doi: 10.1109/TGRS.2004.828196.
- [11] A. Pechnikov, "mobigroup/gmtsar: pygmtsar-v2023.3.11." [object Object], Mar. 12, 2023. doi: 10.5281/ZENODO.7725132.