ISSN Print: 3005-8007
ISSN Online: 3005-8015
Volume 2
Issue 2
July- December 2024

UCP

Journal of Engineering & Information Technology

University of Central Punjab

ISSN:
3005-8015 (Online)
3005-8007 (Print)
Vol. 2, Issue 2
(July - December 2024)

(UCP-JEIT)
UCP Journal of Engineering & Information Technology

Volume 2
Issue 2

Faculty of Information Technology & Computer Sciences
&
Faculty of Engineering

University of Central Punjab, Lahore, Pakistan

Editorial Board

Patron

Editor-in-Chief

Managing Editor

Associate Editors

Advisory Board

International Members

Dr. Haris Javaid
(AMD, Singapore)

Dr. Demostenes Zegarra Rodriguez
(Federal University of Lavras, Brazil)

Dr. Salman Azhar
(Auburn University, USA)

Dr. Ali Kashif Bashir
(Manchester Metropolitan University, UK)

Dr. Muhammad Ramzan
(Saudi Electronic University, KSA)

Dr. Nasir Rajpoot
(University of Warwick, UK)

Dr. Agnes Jocher
(Technical University of Munich)

Dr. Ali Nasir
(King Fahad University of Petroleum and Minerals)

Dr. Moez Ben Houidi
(King Abdullah University of Science and Technology)

National Members

Subscription Charges National: PKR 1000 per issue
International: US$ 200 per issue

Acknowledgment

The Editorial Board of the UCP Journal of Engineering and Information Technology extends heartfelt appreciation to all those who have played crucial roles in bringing Volume 2, Issue 2 to fruition. We sincerely recognize the invaluable contributions of our esteemed researchers/authors, whose dedication to advancing knowledge has enriched this inaugural edition.

We also extend our gratitude to the diligent reviewers whose expertise and insightful feedback have ensured the quality and rigor of the articles published herein. Your commitment to the peer-review process is deeply valued.

Furthermore, we thank all individuals involved in the publication process, including editorial staff, copyeditors, and designers, whose unwavering support and tireless efforts have been indispensable.

Without the collective dedication of these individuals, the publication of Volume 2, Issue 2 of the UCP Journal of Engineering and Information Technology would not have been possible. We anticipate continued collaboration and the exploration of new frontiers in the realm of engineering and information technology.

Warm regards,
Dr. Muhammad Amjad Iqbal
Editor-in-Chief
UCP Journal of Engineering and Information Technology

Disclaimer

The views expressed in these articles are solely those of the respective authors and do not necessarily reflect the views of the Editorial Board or the management and staff of the University of Central Punjab. While every effort has been made to ensure the accuracy of the information provided by the authors, the Editorial Board does not accept any responsibility for any errors or omissions or breach of copyrights, if any.

Every effort has been made to ensure the accuracy and reliability of the information presented in the articles. However, the Editorial Board and the University of Central Punjab make no representations or warranties regarding the completeness, accuracy, or suitability of the content. Readers are encouraged to exercise their judgment and discretion when interpreting and applying the information contained in these articles.

The UCP Journal of Engineering & Technology is committed to upholding the highest standards of academic integrity and ethical publishing practices. Any concerns, questions, or requests for clarification related to the content published in this journal should be directed to the respective authors, who bear full responsibility for their work.

We appreciate your understanding of this disclaimer and hope that you find the content within this journal informative and thought-provoking.

UCP JOURNAL OF ENGINEERING AND TECHNOLOGY

Table of Contents

Date available online: 13-02-2025
Vol. 2, Issue 2 (June - December 2024)

Auto-Classification of FIA-Cybercrime Wing Complaints Using Bidirectional Encoder Representations from Transformers Model

Hafiz Ammar Mazhar[1*], Umair Altaf[2], Syed Muhammad Anwar[1], Muhammad Shoaib Bhutta[4]**

[1]Department of Software Engineering, University of Engineering and Technology UET Taxila,Taxila, Pakistan
[2]University of Central Punjab, Lahore, Pakistan
[3]School of Automobile Engineering, Guilin University of Aerospace Technology, Guilin 541004, China

[*]Corresponding author: Hafiz Ammar Mazhar (e-mail: ammarmb@hotmail.com),
[**]Corresponding author: Umair Altaf (e-mail: umairaltaf8317@gmail.com).

***Abstract*—** The Cybercrime Wing (CCW) of Federal Investigation Agency, which was formerly known as the National Response Center for Cybercrime (NR3C), is governed by rules that were created in 2016 as part of the Prevention of Electronic Crimes Act (PECA) to combat cybercrimes. Criminal activities executed using computers and the internet are referred to as cybercrimes. In order to carry out illegal activities, cyber-criminals make use of any information system as their primary means of communication with the devices that belong to their victims. This research mainly focused on the cybercrime complaints with an automated classification system. To achieve the automatic modelling for classification of different types of cybercrimes, this study used the Bidirectional Encoder Representations from Transformers (BERT). Its obstacles mainly include the possibility of human errors as it manually classifies cybercrime complaints, also that there might be delays during handling in comparison with an automated system. The dataset includes complaints submitted in English during a two years window, and it was encoded, tokenized and cleaned thoroughly. The purpose was to simplify the training process for the model. The study used a lightly fine-tuned, pre-trained (BERT)-base-uncased model. The findings confirm that the model can be used for classifying complaints and exhibits an excellent classification accuracy, precision and F1-scores between different cybercrime offenses indicating its supremacy among advanced Natural language processing (NLP) techniques to strengthen cybersecurity measures.

Index Terms—FIA, BERT, Cybercrime, Classification, NLP

I. INTRODUCTION

The cyber offenses are rapidly increasing, so the Cybercrime Wing (CCW) of Federal Investigation Agency is at Pakistan's forefront for its protection. Imposed under the Prevention of Electronic Crimes Act (PECA) 2016 [1], CCW includes a broad remit to investigate and prosecute cyber crimes. Equipped with the required instruments to counteract digital threats, ranging from financial fraud and online harassment, while ensuring the protection of digital rights, including privacy within the public at large. The formal establishment of CCW signifies a monumental step in Pakistan's drive against cybercrime, presenting firm determination of the state towards curbing this contemporary threat. However, the sheer quantity of complaints and complexity of cybercrimes necessitates creative options to boost the effectiveness & efficiency in CCW operations. Manual classification of cybercrime complaints has inherent problems, such as processing lag and an extended likelihood of human error. Collectively, these issues make CCW ill-equipped to respond quickly and efficiently to cyber threats, necessitating a transition towards automation rather than historic technology implementations. The release of an implementation for automated classification using the BERT model is a major step towards tackling the many problems in complaint classification [2].

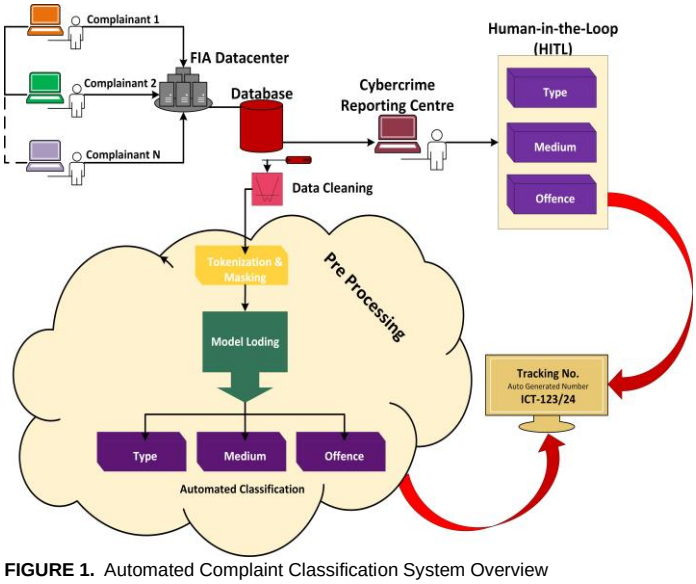


FIGURE 1. Automated Complaint Classification System Overview

In 2019, Kim [3] initially suggested a CNN-based text categorization model, which by utilizing Word2vec

transforms a word into fixed-length vector and then employs multisite convolution to verify word-vector convolution. Pooling and categorization are the final steps. Convolutional neural network's primary benefit is its ability to extract local featured from text efficiently and fast training-speed. The pooling layer, however, will lose a substantial quantity of vital information and ignore the correlation between the whole and local. Wallace and Zhang [4] also suggested a CNN-based text classification approach and conducted multiple comparative experiments with varying hyperparameter settings. In addition, they provided guidance on parameter-tuning and had some experiments with hyperparameter configurations.

As BERT models have pushed the boundaries of text understanding by enhancing context-awareness through multilayer attention mechanisms, the proposed work by Faheem and Al-Khasawneh [5] pushes the boundaries of cyberattack detection in IoBC systems. Their use of deep learning to analyze complex data flows addresses a similar challenge of extracting meaningful patterns from vast and dynamic datasets.

The Deep Pyramid Convolutional Neural Network (DPCNN), proposed by Johnson and Tong [6], enhances text categorization by increasing the network depth to capture long-distance relationships in text. However, their computational demands limit their real-world applications. Similarly, the Graph-Based Convolutional Neural Network (GCN) proposed by Yao et al. [7] excels in classifying small datasets but faces challenges in terms of scalability and adaptability.

Aligns with the challenges of layered architectures discussed in deep learning models like CNNs, RNNs, and BERT for classification tasks. It focuses on how fog computing enhances IoT applications by bringing computational resources closer to the network edge, improving latency and real-time processing, much like CNNs efficiently extract local features in text classification tasks proposed by Burhan et al. (2023) [8]. The study also highlights security concerns in fog-IoT environments, which mirrors challenges in maintaining data integrity and context in NLP models, such as pooling in CNNs that can lose key information.

Mikolov et al. [9] utilized Recurrent Neural Networks (RNNs) for text classification, which are capable of processing inputs of varying lengths but are prone to gradient issues affecting learning efficiency. Models like Recurrent Neural Networks (RNNs) can face "gradient issues". In a vanishing gradient, the adjustments become too small to make progress, leading to slow or stalled learning. In an exploding gradient, they grow too large, causing unstable and erratic training. Both issues make it hard for models to learn effectively, especially in complex tasks like language processing. To overcome RNN's limitations of RNNs with long sequences, Schmidhuber and Hochreiter developed Long Short-Term Memory networks (LSTMs) [10], which, despite their improved performance, require extensive computation owing to their complex structures and numerous parameters.

Chung et al. [11] introduced the Gated Recurrent Unit (GRU) model, which streamlines the LSTM architecture for better training efficiency, though it still struggles with parallel computation and gradient issues. Graves and Schmidhuber [12] advanced LSTM to a bidirectional form (BiLSTM), improving classification, but at the cost of increased complexity. Cao et al. [13] combined BiGRU with contextual understanding to effectively categorize Chinese text, offering simplicity and faster convergence. Li and Dong [14] integrated Convolutional Neural Networks (CNNs) with BiLSTM to enhance text feature extraction for classification.

Faheem and company introduced a blockchain-based framework designed to enhance the security and resilience of distributed renewable energy systems. [15] This framework employs blockchain technology to ensure transparency and immutability, thereby securing the data related to energy events and mitigating unauthorized access through smart contracts and cryptographic algorithms. Although primarily focused on energy management, the principles of data integrity and decentralized control are highly relevant to the field of cybercrime, particularly in the context of automatically classifying cybercrime complaints using BERT. By leveraging the strengths of blockchain in securing complaint data, it is possible to enhance the reliability and transparency of automated classification systems, ensuring that the integrity of the complaint logs is maintained.

The bidirectional encoder representations from transformers (BERT) model [16] was further classified using its two way reading capacity to better understand the text context. Researchers [17] fine-tuned BERT and compared it with other models, such as KNN and SVM, using various sequences, batch sizes, and hyperparameter tuning to achieve sentiment classification. Hyperparameter-tuning is the process of finding the best combination of these settings to improve the model's performance. For example, in this study, different learning rates and batch sizes were tested to see which produced the best results. Fine-tuning these parameters helps optimize the model's ability to learn from data without overfitting or underfitting. Finally, [18] discussed utilizing BERT embedding with a deep neural network for classifying cognitive domains, emphasizing precision, recall, and F-measure for a balanced evaluation against class imbalance.

The original structure of the BERT model utilized only the last layer for classification, neglecting the semantic insights from the lower layers. To remedy this, the BERT-MLF model was created by integrating BERT's full 12-layer architecture via a CNN, but it still lacks dynamic weight assignment to semantic information across layers. The BERT-MLDFA model [19] addresses this by dynamically incorporating parameters from all BERT layers using a multilevel attention mechanism, optimizing the classification for similar content

categories, and enhancing the discrimination of key semantic information. Further developments in sentiment analysis for Weibo text involve enriching word vectors with an external sentiment dictionary and combining BERT, BiLSTM, and attention mechanisms with a CNN for feature extraction [20]. This enhances the classification accuracy.

For hate speech detection [21], researchers have optimized BERT training with fine-tuning and logistic regression, specifically tailored to the concise format of Twitter posts. The CyBERT classifier [22] identifies cybersecurity feature claims within small datasets with high confidence by finetuning BERT and analyzing the impact of randomness on model accuracy. This involved training with various random seeds to achieve reliable accuracy results, enhancing the understanding of how randomness affects model performance. An analytical literature review [23] revealed a gap in the use of advanced deep learning for efficient complaint management. The integration of multi-criteria decision-making with deep learning, particularly BERT, is highlighted for enhancing customer satisfaction and complaint-processing efficiency.

Faheem et al. [24] investigate cyberattack patterns in blockchain-based communication networks for distributed renewable energy systems, utilizing large datasets to identify various cyber threats and vulnerabilities. Their analysis employs advanced data analytics and machine learning techniques to uncover trends in cyberattacks, highlighting the necessity for robust security measures. This study is particularly relevant for the autoclassification of cybercrime complaints using BERT models, as it demonstrates the efficacy of machine learning in threat detection and classification. The insights gained from their research can enhance BERTbased systems' ability to recognize and categorize cyber threats, improving the efficiency and accuracy of cybercrime classification processes. By integrating findings on attack patterns into complaint management, the study supports the development of data-driven approaches for better cybersecurity in the realm of cybercrime.

The BERT4TC-S model [25] was evaluated across several datasets, with learning rate adjustments significantly impacting the performance, suggesting a learning rate of 2e-05 for optimal accuracy and macro F1 scores. Finally, the increasing importance of NLP and text classification across sectors is emphasized [26], particularly the transformative impact of the BERT model in yielding more accurate and context-aware interpretations of vast, unstructured text data on social media.

II. DATA ANALYTICS OF FAULT CLASSIFICATION

The first histogram shown in 2 detailing the offense (sections of law) that is Cyber Terrorism, unauthorized use of identity information presents the number of complaints associated with a unique identifier for offenses. They suggested that this offense is significantly more common or more frequently reported than others within this dataset. This information could be indicative of prevalent crime trends or reporting behaviors within the jurisdiction of the FIA. The type of histogram, we observed the distribution of complaints across various detailed crime types, such as stalking and identity theft. Certain types stand out with higher frequencies, signaling that these specific categories of crime are encountered more often in reports. This pattern is vital for understanding the landscape of crime types and could potentially inform targeted approaches to crime prevention and reporting. Finally, the medium histogram shows a distribution that highlights the medium related to crime occurrence or the reporting channels used, such as social media and websites. One medium, in particular, is represented significantly more than the others, implying that it is the most common avenue through which crimes are committed or reported in the collected data. This could influence how resources are allocated for monitoring and provide insights into the most effective means for crime reporting outreach.

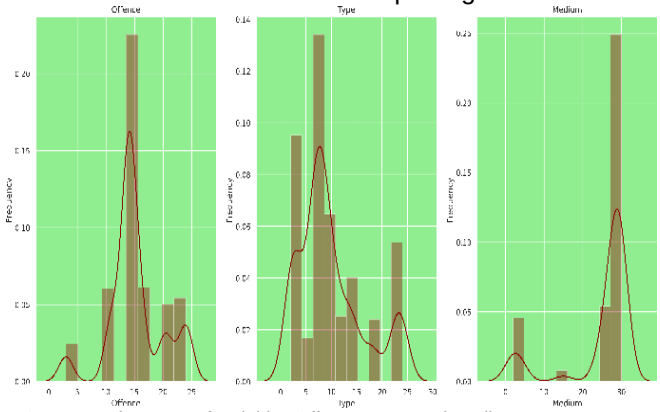


FIGURE 2. Histogram of variables Offense, Type and Medium

The pair plot is a comprehensive visualization that illustrates the relationships between three key variables from the dataset: OFFENSE, Type, and Medium. On the diagonal, we see plots (likely histograms or density plots) representing the distribution of each variable individually, giving insights into the frequency and spread of each category of offenses, crime types, and crime mediums. The off-diagonal elements are scatter plots that map the intersection of two different variables, showing how they correlate with each other. For instance, a plot comparing offense on the x-axis with type on the y-axis would display the extent to which certain types of crimes are associated with specific offenses as shown in fig.3. Contour lines in these plots indicate the density of data points; tightly packed contour lines suggest a higher concentration of data points, which can be indicative of a strong correlation or a prevalent combination of categories. In cases where the data points form distinct clusters, this could suggest sub-groupings within the data that might be significant for classification tasks. This visualization technique allows for simultaneous examination of potential linear relationships, outliers, and groupings across multiple dimensions of the dataset, which is crucial for identifying patterns that the BERT model should learn to recognize and classify effectively.

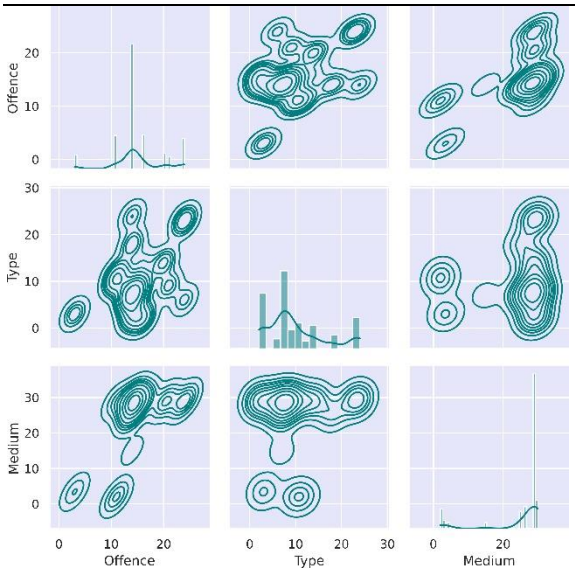


FIGURE 3. Pair Plot of Input Data

III. PROPOSED FRAMEWORK

A. DATASET OVERVIEW

The dataset under consideration comprises cybercrime complaints submitted by the public to the Federal Investigation Agency's Cybercrime Wing. Specifically, this collection focuses on complaints expressed in English by the complainants, thereby ensuring linguistic uniformity that is critical for the analysis. The dataset encompasses a comprehensive range of cybercrime incidents reported over a two year period.

Data gathering was conducted in compliance with stringent ethical standards and with the approval of the FIA Cybercrime Wing. Measures have been introduced to protect privacy and shield personal information from any data analysis or sharing. It is an important learning material for my research work on "Auto-Classification of FIA-Cybercrime Wing Complaints Using Bidirectional Encoder Representations from Transformers (BERT) Model." Focusing on English language complaints provides a uniform background and allows applying Natural Language Processing (NLP) as well as the BERT model accordingly across the dataset.

B. DATA COLLECTION METHODOLOGY

The prerequisite criterion was that they exist in English, acting as submissions by complainants themselves. This construct was formed for uniform language of analysis and data dependability, restricted to first-person accounts. Reinstated officers who, because of rank, were Inspectors or above revisited all the allegations included in a data-referred case. Even though the data was filtered several times because of the above-mentioned process, it played an important role in ensuring authenticity and significance in each complaint. Formal access permission was obtained from CCW headquarters to visit the FIA Datacenter and harvest data. Permission was essential for legally obtaining the filed complaints and ensuring that other ethical requirements were followed in conducting this study.

Data was discarded which contains any of the complainant credentials (Name, Phone Number, Address, and CNIC Number) for privacy concerns and ethical reasons. A total of 701 complaints alleging criminal nature with FIACybercrime Wing make up our dataset. These complaints have been categorized by type of offense, and the percentage distribution is as follows:

TABLE 1. Distribution of Complaints by Offences

Offence Name	Count
Cyber Stalking	80
Electronic Fraud	286
Hate Speech	89
Offences against dignity of a natural person	42
Offences against modesty of a natural person and minor	32
Unauthorized access to information system or data	82
Unauthorized use of identity information	90

Complaints in the dataset are, on average, 489 characters long, with an average of about 90 words per complaint. These results show a low number of complaints filed and some variability in type, which suggests that using an automatic classification system to process them would be essential.

C. DATA PREPROCESSING

Text data extracted from the 'Description' column to remove any discrepancies or mistakes. Some of these inaccuracies include HTML tags within the text, typographical errors like phone numbers, and unnecessary punctuation, which could confuse our model from understanding the input data. Text tokenization was performed to store these clean descriptions in a format BERT can understand. This includes tokenizing the text into components recognized by the model, such as adding special tokens if needed (e.g., BERT expects [CLS] at the start of each record and [SEP] at the end or between sentences) shown in Fig. 4.

This final step in the data preprocessing process involves encoding the Offense, Type, and Medium output labels to a format suitable for training. Dataset is multi-label in the sense that each 'Description' may have more than one output label, we ensured that this concept was captured cleanly via encoding, further strengthening our model training phase.

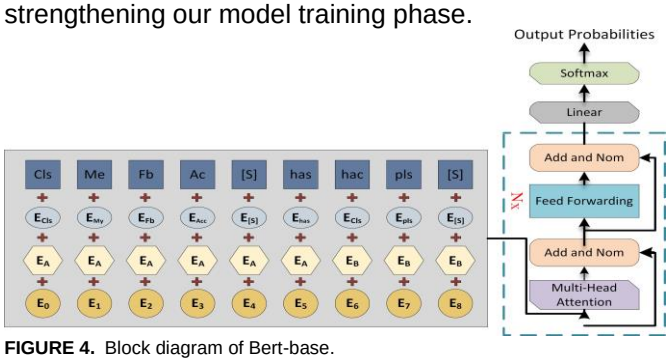


FIGURE 4. Block diagram of Bert-base.

The dataset was divided into an 80% training (model training) portion and the remaining 20% testing (testing the model) portion, reserved for model evaluation on unseen observations.

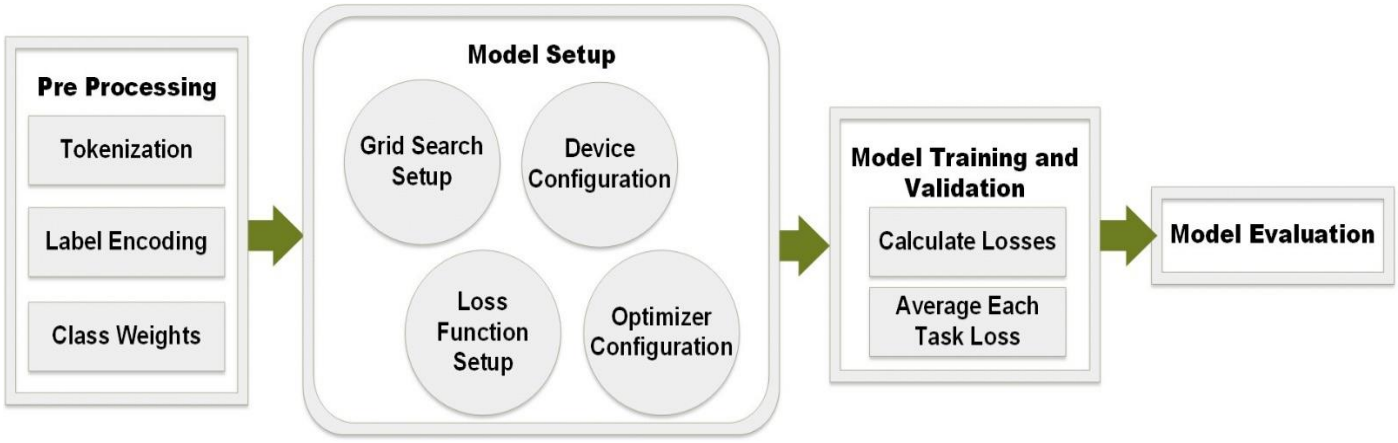


FIGURE 5. Our Cybercrime Complaint Categorization Workflow Model

D. MODEL SETUP AND FINE-TUNING PREPARATION

The pre-trained BERT variant 'BERT-base-uncased' is used, which is known to effectively tackle NLP tasks and is more generalizable over varied text data owing to its case insensitive nature. BERT was adapted in the uncased version released by Google, as it is fully featured and pre-trained on a large part of uncased text, ensuring robust performance across different text classification tasks. To adapt this pretrained model for multi-label classification, aiming to predict 'Offense,' 'Type,' and 'Medium' from the descriptions, AdamW optimizer is utilized because of its advanced handling of weight decay, which effectively minimized overfitting.

E. MODEL FINE-TUNING

The fine-tuning of the pre-trained BERT-base-uncased model for classifying cybercrime complaints into Offence, Type, and Medium categories is a critical process. In various cases, complaints may span multiple categories of cybercrime. However, our model is trained on data where each complaint is assigned to only one initial category. This category is determined based on the severity of the offense, specifically selecting the category with the maximum punishment as per local law. By focusing on the most severe category, we ensure that the model prioritizes the most critical aspects of each complaint, thereby streamlining the classification process and aligning with legal standards. This approach allows the model to effectively handle ambiguous cases by emphasizing the most significant category of the offence.

The model was fine-tuned with a 10-epoch train/validation split of 20%, allowing the learning progress and generalization ability of the trained model to be evaluated over time. We also used the AdamW optimizer for its enhanced sparse gradient support and adaptive learning rate capabilities, which are essential for optimizing a complex training data setup.

To accurately assess the effectiveness of the model and guide its training, we employed the cross-entropy loss function. This choice allowed for the measurement of the disparity between the model's predictions and actual labels across each output category. By computing and monitoring the average of these losses, we ensured a balanced optimization strategy that addressed the nuances of each classification task. This methodical monitoring of training and validation losses, both at the category level and overall, was instrumental in making real-time adjustments to improve the model performance. Through this vigilant oversight, we were able to refine the model to a point where it demonstrated robust and precise performance on the new data.

Data anonymization has been performed to remove names, phone numbers, and addresses. The research was conducted in alignment with ethical guidelines for handling sensitive data, including obtaining necessary approvals from concerned higher authorities. Data access was restricted to authorized personnel only, and all data processing activities were conducted in secure environments to prevent unauthorized access. The entire model training was performed on an offline machine after downloading the required libraries to secure the data.

IV. RESULTS AND ANALYSIS

A. EXPERIMENTAL SETUP AND HYPERPARAMETER GRID

The foundational step in our empirical investigation was the establishment of a robust experimental setup tailored to explore a diverse array of hyperparameter combinations.

This process was driven by the idea of finding the best configuration that maximizes model performance, especially generalization, which can be evidenced by validation loss and accuracy. Here's how the qualitative hyperparameter space was mapped out:

- Learning Rates: [1×10^{-4}, 2×10^{-5}, 1×10^{-5}]
- Batch Sizes: [8, 16]
- Weight Decay Factors: [0.01, 0.05]

Through this grid, a total of 12 unique configurations were examined, providing a comprehensive view of the hyperparameter landscape.

B. METHODOLOGY OF MODEL EVALUATION

A standardized dataset was used to rigorously test each configuration for consistency across trials. To assess the performance of each configuration, measures were taken from the final epoch metrics for both validation loss and accuracy, as these are key indicators of model effectiveness and generalization

capability. This methodology enabled fair benchmarking, ensuring that the results reflected intrinsic hyperparameter effects rather than external variabilities.

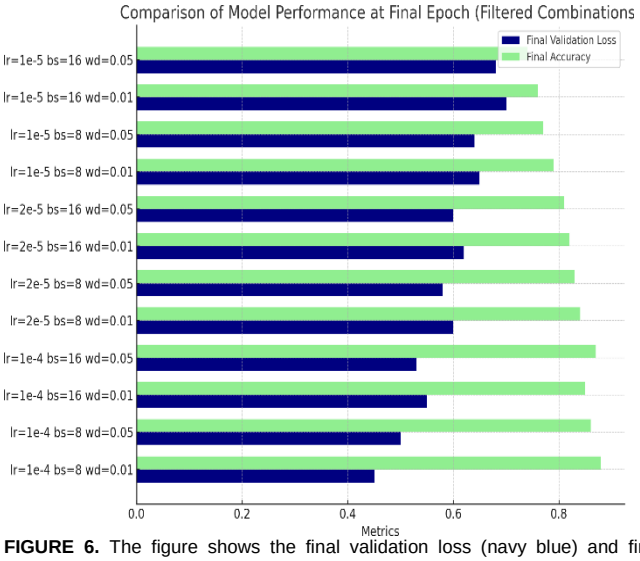


FIGURE 6. The figure shows the final validation loss (navy blue) and final accuracy (lime green) for 11 well-performing sets of learning rate, batch size, and weight decay.

C. ANALYSIS OF OPTIMAL AND SUBOPTIMAL CONFIGURATIONS

Post-evaluation, to demonstrate the discrepancies in performance across various hyperparameter configurations, a detailed visualization was generated. We plotted a horizontal bar chart using the matplotlib library to show the final validation loss and accuracy for 11 prominent configurations, as shown in Fig.6.

The chart in Fig.6 shows the comparison of how changes in learning rate, batch size, and weight decay influence overall model performance through three metrics. The performance metrics, validation loss, and accuracy are shown on the X-axis with different color codes for each. The navy blue color indicates the validation loss, reflecting how much error the model encountered on unseen data, while lime green shows the overall accuracy of the model's predictions.

This visual approach is not only more human-readable, allowing us to view the data, but it also highlights how even small variations in hyperparameter settings can drastically change certain effects. For example, models trained with lower learning rate configurations tend to be more accurate, and the validation loss generally decreases. These charts emphasize the profound interdependencies of hyperparameters and how, collectively, they significantly influence model performance.

The analysis identified the configuration with a learning rate of $1X10^{-4}$, batch size of 8, and a weight decay of 0.01, yielding superior results. These hyperparameters significantly minimized the validation loss while simultaneously maximizing accuracy, indicating an impressive model training scheme. The small learning rate likely facilitated smoother and more stable backpropagation updates, allowing the model to learn a better set of weights.

Conversely, the configuration deemed least effective featured a learning rate of $1X10^{-5}$, batch size of 16, and a weight decay of 0.05. The higher learning rate in this setup seems to have led to less stable convergence during training, resulting in higher validation losses and lower accuracy. This outcome further highlights the critical importance of selecting an appropriate learning rate, which plays a key role in balancing the trade-off between convergence speed and training stability.

D. TECHNICAL ANALYSIS OF MODEL PERFORMANCE

After fine-tuning the BERT-base-uncased model for cybercrime complaints classification, an in-depth analysis was performed to assess its progress. The metrics included accuracy, precision, recall, and F1-score, all chosen for their relevance to the multi-label classification task. Additionally, the model's performance was evaluated over various epochs in terms of training and validation loss, providing a narrative about how well the model's learning function progressed over time.

- **Accuracy:** This metric measures how well your model accurately classifies complaints into their respective categories and serves as a direct gauge of its overall performance.
- **Precision:** These metrics tell us the accuracy of our model for each category. Precision represents how many of the most relevant results were returned,
- **F1-Score:** The F1-score is calculated using precision and recall, providing a combined view of the model's precision and sensitivity. This metric is particularly useful in scenarios where both false positives and false negatives are equally undesirable.

E. LOSS ANALYSIS

The learning performance and general applicability of the model are primarily examined through training versus validation losses. Plots for training and validation losses allow you to see how the model learns over epochs.

- **Training Loss:** Graphical analysis of the training loss provides insights into the model's ability to learn from the training dataset over time. A steady decrease in training loss indicates effective learning, whereas plateaus or increases could signal overfitting or insufficient model complexity.
- **Validatiobility** The validation loss graph is crucial for assessing the generalization capabilities of the model. Ideally, the validation loss should decrease alongside the training loss, converging to a point that indicates the optimal model performance. The divergence between the training and validation loss, particularly when the validation loss increases, suggests overfitting to the training data.

F. TRAINING LOSS BY TASK

Training losses associated with each distinct task:

Offense, Type, and Medium. Unlike a consolidated average training loss, this approach enables an in-depth examination of the model's learning dynamics and performance nuances for each task separately.

The training loss for all three tasks Offense, Type, and Medium across 10 epochs shows how the model's performance evolves during supervised learning. There is a clear downward trend in loss, indicating that the model is learning and improving its predictions over time. This decreasing curve suggests that the model is effectively adapting to each task, further enhancing its ability to handle and tailor its performance to the specific tasks.

The Fig. 7 depicting the training loss for each task (Offence, Type, and Medium) over 10 epochs illustrates the mechanism through which the model learns throughout its training process. There is a clear decrease in loss for every task, indicating that the model learns correctly from the training data and eventually makes better predictions. This downward trend is a good signal that the model can learn and improve its understanding of each task with every epoch.

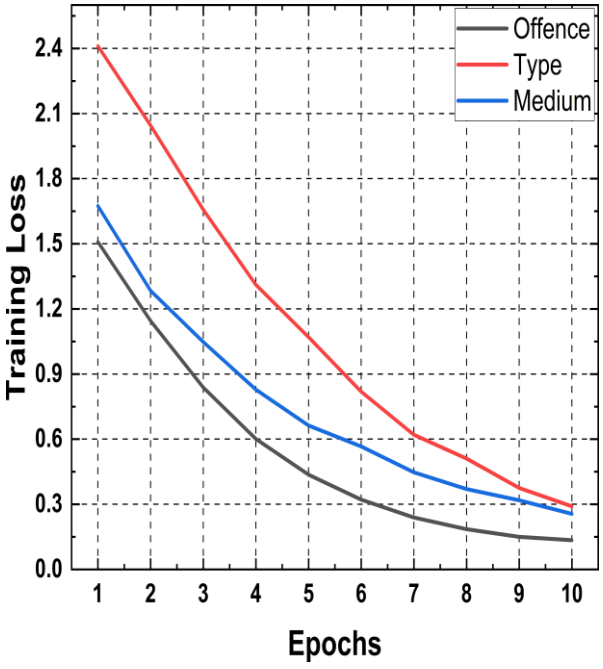


FIGURE 7. Training Loss of each task: Offence, Type, and Medium

G. VALIDATION LOSS BY TASK

The validation losses for the tasks Offence, Type, and Medium across 10 epochs provided insight into the model's generalization capabilities. The descending path of the validation loss for each task indicates that the model not only absorbs the training data but also generalizes effectively on new, unseen data. This trend reflects a model that improves with more data and generalizes well over time.

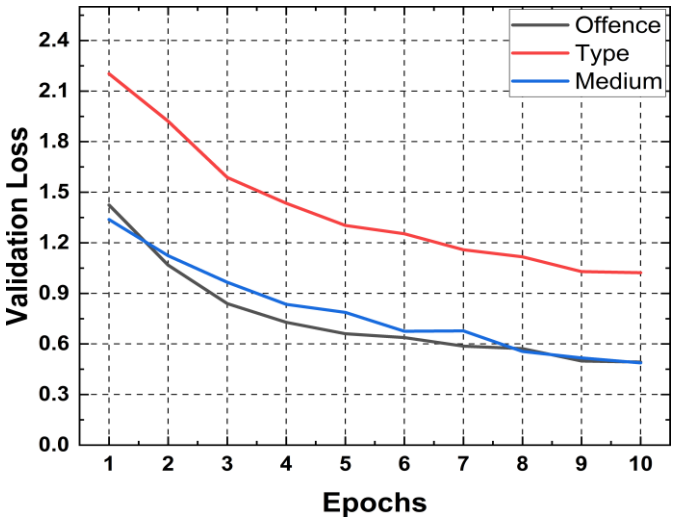


FIGURE 8. Validation Loss for each task: Offence, Type, and Medium

The rate of reduction in validation loss versus training is task-dependent, characterizing the difficulty of each problem when generalizing. For instance, a steeper decrease in validation loss for the Type task implies that the model can better generalize what it learned during training. Conversely, a slower decline in validation loss for the Offense and Mediumtasks might suggest that we did not fully explore the model space or that these tasks are inherently more challenging.

Fig. shown in 8 the validation loss per task over the epochs, showing the model's learning process on out-of-sample data. The consistent decline across tasks confirms the model's increasing performance, while the distinct loss trajectories highlight the unique challenges of generalizing each task.

H. COMPARATIVE VISUALIZATION OF AVERAGE TRAINING AND VALIDATION LOSSES

A comparative visualization of average training and validation losses, showcasing the model's learning trajectory over successive epochs.

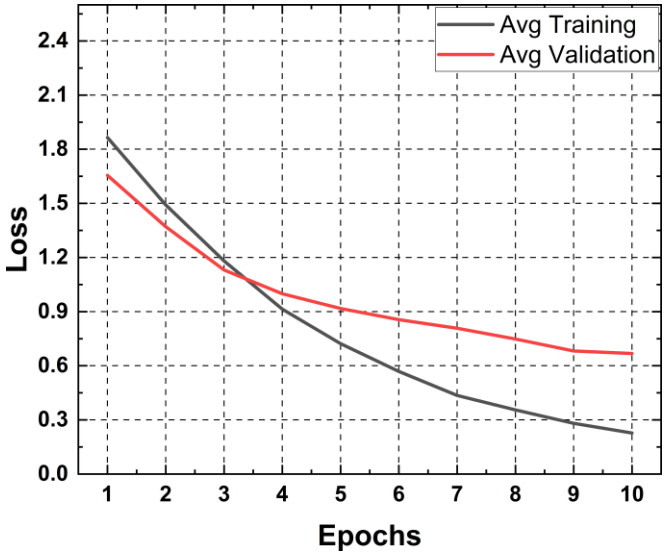


FIGURE 9. Average Loss

Fig. 9 illustrates the progression of average training and validation losses across all tasks over the epochs during the model's training phase. Each point

represents the mean loss averaged over all batches for the training and validation datasets at the end of each epoch. The trend lines describe the model's learning efficiency and its ability to generalize from unseen data. A decreasing difference between the training and validation losses indicates improved model generalization as it is trained.

The Average Training Loss is a metric that provides an aggregate measure of the performance of a model across multiple tasks. It was computed as the arithmetic mean of the individual training losses from the respective tasks. The Average Training Loss is expressed as follows:

$$L_{\text{average, train}} = \frac{1}{3}\left(L_{\text{train, Offence}} + L_{\text{train, Type}} + L_{\text{train, Medium}}\right) \quad (1)$$

where:

- $L_{\text{train, Offense}}$ denotes the training loss for the Offense task,
- $L_{\text{train, Type}}$ denotes the training loss for the Type task, and
- $L_{\text{train, Medium}}$ denotes the training loss for the Medium task.

Similarly, the Average Validation Loss is defined as the mean of the validation losses across the same tasks, which provides an indicator of how well the model generalizes to new data. It is defined as:

$$L_{\text{average, val}} = \frac{1}{3}\left(L_{\text{val, Offence}} + L_{\text{val, Type}} + L_{\text{val, Medium}}\right) \quad (2)$$

• Lval, Offence is the validation loss for the Offence task,

• Lval, Type is the validation loss for the Type task, and

• Lval, Medium is the validation loss for the Medium task.

AVERAGE TRAINING AND VALIDATION LOSS

The average training loss over all batches in an epoch is calculated by summing the loss of each batch and then dividing by the number of batches:

$$\text{Average Training Loss} = \frac{1}{N}\sum_{i=1}^{N} L_i \quad (3)$$

The average validation loss over all batches in the validation set is calculated similarly:

$$\text{Average Validation Loss} = \frac{1}{N_{\text{val}}}\sum_{i=1}^{N_{\text{val}}} L_{\text{val},i} \quad (4)$$

I. ACCURACY

The ratio of correctly predicted observations to the total observations:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \quad (5)$$

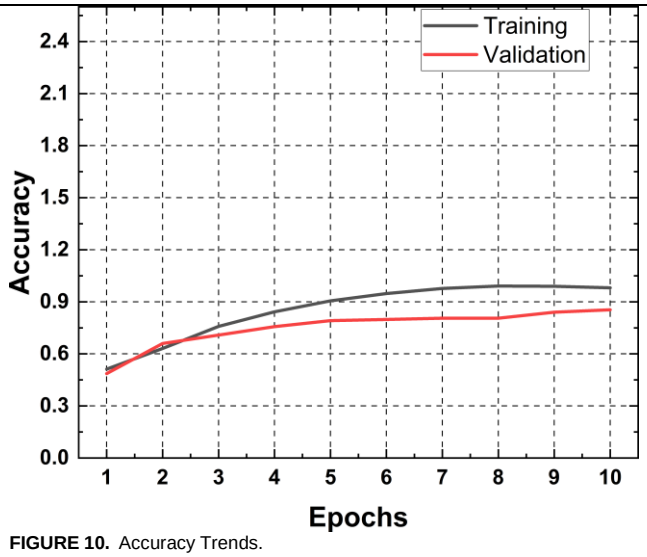


FIGURE 10. Accuracy Trends.

Figure 10 presents a promising outcome of the model's performance over a span of 10 epochs, highlighting its robust learning from the training dataset, as evidenced by the steady upward trend in the training accuracy. This consistent increase is indicative of the model's ability to effectively grasp and memorize the underlying patterns in the training data.

In terms of the validation accuracy, the model demonstrated a reasonable level of generalization from the outset. The early rise and subsequent plateau in the validation accuracy suggest that the model quickly reaches an optimal level of performance on unseen data. This is a positive feature because it indicates a stable and reliable prediction capability after the initial learning phase. The plateau may also imply that the model has achieved a balance between learning and generalization, avoiding the common pitfall of over-fitting, where further training does not yield significant gains in the validation performance.

Overall, the model exhibited strong predictive abilities, with the potential for further fine-tuning to incrementally improve validation accuracy, if necessary. This performance underlines the model's applicability in practical scenarios, where it can be expected to perform with a reliable level of accuracy on new data.

J. PRECISION

For a particular class: Precision (P) is the ratio of correctly predicted positive observations to the total predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

The precision graph for our model over ten epochs reflects a commendable degree of predictive quality, particularly in relation to the model's specificity in the classification tasks. Precision, which is the proportion of true positives against all positive predictions, is a crucial indicator of a model's performance, particularly when the costs of false positives are high.

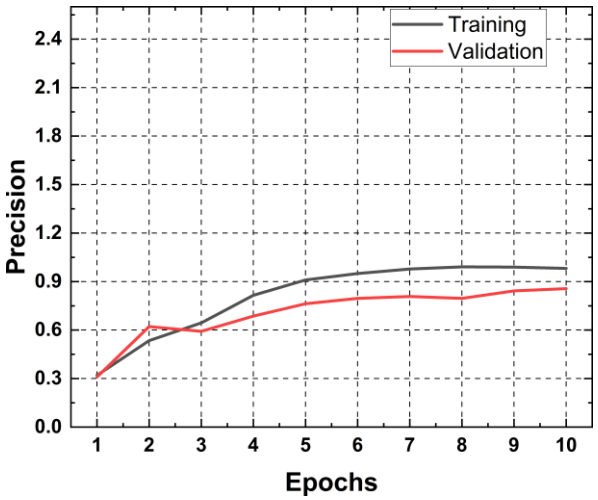


FIGURE 11. Precision Trends.

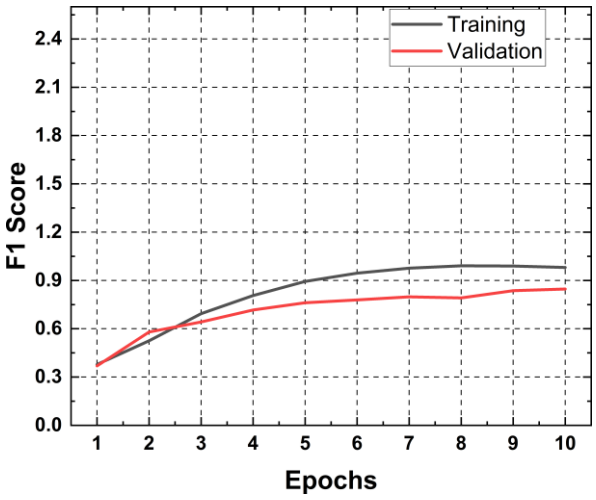


FIGURE 12. F1 Scores.

The training precision shows a gradual and stable improvement, suggesting that the model consistently learns to make more accurate positive predictions as it processes more data. Such an improvement indicates an underlying robustness in the model's ability to discern and predict the correct classes over time.

On the validation side, the precision starts off strongly and maintains a level course, underscoring the model's capability to generalize well from the training data to unseen data. The early convergence to a stable precision rate in validation also implies that the model not only memorizes the training data, but also effectively learns the distinguishing features that generalize across different datasets.

In practical terms, this stable precision suggests that, once trained, the model can be expected to maintain a consistent level of performance, making it a reliable tool for deployment in real-world complaints where precision is valued and necessary for the given task.

K. F1 SCORE

F1 Score is the weighted average of Precision and Recall:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (7)$$

The Fig. 12 monitors the harmonic mean of precision, shows an upward trend in the model's ability to balance these metrics over 10 epochs. An upward arcing training F1 score indicates that the model is learning to categorize positive instances correctly while reducing both false positives and negatives. In the realm of validation, the graph on F1 score starts strong and maintains a steady level, indicating consistency in the model's behavior when applied to unseen data. This balance between precision and recall ensures the model does not become biased toward either metric.

The convergence of the training and validation score graph toward the end of the epochs strongly suggests that the model is not overfitting and will likely perform well in practical scenarios, maintaining high accuracy in predictions. This demonstrates the reliability of this model as a predictor for real-world applications.

V. CONCLUSION

The culmination of this research integrates the meticulous fine-tuning of the BERT-base-uncased model with a strategically chosen hyperparameter set, yielding an optimal balance between precision and adaptability in the classification of multi-faceted cybercrime complaints. The selected hyperparameters, consisting of a learning rate of 1X 10^{-4}, batch size of 8, and weight decay of 0.01, demonstrated the model's heightened proficiency in minimizing losses and enhancing predictive metrics such as accuracy, precision and F1 score across multiple dimensions: Offense, Type, and Medium.

Incorporating the BERT model's capabilities, this study highlights its significant potential in processing extensive volumes of cybercrime complaints, thereby reducing human errors and dependency, and refining the allocation of investigative resources. The application of advanced machine learning and natural language processing techniques showcased herein is not only a testament to their efficacy in cybercrime combat but also a scalable, adaptive approach to the challenges posed by an evolving digital threat landscape. The findings serve not only as a robust basis for furthering automated systems in the enforcement but also as a harmonization of AI with cybercrime law enforcement operations paves the way for future advancements, ushering in a new era of technologically empowered legal frameworks. This research aligns with government initiatives promoting digital transformation and AI adoption in public services, enhancing efficiency, effectiveness, and decision-making in cybercrime law enforcement. By leveraging advanced machine learning and natural language processing techniques, this study supports strategic objectives.

The Agency can benefit from Automatic Complaint

Classification, nothing more unsatisfying than completing skillor labor-intensive repetitive task all day long. Reducing manual tasks will liberate the officers' time for higher-value responsibilities. Delay in Complaints Processing mostly left the complainants aggrieved. By accelerating the process, the Agency will be able to promptly respond back to people, which will ultimately boost public satisfaction and trust in public organizations. Apart from savings from hiring less personnel, the greatest cost reductions from automation will also be realized through the decrease of employee hours.

A. LIMITATIONS OF THE STUDY

One limitation of this study is the reliance on a specific dataset of complaints received via online channels at the cybercrime. While this dataset provides valuable insights, its representativeness for all types of complaints and cases handled by the agency might be limited. Additionally, the focus of this study on automating the initial classification procedure may not account for the nuanced nature of certain complaints that require human judgment and context. Furthermore, the effectiveness of the automated classification method may vary depending on the quality and quantity of the data available for training the model. Therefore, generalizing the findings beyond the specific context of the cybercrime complaints portal might require further validation and testing in diverse settings.

VI. ACKNOWLEDGEMENTS

Data Access To address privacy and confidentiality concerns, each complaint was anonymized by removing personal identifiers such as names, phone numbers, addresses, and other sensitive information. The authors adhered to stringent data protection protocols and obtained formal authorization to access the FIA's data facilities for data collection purposes. All data processing was conducted offline, with restricted access to authorized personnel only, ensuring compliance with privacy standards throughout the study.

Funding Disclosure: This research received no external funding. The study was independently conducted and financed by the authors.

Conflict of Interest The authors declare no conflict of interest.

REFERENCES

[1] AuthorFirstName AuthorLastName. Title of the webpage. https://search.yahoo.com/search?fr=mcafeetype=E210US91213G0p=PECA+2016,, Year of Publication. Accessed: date-of-access.

[2] Author(s) or Organization. Title of the document. Technical report, Publishing Institution or Organization, Publication Year. Accessed: Your Access Date.

[3] Hannah Kim and Young-Seob Jeong. Sentiment classification using convolutional neural networks. Applied Sciences, 9(11):2347, 2019.

[4] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820, 2015.

[5] Mahmoud A. Al-Khasawneh Muhammad Faheem. Multilayer cyberattacks identification and classification using machine learning in internet of blockchain (iobc)-based energy networks. Data in Brief, 2024.

[6] Rie Johnson and Tong Zhang. Deep pyramid convolutional neural networks for text categorization. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 562–570, 2017.

[7] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 7370–7377, 2019.

[8] Ahmad Arsalan Rana Asif Rehman Muhammad Anwar Muhammad Faheem Muhammad Waqar Ashraf Muhammad Burhan, Hina Alam. A comprehensive survey on the cooperation of fog computing paradigmbased iot applications: Layered architecture, real-time security issues, and solutions. IEEE Access, 2023.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26, 2013.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[11] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.

[12] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural networks, 18(5-6):602–610, 2005.

[13] Y Cao, TR Li, Z Jia, and CF Yin. Bgru: a new method of emotion analysisbased on chinese text. Computer Science and Exploration, 13(6):973–981,2019.

[14] Yang Li and Hongbin Dong. Text sentiment analysis based on feature fusion of convolution neural network and bidirectional long short-term memory network. Journal of computer Applications, 38(11):3075, 2018.

[15] Raza B. Bhutta M. S. Madni S. H. H. Faheem, M. A blockchainbased resilient and secure framework for events monitoring and control in distributed renewable energy systems. IET Blockchain, 2024.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[17] Ali Areshey and Hassan Mathkour. Transfer learning for sentiment classification using bidirectional encoder representations from transformers (bert) model. Sensors, 23(11):5232, 2023.

[18] G PRAKASH and J DHAYANITHI. Auto classification of blooms cognitive domain using word embedding deep neural network classifier. Authorea Preprints, 2023.

[19] Xiangdong Li, Jian Shi, Qianru Sun, and Renxian Zuo. Auto-classification of similar categories based on an improved bert-mldfa method——taking e271 and e712. 51 of chinese library classification as an example. 2022.

[20] Hongchan Li, Yu Ma, Zishuai Ma, and Haodong Zhu. Weibo text sentiment analysis based on bert and deep learning. Applied Sciences, 11(22):10774, 2021.

[21] Shailja Gupta, Sachin Lakra, and Manpreet Kaur. Study on bert modelfor hate speech detection. In 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), pages 1–8. IEEE, 2020.

[22] Kimia Ameri, Michael Hempel, Hamid Sharif, Juan Lopez Jr, and Kalyan Perumalla. Cybert: Cybersecurity claim classification by fine-tuning the bert language model. Journal of Cybersecurity and Privacy, 1(4):615–637,2021.

[23] Carla Vairetti, Ignacio Aránguiz, Sebastián Maldonado, Juan Pablo Karmy, and Alonso Leal. Analytics-driven complaint prioritisation via deep learning and multicriteria decision-making. European Journal of Operational Research, 312(3):1108–1118, 2024.

[24] Al-Khasawneh M. A. Khan A. A. Madni S. H. H. Faheem, M.

Cyberattack patterns in blockchain-based communication networks for distributed renewable energy systems: a study on big datasets. Data in Brief, 2024.
[25] Shanshan Yu, Jindian Su, and Da Luo. Improving bert-based text classification with auxiliary sentence and domain knowledge. IEEE Access, 7:176600–176612, 2019.
[26] Rukhma Qasim, Waqas Haider Bangyal, Mohammed A Alqarni, Abdulwahab Ali Almazroi, et al. A fine-tuned bert-based transfer learning approach for text classification. Journal of healthcare engineering, 2022, 2022.

Date available online: 13-02-2025
Vol. 2, Issue 2 (June - December 2024)

DDoS Attack Detection System Using Machine Learning Techniques

Muhammad Zunnurain Hussain[1*], Muhammad Zulkifl Hasan[2], Khawaja Qasim Maqbool[1], Aisha Nazir[1], Hafiz Muhammad Furqan Farid[1]

[1]Bahria University Lahore Campus
[2]Universiti Putra Malaysia

[*]Corresponding author: Muhammad Zunnurain Hussain (e-mail: zunnurain.bulc@bahria.edu.pk)

***Abstract*—** Distributed Denial of Service (DDoS) attacks are the major issues that introduce disruption of accessibility and reliability of the network services. The purpose of this paper is to demonstrate an overall recognized machine learning based system that can efficiently identify and classify DDoS attacks using a rich dataset allowing us to work with various installation network traffic attributes, we have developed an automated classification pipeline the Random Forest Classifier which is known for its high performance in handling large datasets and heterogeneous data. These learnt models were then combined with Decision Tree, Gradient Boosting, and Logistic Regression models to provide a better way analyze the product. An important step involved in this framework is the data preprocessing pipeline that involves one-hot encoding of categorical features to numerical features and scaling for the numerical features, leading to model input optimization. Efficiency of our models is assessed through metrics just as accuracy, precision, recall, and F1-score and it is further validated using cross-validation techniques. The top models are being evaluated by powerful tools, which include feature importance visualization, confusion matrices, precision-recall curves, and calibration curves, for a deeper understanding of their predictive ability as well as their decision-making processes, within those models. A feedback loop mechanism for the iterative betterment and adaptation of the model is accounted which learns from new patterns actively. This approach demonstrates good evaluation and robust in identifying DDoS attacks that are threat of cybersecurity defenses using machine learning.

Index Terms—Decision Tree, Gradient Boosting Classifier, Logistic Regression, Feedback Loop system, Network traffic, DDoS Attack, Network Security, Cross Validation.

I. INTRODUCTION

In the cyber security field, the Distributed Denial (DDoS) of service attacks is the main obstacle that is preventing many of the system's productivity by jamming it with packets of traffic from different sources. Despite the evolution of the DDoS technique, it remains a top concern of cyberattack sector and every subtle shift can present significant outcome. Besides a cross-border level of investigation, the team of investigators additionally may encounter the leading problem as the practice of flooding the objective from several places becomes usual. Even though not any security system usually proves to be effective all the time when fighting this two-edged sword given its commonness and complexity, the ability to adapt to the newest technological developments can help the situation. Though traditional deterrence systems are not sufficient for the interception of such sorts of malicious attacks since they can generally provide only a shock and awe which leads to the inability of dynamicity and adaptivity while encountering enveloping threats. The ML algorithm working hands on with conjunctive rules implementation can be regarded as a machine minded learning from the data and trace the most foolish patterns that are always attacking the cyber universe.

ML is one of the most effective techniques of the past that had been very successful when it comes for instance, to the decision making since it does not require any kind of training and instructions and so decision making happens on its own only. In contrast to the learning algorithms which leave the system free to detect the variations to the standard while working in the network traffic pattern by using any means which go beyond the normal range. The system will be developed by combining two, different types of approaches namely - machine learning and deep learning. This provides a DDoS detection and classification function to detect multiple types of DDoS attacks. In future, such systems could be also applied to some network and then be assured of their completion as well as protection against attacks. The foundation of our super-multi-model machine learning pipeline is the team itself. This support allows all algorithms to arise, Random Forest Classifier, Decision Tree, Gradient Boosting, and Logistic Regression models. In addition, the models chosen should be for withstanding computer applications with different types of data, also the models should be able to adapt to real-time traffic conditions in networks.

This work first involves the development and consequent deployment of a DDoS attack defense system, based on machine learning, to be presented. The model will include different steps, which are data acquiring and pre-

processing, models training, evaluation, and optimization. The theme of work is data preprocessing providing the data quality and data model fitting, and we offer the reasons of listed models and methods for model evaluation and selection.

Furthermore, the architecture is expected to continue to be effective with the implementation of the feedback mechanism, which is a feature that comes with the employment of systems that constantly change as they adapt to new threats keeping having a high detection accuracy over time. Such an ability to adjust is of particular importance for the system in which there is the endless process of digital transformations going on with considerable difficulty connected to effectiveness of the measures updated and their being relevant.

This paper not only suggests the development of system against the DDoS attacks, but also enhance the deployment of effective measurements based on the application of machine learning techniques which are required for the progress of the academic purposes. Our approach to the initial cleaning up, feature engineering and then the incorporation of machine learning algorithms will target optimizing a model capable of identifying normal and malicious traffic competently. Hence, such kind of measures will help in development of new approaches and organizations of more advanced attacks detecting and implementing of unified legitimate monitoring systems.

II. Related Works

The area of Distributed Denial of Service attack recognition has been spared by constantly active research communities trying to propose new ML and DL methods able to strengthen the cybersecurity utilities. Herein, there is the discussion of milestone achievements among the researchers, through various systematic techniques, data sets used and the goal of higher accuracy in detection.

[1] This Research aims to classify DDoS attack packets using the botnet dataset obtained from IoT using PCA as well as Decision Tree, Random Forest, and SVM algorithms. Species of PCA are applied to reduce dimension of dataset, that would not only increase computing power but also preserve the variability, the essential aspect of classification accuracy. It turned out that the Decision Tree and Random Forest methods have more suited for the case of handling unbalanced data then Support Vector Machine. Therefore, they have shown better performance in terms of accuracy and speed. [2] And this study used a DNN model, which was put into a SDN environment, to enhance the accuracy in detection of DDoS attacks. As for the protocol, an algorithm with a new and specially designed flow collector module is applied which, in turn, aids the extraction of features from the network flows. The DNN exploits these attributes and demonstrates its power to address difficulties faced by conventional machine learning models as it accurately recognizes complicated patterns and attacks in the network stream.[3] The approach using LSTM Recurrent Neural Networks is aimed at creating models of network traffic patterns for detecting abnormalities This model goes past pattern recognition and relies on self-learning for future threats. Such model is dynamic as it can recognize and adjust to new attack vectors that are not known today. The AUC of 0.84 means that the trained LSTM RNN can differentiate between normal and abnormal grid systems effectively, which implies that LSTM RNNs have the capability to learn and adapt continuously and are suitable for security systems. [4] The question as to what machine learning algorithms are the most efficient to detect DDoS attacks coming from consumer IoT devices is being focused on in this work. Through this combination of some of Algorithms like K-Nearest Neighbors, Support Vector Machine, Decision Trees, Random Forests, and Neural Networks, the study shows Neural Networks which are fast and precise give the best detection capability. It presents that the observance of special features by specific devices in the IoT might be essential in timely detection of DDoS attacks. [5] Using Random Forest and a Splunk software tool, this report focuses on the identification of "Ping of Death" DDoS attacks. Collaborating the machine learning with Splunk's capability in real time data monitoring, this study does a good job of achieving incredibly high detection accuracy which is 99.8%. By this way, the biggest advantages of hybrid usage of advanced analytical tools in combination with machine learning techniques is proved, which aims at improvement of detection and monitoring of network- based attacks. [6] Research-based learning about LSTM, SVM, and logistic regressions is used in the paper to detect DDoS attacks and models are assessed on their performance using conventional methods. LSTM model has put up a quite strong show of it. The said model not only has good accuracy but also low false positive rate. This proves the excellence of LSTM in the sequence problems prediction, which makes it very convenient for application in dynamic and constantly evolving areas of cybersecurity. A low-weight detector technique is proposed using the Random Forest algorithm and decision tree algorithms-boundary split as well as decision stumps that classify the systems as severe likely. Testing on CICIDS2017 data illustrates that the PDT, especially, allows models to attain both very high accuracy and computational efficiency, which makes the PDT a nice model when the available computational resources, especially, are limited. [8] The testing makes usage of Artificial Neural Networks (ANNs) to enhance the classifier using Mutual Information, which assists in better selection of features; because of this, detection of DDoS attacks is made more efficient and accurate. The ANN's model capability, where the ER of 89.62 percent is achieved, well prove how much feature selection could improve the neural network performance in cybersecurity applications, particularly in differentiating between the ordinary and malicious activities. [9] Deploy CNNs on this research and as a result you will receive 99% accuracy of DDoS traffic identification and files classification. Model

CNN, that can process the data both spatially and temporally, is a perfect tool for dealing with complicate network communication issues in Mobile Cloud Computing (MCC) environments. This result confirms that the model acts the best rather than other typical classifiers and this might be a positive sign for the advanced network security setups. [10] This model was implemented with different types of machine learning algorithms, and it has been demonstrated to produce an intrusion detection system that is effective, using the CICIDS-2017 dataset. The study shows that hereby the Random Forest algorithm exhibits superiority with regards to detecting DDoS attacks, by being capable to process large datasets and since it is well-functioning in handling the balance between bias and variance so that it has a high accuracy and high recall rate. [11] We implement a unique multi-classifier algorithm that unites a complex pyramid of deep neural networks involving CNN, LSTM, and GRU to get hold of numerous types of DDoS attacks including singular and mixed type ones. This ensemble approach leverages the strengths of each model type: For example, applying CNNs to spatial information, LSTM to sequence of temporal recurrence, and GRU with fewer parameters compared to LSTM, for modeling of sequences. Employing an ensemble model merges different models of traffic patterns improvement in detection rates and lessening false positives as it captures more intricate designs and abnormalities in network traffic. [12] This research is about applying a CNN modified particularly for use in IoT environments that are meant to prolong the process of detecting and stopping DDoS attacks. In the deep neural network, the traffic data of the network is processed to point out the malicious activities by learning the complicated structure that can be easily seen from the traditional machine learning methods. The deep model provides the network with the ability to detect tiny irregularities from big amount of data that clearly is more precise in the case of IoT networks, where the behavior of devices may be very different from each other. [13] The model designed to solve the problem of unbalanced network data is presented and can lead to poor machine learning performance. It is guided by a plethora of algorithms, including Random Forest and Convolutional Neural Networks (CNN), to achieve data balance and then training. Iterative and repeated partitioning of data can be helpful in spotting the attack types, mostly missed in skewed labeled sources. The research let us know, that application of model training on balanced dataset has been proved effective and investigate the possible roles of CNNs in extracting the complex pattern to discriminate between normal and anomalous data. [14] One of the approaches is by using multiple linear regression to analyze data attributes characterizing network traffic and setting up typical traffic behavior model of data prediction and comparing it with the observed data, and then, identifying abnormal patterns. The regression analysis is particularly good for doing the job since it can be very fast in the calculation and interpretation of the results, which is a great advantage for environments where the fastest reaction is necessary. Application of PCA-based feature selection allows models to perform more efficiently by removing a number of variables from model without losing necessary information. [15] The authors provide a two-phase detection approach based on the use of linear regression to distinguish routine inbound traffic peaks against a DDoS attack. In the first phase, namely the training phase, the model is developed where historical data is used for understanding the typical traffic patterns typified as the compliance with traffic laws. The prediction phase in this model is designed to match actual traffic behavior and prompt the identification of statistical deviations that exceed the standard limit as possible DDoS attempts. Thus, this model greatly reduces false positives and improves the reliability of such networks in the real world [16] Anomaly is carrying on the research on extracting a particular set of network traffic features that are more useful than others for web-based attacks detection algorithms such as Support Vector Machines (SVC), Random Forest (RF), and Logistic Regression (LR). Innovative design which is capable to highlight the patterns that demonstrate a malicious behavior of web traffic creates a possibility of implementing real-time detection with a higher accuracy, and thus, overcoming the typical limitations of datasets in common use. This study will review the classifiers effectiveness in the detection of DDoS attacks and will pay special attention to feature engineering to improve model efficiency. Targeting only the most representative characteristics of attack traffic allows the models, and especially Random Forest, to efficiently attain a high rate of success. Such an approach centers around the fact that the quality of detection systems is enhanced by the detailed options of features. MC- CNN model, namely DAD-MCNN, employs multiple channels for processing network data, catches DDoS attacks more precisely. Such an approach facilitates the processing of different types of network data in parallel, thus providing the model with improve in accuracy and speed of attack detection. Incremental training is involved to constantly re-train the model with new data, thus allowing it to become more familiar to new attack vectors. [19] It merges signature-based and anomaly-based techniques with LSTM deep learning models to build a customized IDS for IoT ecosystems. This hybrid approach enables the system to detect known attacks using signatures as well as to identify the novel or unknown attacks by means of the behavioral anomalies that LSTM type models detect. [20] It describes the TaxoDaCML framework which uses both Decision Tree and Random Forest models to create a taxonomy-based system of DDoS attacks classification. Such an organized approach enables quick and accurate detection of all known DDoS attack types, improving the system's responsiveness towards attacks based on their specific shapes, patterns, and mechanisms. [21] Designs a bio-inspired model which employs the bat algorithm to effectively detect the Application Layer DDoS attacks within the shortest possible time. The originality of this method is similar to the echolocation behavior of bats to detect changes in

network traffic, which allows quick DDoS attacks mitigations. The real-time detection effect of the model makes it as a promising alternative for the common detection methods This research work also includes developing a DDoS detection system for OpenStack-based private clouds to evaluate the suitability of traditional algorithms such as Decision Trees and more advanced methods such as Deep Neural Networks. The research shows that DNN is superior and adaptable in dealing with different variations of attacks outgoing in the dynamic cloud in terms of detection capabilities. [23] Exposes the AIMM framework, which amalgamates neural networks to a large garden KNN as an effective mechanism of identifying DDoS attacks at their early stages. This method blends the evidencing feature of AI with the convenience of k-NN to produce a powerful detection system. Its high effectiveness in strange situations testifies to the framework's ability to identify even the most intricate of attacks accurately. [24] Suggests the E-HAD architecture, a (creative and distributed) scheme, which is based on Hadoop, correctly manages big volumes of data with the target of early detection of high-speed DDoS attacks. The application of the Shannon metric in network traffic monitoring not only steers the system to have more precision but, in the process, increases the accuracy level fitting the system for use in big networks. [25] Efforts to reveal a shield mechanism against DDoS attacks in SDN and deep learning using fog computing, especially in LSTM modeling are underway now. What is more, the strategy creates obstacle for sending and establishing legitimate traffic that is later on forwarded in the optimal way while blocking anything malicious, thus, providing a good security solution for fog networks. [26] In this work, two models combining LSTM with Random Forest (RF) as well as entropy measure and attribute threshold are applied to not only locate the possible Malicious Attack Behavior, but also to detect DDoS which further improves the detection accuracy. The application of this methodology is built on a foundation of long short-term memory (LSTM) network to process time series information and an ensemble of Random Forest for the robust decision-making. Having employed such a hybrid method, detection accuracy of wrong positives, if any, has been noticeably decreased which is demonstrative of a successful direction towards the development of fully functional network security solutions. This research presents an SDN environment framework that connects multiple kind classifiers KNN, Decision Trees, SVS, Logistic Regression, and XGBoost. Within this architecture, classifiers are integrated to create the final hybrid predicting model. This approach is all about handling the creation of an experimental SDN to let these algorithms conduct processing of traffic data from the network. It is this setting that both features are enabled as well, the real- time attack detection is precisely made, and malicious and legitimate traffic are distinguished effectively. The use of XGBoost what is actual for high performance, to be specific, to obtain first-class precision and recall rates, is an example of its benefit. [28] Random Forest and XGBoost classifiers are initially evaluated in the context of this research, followed by the introduction of a new XGBoost classifier modification targeting improving detection performance in DDoS instances. Through using CICDoS2019 dataset, study purposely test and compare the performance of these models in a well-controlled laboratory environment, which brought about a huge increase in effectiveness such as precision, accuracy, and recall. The modified XGBoost classifier signifies a more customized approach having consideration for the peculiarities of DDoS attack traffic and offers a more exact and fitful detection mechanism. [29] The paper analyzes the capacity of LSTM, SVM, and Logistic Regression models, focusing on their ability to process and foretell malicious activities. The core strength of the LSTM architecture is its capability to accurately recall patterns across time even where the attack signatures can change. The SVM (Support Vector Machine) and Logistic Regression, possess excellent classification capabilities, and are therefore suitable for instances where instant decisions are necessary. The study not only contrast these models, but also investigates their association and how that might affect the overall reliability of detection.[30] Powered by a trimodal combination of Random Forest, Gaussian Naive Bayes, XGBoost, and K-Nearest Neighbors, the study overshoots toward the network traffic classification and helps protect against DDoS attacks. Using the NF-UQ-NIDS-v2 dataset, that covers broadly multiple network traffic scenarios, researching strength of the feature of the algorithms at handling different zones of the data is the point of highlight. Among all models, Random Forest got the highest accuracy, which explicitly verifies its ability to manage large data processing and find key features that are crucial for detecting DDoS activities. [31] This study will mainly put the supervised learning technique into practice with three of them namely Random Forest, Logistic Regression, and K Neighbors Classifier on NSL-KDD dataset with purpose to detect DDoS attacks. Through preprocessing the data and exploiting these classifiers, the analysis demonstrates its outcomes through different performance metrics like accuracy, precision, recall, and F1-score. The analysis clarifies the advantages and disadvantages of each model together with the comparative view that informs better decision-making regarding the use of the most effective strategies in practical applications to the network security infrastructures. The cyber security field is majorly benefited by this study by uplifting the supervised learning techniques as well as their practical application in DDoS threat detection.

Prior work is taken as a foundation to the current research, with machine learning-based DDoS detection examined in several works, including integrating multiple classification techniques and feature importance analysis. For example, some of the easiest models that have been applied previously are Support Vector Machines (SVM) and Decision Tree, the issue with such models is the scalability and their ability to handle new

dynamic attacks. As such, this study adds to the literature by applying ensemble learning techniques including Random Forest and Gradient Boosting to enhancing the received detection accuracy. Nevertheless, it is suggested that the proposed approach may need more thorough testing in managing more diverse and complex cyber threat phenomenons than those emerging from recent deep learning frameworks. For that, the deeper comparative discussion is provided here in order to emphasize what particular strengths and weaknesses are implied by this approach in contrast to the current procedures.

Contribution

This work also proposes an adaptive machine learning over a variety of algorithms' based DDoS detection framework with feedback mechanism. As opposed to previous studies that are concerned with concerted categorical models only, our model pays much attention to performance assessment on an ongoing basis and dynamic optimization in real life. Further, involving an extensive variable selection procedure to improve the model interpretability and the computational cost is minimized. The current work enriches the proposed methodology for cybersecurity studies in terms of reproducibility and scalability with account for practical implementation.

Methodology

This paper deals with detecting DDoS attacks using machine learning methods. The initial part of our system comprises the data collection, cleaning and analysis, feature extraction, model selection and evaluation, followed by a structured algorithm to implement our detection models.

A. DATASET

This paper utilizes dataset from Kaggle. The dataset utilized in this work is a large repository of more than 100,000 network traffic records tailored to empower the detection and classification of the Distributed Denial of Service (DDoS) assaults, which are designed for user operating system. Each line of the dataset refers a unique network flow and includes 23 attributes with an extensive number of components warranting monitoring of the network traffic during routine activity and exceptional conditions. Main attributes involve **dt** (simply the time when the traffic was monitored); **src** and **dst** (popular for indicating IP addresses); **pkt-count** and **bytecount,** detailing the packets and bytes respectively; **dur** (duration in seconds); and **tot-dur** (a cumulative count when several time related attributes are considered). Further, the dataset contains columns specifying the communication protocol used (e.g., TCP, UDP) and flows recording the count of flows observed during the capture interval and **tx_bytes** and **rx_bytes**, indicating the bytes transmitted and received as respective fields of the dataset.

B. DATA PREPROCESSING

At the pre-processing stage in our project, we aimed to improve data quality in the format as well as offering high efficiency and accuracy in detection of DDoS attack. First, we began by imputing the missing values as numerical values for the columns **'rx kbps'** and **'tot kbps'** to replace the blank ones. The imputer that worked for this problem well with that strategy was **SimpleImputer** and it being used to handle the null case with median which preserved the distribution tendency of the dataset. So, median value allowed to mitigate the issue highly relevant for the network dataset due to high frequency of certain outliers. Moreover, the features used in the data represented as string variables such **as 'Src', 'Dst'** and **'Protocol'** were converted into one hot vector using **OneHotEncoder** for the machine learning algorithms to work them as binary values. For improving the data quality, we apply **RobustScaler**, which is useful to handle those columns that are with the numerical data type since it helps to avoid the using of extremes. Similarly to this, the other empty values are simply taken as the median to make the information more reliable. Therefore, we used an 80-20 ratio for training and for validation of the model to ensure that even during the training stage we stay unbiased while during test we provide the model with the opportunity to stay unbiased too. As a result, the signature, which is used as an advantageous processing, implies that no data that can provide input is left, and the dataset completely catches all the discrimination.

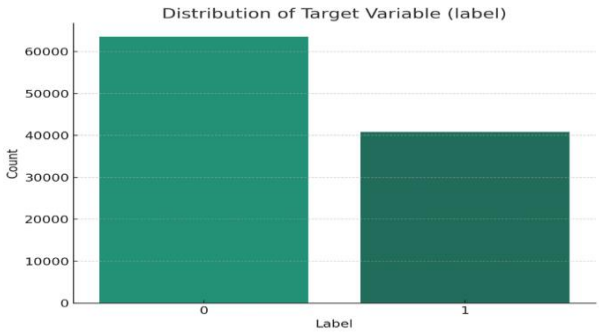


FIGURE 1. The chart showing the target variable distribution indicates that the dataset is somewhat unbalanced between these two types (0 stands for the normal traffic, and 1 for the DDoS attack). The resulting stimulation implies one of the possible learning methods that might lead to detection of both normal and malicious traffic.

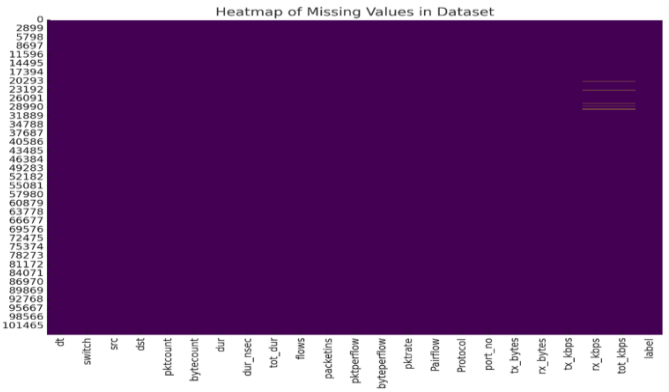


FIGURE 2. The Missing Value Heatmap indicates a minimum level of missing values represented by yellow patterns across features. Consequently, the process of median imputation is applied to ensure a pertinent and thorough dataset for subsequent data analysis.

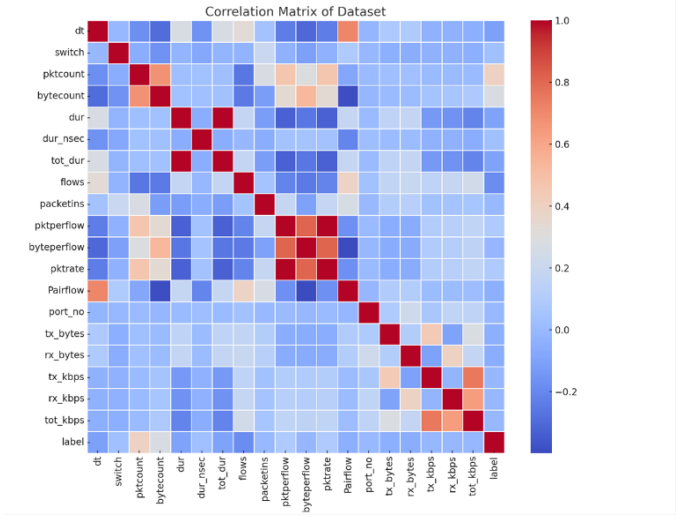


FIGURE 3. The correlation heatmap from blue to red shows the correlation range beginning from low to high. The matrix reveals a potential multicollinearity between features like 'bytecount' and 'pktcount', which further correlate with a label that indicates a DDoS attack.

C. FEATURE ANALYSIS

An integral part of our methodology involved analyzing the dataset to identify and select features critical for distinguishing between normal traffic and malicious DDoS attacks. This process was crucial in refining our models, inputs for optimal performance.

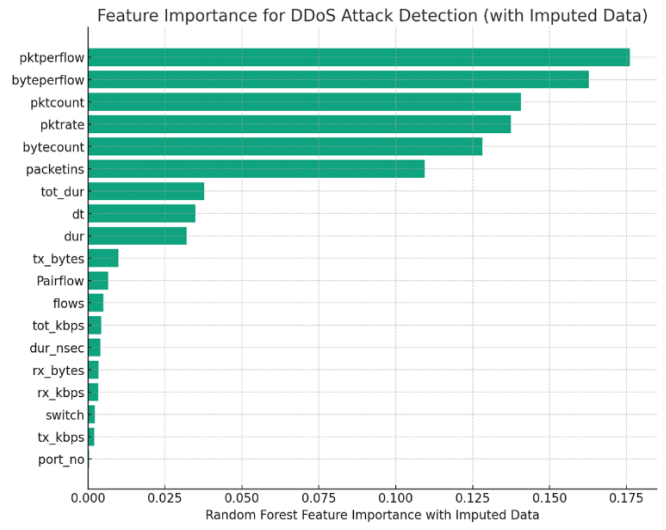


FIGURE 4. The bar chart shows the top feature importances, as identified by the Random Forest Classifier of DDoS Attack Detection after data imputation of missing data. The length of each bar shows how the features 'pktperflow', 'byteperflow' and 'pktcount' matter in the predictive model which plays a big role.

D. MODEL SELECTION

In this paper, we used a few machine learning techniques and those were selected from others for being the most appropriate for classifying the data.

Performance: Accuracy of the model in appropriately categorizing network traffic.

Complexity: Models which can find some compromise between the predictive power and the computational efficiency.

Interpretability: The clarity that the model decisions can be explained.

Robustness: Models of general type that deal with problems such as imbalance which is another network traffic characteristic.

E. PROPOSED MODEL

We have chosen four specified algorithms which are Decision Tree, Logistic Regression, Random Forest Classifier, and Gradient Boosting Classifier, because they are all effective classification algorithms. For evaluating, we utilize the Accuracy, Precision, Recall, F1 Score and AUC Metrics to comprehensively measure individual model levels of performance.

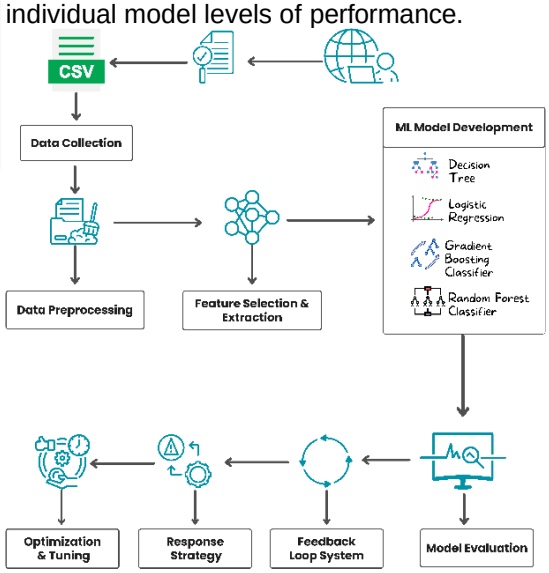


FIGURE 5. System Flowchart

We utilized machine learning algorithms renowned for their classification effectiveness. The Decision Tree offers a clear visualization of decision-making paths and is utilized for its ease of interpretation and capability to handle non-linear data. The Logistic Regression model was chosen for its simplicity and interpretability, providing a benchmark for the project. The Random Forest Classifier was selected for its robustness in handling complex tasks and its feature importance capability. Lastly, the Gradient Boosting Classifier was incorporated due to its sequential error minimization, making it highly adaptive and efficient for varied data types. The Decision Tree these times its visualization force comes from decision levels and paths which in turn ensure that it is used in its ability to deal with the data which is non-linear. The Logistic Regression model is very easy to interpret. We found the Random Forest Classifier function to be more determinant and effective compared to other algorithms that we considered for large or complicated tasks and the feature importance functionality that it presents. Secondly, Gradient Boosting Classifier is the most significant instrument of the designed model which is sequential error minimizing and efficient for diverse data types.

F. MODEL TRAINING

Four machine learning algorithms are engaged in solving the problem of detecting DDoS attack, each algorithm due to its characteristics being the best match for a

classification task.

Logistic Regression: For this case, we will start the Logistic Regression modeling with max_iter = 1000 in order to make enough iterations for the model convergence. This model is essentially a starting point which a relatively plain but a powerful way to distinguish between traffic types representing normal or malicious. Random Forest Classifier: The Random Forest model which is a well-known accurate algorithm on handling even the most complex datasets, are configured and 100 trees resources are assigned. The algorithm gains this feature with the help of performing a feature importance testing, using which we find out the primary attributes that are indicative of DDoS attacks.

To prevent from over-fitting the Decision Tree model, it is limited to a depth of 10. The model goes through a preprocessing phase where numerical features are median-imputed and scaled, while categorical features are imputed with the most frequent values and one-hot encoded. Built-in into the pipeline with the preprocessor. Gradient Boosting Classifier: Considering the Gradient Boosting classifier as one that is adaptable and using a sequential error correction technique it is possible to benefit from this. This model that is very good at bot fixing previous trees faults is a more advanced approach to model training.

G. *MODEL EVALUATION*

We split our dataset into train and test as later we will be using cross validation to validate our results. Accordingly, we will make certain that our model is not overfitting to the training data and that our outputs are generalized. Evaluation of the model using the metrics such as accuracy, precision, recall as well as F1 score and advanced evaluations measures like ROC curves and precision-recall curves. Also, we use confusion matrix to represent the performance of our classifier in the traffic detection process as well.

III. Results

We got the important points from the evaluation of the machine learning models training for DDoS attack detection, showing that our method is working and is a reliable tool.

A. *MODEL PERFORMANCE*

The model's performances were evaluated using a suite of metrics: they are, accuracy, precision, recall, and F1 score. The Random Forest algorithm achieved an accuracy of 99.97%, which confirms it as the most reliable model for DDoS detection in our experiments. The Decision Tree and Gradient Boosting models exhibited the scores as 99.65% and 99.63 %, both of which were highly impressive. However, Logistic Regression timed in with an accuracy of 84.53%, lower than other models and far from being an ideal option for such a complex task.

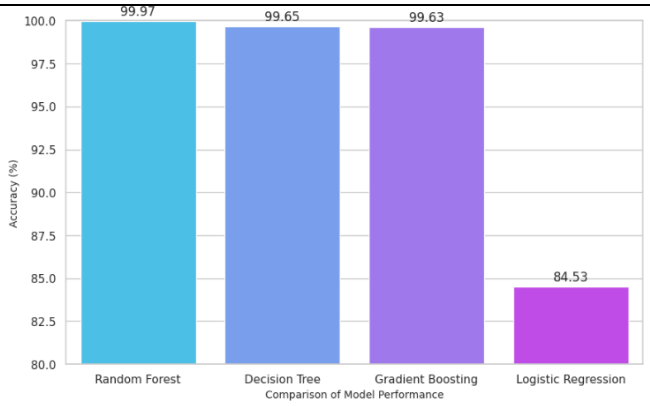


FIGURE 6. Model's Performance

Table 1. Model Performance Metrics

Model	Accuracy	Precision	Recall	F1-Score
Gradient Boosting	99.64%	100%	100%	100%
Decision Tree	99.65%	100%	100%	100%
Logistic Regression	84.53%	0.85%	0.90%	0.88%
Random Forest	**99.98%**	**100%**	**100%**	**100%**

A. *COMPARISON*

When comparing the machine learning models for DDoS attack identification, the random forest classifier beats most of the other models by showing its power through 99.97% accuracy by employing ensemble methods which can especially handle complex data. Those two classifier models show high accuracy as well 99.65% by decision tree and 99.63% by Gradient Boosting according to the result. This reflects on their efficiency in the classification task. Linear Logistic Regression models, which demonstrated an accuracy of 84.53%, come just behind thus hinting that linear models might be having problems coping with the complexity of the technical security data. This analysis asserts that those ensemble techniques outperform others in terms of cyber threat identification amongst network traffic.

B. *INSIGHTS*

The Feature Importance analysis by the Random Forest model helps to understand what the most important predictors of DDoS attacks would be as well as to suggest the effective measures for the networks. High precision and recall scores remain across models, thus, confirming good ability to properly identify the real DDoS attacks, avoid false positives, which is very important, because of the avoiding of unnecessary defensive actions that could disrupt service.

C. *ROC CURVE AND AUC SCORE*

A ROC curve and AUC demonstrate the general potential of the model to classify things using threshold value. The Random Forest model has a high true positive rate, and finally it was able to obtain an AUC close to 1 meanwhile the false positive rate was low, which represents the

highest model performance.

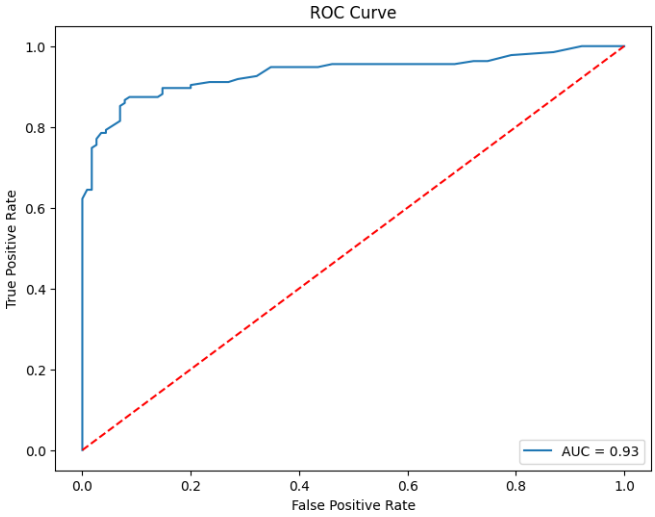


FIGURE 7. The red dashed once is that of a random accurate estimation, which contrasts with the ROC curve of the model near these diagonal lines in a distance because is far from them. The AUC of 0.932 shows a similar picture which is some kind of measure of how accurate the model is; the closer the AUC outcome is to 1, the better the model is in correctly predicting as true positives and is capped at false positives. AUC 0.93 indicate that the model matches with the given data well.

D. FEATURE IMPORTANCE

The important roles of the attributes such as packet count, byte count and particular protocol activities confirmed by the analysis from the Random Forest model have a direct bearing on the enhancements of future features of engineering as well as their selection to achieve improved model accuracy. The outcome of this analysis on the most important variables in the DDoS attack detection. 'Feature 12', 'Feature 2', and 'Feature 5' emerged as major, allowing the modified algorithm of features engineering and construction to reach better performance of the model.

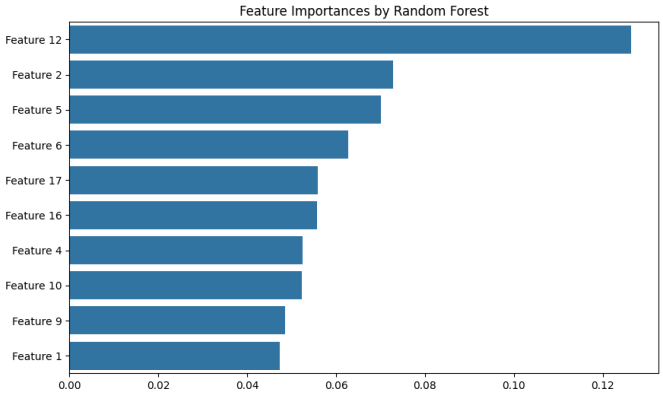


FIGURE 8. Feature Importance by Random Forest, Feature importance ranking based on Random Forest classification, illustrating key network attributes influencing DDoS detection. Higher-ranked features contribute more significantly to attack classification.

E. CONFUSION MATRIX

Visual inspection by using confusion matrix for our models, especially the Random Forest, could indicate the extremely rare misclassification result and such process will reinforce the capability of model usage in real cases. The deep and detailed values of these metrics highly demonstrate accomplishments of our model, which seems to be wired for the accurate identification of DDoS attacks, that is why sources for future profound research and application in network security solutions are built.

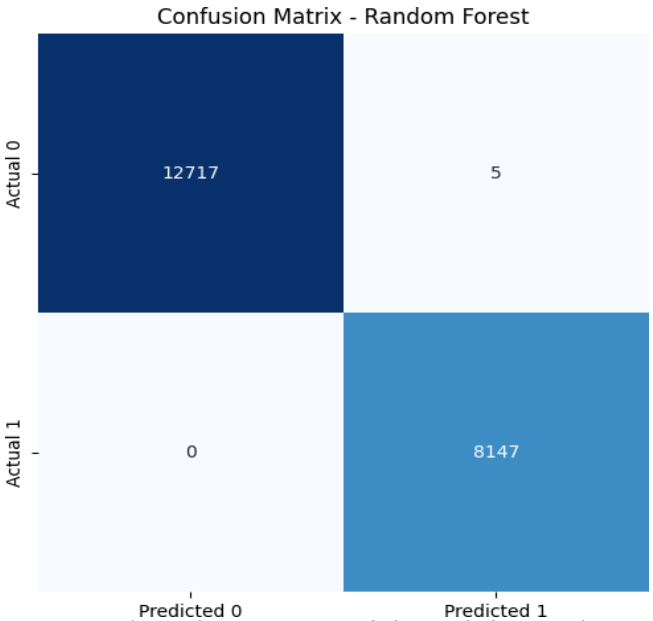


FIGURE 9. The matrix represents a confusion matrix for a Random Forest classifier, which shows the final outcome of class predictions on classifying problems. The number of correct predictions made by the model is represented by the large numbers on the diagonal of the matrix: 8,147 true positive cases (Actual 1, Predicted 1) and 12,717 true negative cases (Actual 0, Predicted 0). The off-diagonal numbers represent the few errors made by the model: 5 actual 0 and 1 predicted, 0 actual 1 and 0 predicted. This means that the model is capable of precisely predicting both cases, and that it is even more reliable in correctly predicting all positive cases with no cases of false negatives being present. Confusion matrix for Gradient Boosting, providing a detailed breakdown of true positive, false positive, true negative, and false negative classifications.

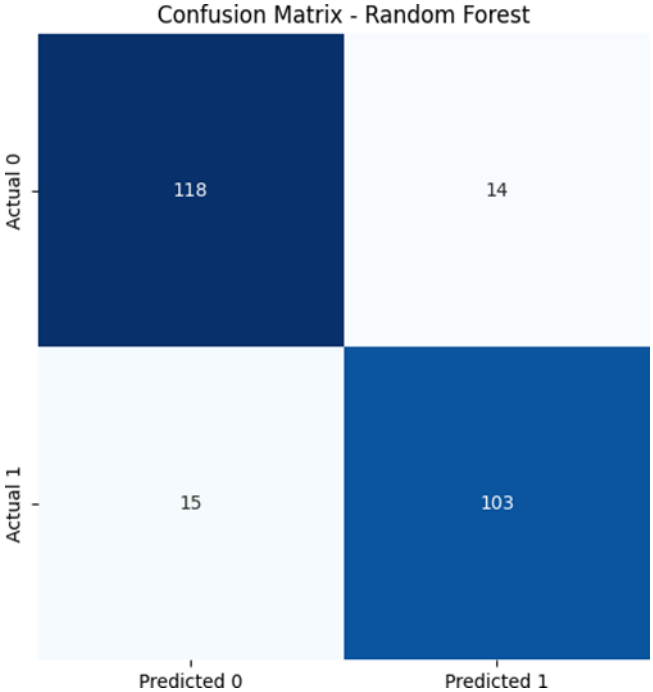


FIGURE 10. This confusion matrix serves as a visual depiction of Random Forest classifier capabilities in the investigation. It shows four key metrics: We have the total of 118 true negatives and 103 true positives. The number of these actual outcomes makes me feel that the prediction is right for both negative and positive classes. An odd reality together with 14 false positives and 15 false negatives evidence when network does not classify the correct target.

IV. Feedback Loop System

The feedback loop system in our study serves as a core that tunes the detection and prediction models for DDoS attacks in real time. These models adjust the accuracy description against real outcomes to obtain an improved model behavior. When differences occur, the model, which is automatically modified to reach higher results in any future predictions; thus, rather than being one-dimensional toward a certain attack type, it is now able to adapt to new and emerging patterns of cyber threats. Whenever disruptions occur, the model is going to be adjusted automatically to the future model for the purpose of improvement, which implies that the model will progress with new attack patterns. Therefore, this ability to improve our model is the vital trait possessed by our defense mechanism in the practical environment to overcome DDoS attacks. It provides the room for the development of the system that can maintain the transformation in the dynamic space of network threats. The Feedback loop in the model is not only bringing into account the amplified precision of the model over the time but also, this learning loop is making the model secure through integrating the hands-on experience in the model's operational system.

Technically, it's the ability to respond to the feedback actively and automatically meaning that it's much easier for such a system to be efficient where the speed and the accuracy are the most vital. This approach provision that our predictive models remain at the bleeding edge of technology so that we are capable of countering the existing and the new threats.

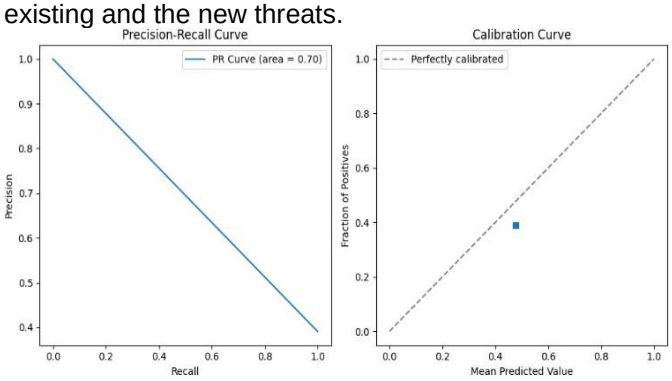


FIGURE 11. Feedback Loop Evaluation: The left panel shows the Precision-Recall Curve and the model balance by assigning the PR AUC of 0.70 that is indicating moderate precision-recall balance. The right side shows the Calibration Curve, a method to check the model prediction reliability; closeness to the dashed line shows that the loop adjustment improved the model's prediction calibration.

V. Response Strategy

Our project employs a differentiated response strategy based on the classification of network traffic into three categories: whitelist, graylist, and blacklist. This approach that we have quite actively followed toward our cybersecurity measures is particularly essential in mitigating the DDoS attack ineffectiveness very effectively.

1) WHITELIST

The traffic on the whitelist is positive and passes through the network without any obstacles. Therefore, internet traffic is trusted and allowed freely. It has on this list already trusted domains which have been confirmed and from which there is no personal data threat. The modeling process is arranged so that security procedures wouldn't disrupt the normal processes of business, always providing access to trusted entities and places.

2) GRAY LIST

The traffic that is listed gray is recognized as suspicious but meanwhile, does not provide conclusive evidence of malicious code. This go-through is basically for more rigorous evaluation and surveillance with advanced processes like throttling or more flexible check. Another intermediary takes a broader view of the situation, thus, helps to exclude possible false positives but not to do that, at the same time, with accurate informers.

3) BLACKLIST

Blacklist is the list of those sources which are judged as malicious, and their traffic is blocked right away. This final action is a key factor for avoiding threats and a most important defense against known and active attack vectors.

VI. System Responsiveness

It was evaluated how fast the system is analyzing and labeling traffic in real-time. It takes 0.5 seconds on average. This processing time is key and proves the applicability of the models to real-time security applications ready for action within a relatively short period.

VII. Conclusion

This project has clearly shown the justification and feasibility of implementing machine learning techniques which detect and classify DDoS attacks. This study was based on a sophisticated multi-model machine learning pipeline which was developed by integration between four models, i.e., a Random Forest, another Decision Tree, Gradient boosting Machine and a

Logistic Regression, on an immense dataset of about 100,000 network traffic entries. This study was proven to have the capability to distinguish most of the normal and malicious network traffic high reliability. The Random Forest Classifier turned out to be the best model because of its capability of finding an accurate solution in a complex dataset and its capability of delving into feature importance metrics.

We arrived at the conclusion that machine learning solution is developing unlike the old, static system based on rules and keeps learning as new rules emerge in the cybersecurity world. For feedback to be more effective, the system must improve over time on newly provided data, which automatically leads to more accuracy and reliability of service providers. The ability to continuously learn and adapt already exists in cybersecurity which, among many other things, proves to be vital since the attack vector is constantly changing and developing. Categorizing the different traffic streams into three fine categories: whitelist, gray list, and the blacklist, will give the system its unique response intended to ensure the least number of false positives.

However, the research had its own difficulties as well. There are many challenges, one of them is computational load of analyzing immense network data in real-time which could make the deployment of such systems in environments with limited resources very difficult. Similarly, although the models were great in detecting known patterns of DDoS attacks their performance decreased when it came to the complex attacks simulating the legitimate traffic, indicating a requirement for improvement in feature extraction the integration of anomaly detection which doesn't rely on system signature-based detection.

VIII. Limitation

It is required to note certain weaknesses of the proposed models: Firstly, higher accurate detection of DDoS attacks is achieved including its certain limitations. First, the datasets selected may contain limited diversification of actual network traffic, and thus the generalization of results could be a problem. It is acknowledged that the results may have been influenced by biases arising from data collection mechanisms, for instance, considering preferred types of attacks would lead to model overfitting. Further, the nature of the created models may include certain levels of computational burdens which may limit its applicability for for real-time high throughput networks. Further research shall be conducted in a sequel of this work to evaluate the efficiency of the models under attack scenarios in addition to identifying ways of increasing the preciseness of the models.

IX. Future Work

Moving ahead, improvement of the DDoS detection system contains several emerging research and development fields. High-level feature engineering is a vital component of the process, because more detailed variables that provide better representation of the global image of the network traffic and precise attack techniques are necessary for the models to perform the right task and identify the legitimacy of network traffic stream and malicious attacks. Deep learning systems might be able to develop the ability to replace the human feature-engineering process by automatically detecting even the most variety patterns and the interactions within the data. Additionally, runtime modification of models for the purpose of analyzing real-time data is considered a vital aspect of practical implementation, emphasizing the necessity to lower the number of computations to enhance data processing speed. Moreover, combining this machine learning system with existing security structures may provide a more integrated defense and utilizes the best inherent characteristics of both the old and the new security procedures for network defense. In all, this project secures defenses against DDoS because of the advanced machine learning techniques applied to the real cyber security challenges and, we can proudly claim that it has academic contribution as well. The system shall continue building on this foundation and if the process continues, we can be looking at an increased number of security measures that will address the current concerns as well as those that will emerge as time passes.

REFERENCES

[1] D. Nanthiya, P. Keerthika, S. B. Gopal, S. B. Kayalvizhi, T. Raja, and R. S. Priya, "SVM Based DDoS Attack Detection in IoT Using Iot-23 Botnet Dataset," in 3rd IEEE International Virtual Conference on Innovations in Power and Advanced Computing Technologies, i-PACT 2021, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/i-PACT52855.2021.9696569.

[2] W. Zhao, H. Sun, and D. Zhang, "Research on DDoS Attack Detection Method Based on Deep Neural Network Model in SDN," in Proceedings - 2022 International Conference on Networking and Network Applications, NaNA 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 184–188. doi: 10.1109/NaNA56854.2022.00038.

[3] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, "Network Traffic Anomaly Detection Using Recurrent Neural Networks," Mar. 2018, [Online]. Available: http://arxiv.org/abs/1803.10769

[4] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018, Institute of Electrical and Electronics Engineers Inc., Aug. 2018, pp. 29–35. doi: 10.1109/SPW.2018.00013.

[5] C. Murukesh, B. Kishore Kannan, A. Thilak kumar, B. Venkat, and V. Haris kumar, "Detection of Distributed Denial of Service Attack using Random Forest Algorithm," in International Conference on Automation, Computing and Renewable Systems, ICACRS 2022 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 382–386. doi: 10.1109/ICACRS55517.2022.10029249.

[6] M. V. Uma, M. Vishnukumar, P. Meganathan, and C. M. Shyamsunder, "Detection and Mitigation of DDoS Attacks in Network Traffic Using Machine Learning Techniques," in 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation, ICAECA 2023, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICAECA56562.2023.10200383.

[7] M. I. Kareem and M. N. Jasim, "DDOS Attack Detection Using Lightweight Partial Decision Tree algorithm," in Proceedings of the 2nd 2022 International Conference on Computer Science and Software Engineering, CSASE 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 362–367. doi: 10.1109/CSASE51777.2022.9759824.

[8] J. Dalvi, V. Sharma, R. Shetty, and S. Kulkarni, "DDoS Attack Detection using Artificial Neural Network," in ICIERA 2021 - 1st International Conference on Industrial Electronics Research and Applications, Proceedings, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICIERA53202.2021.9726747.

[9] A. R. Shaaban, E. Abd-Elwanis, and M. Hussein, "DDoS attack detection and classification via Convolutional Neural Network (CNN)," in Proceedings - 2019 IEEE 9th International Conference on Intelligent Computing and Information Systems, ICICIS 2019, Institute of Electrical and Electronics Engineers Inc., Dec. 2019, pp. 233–238. doi: 10.1109/ICICIS46948.2019.9014826.

[10] S. S. Panwar, Y. P. Raiwani, and L. S. Panwar, "An Intrusion Detection Model for CICIDS-2017 Dataset Using Machine Learning Algorithms," in 2022 International Conference on Advances in Computing, Communication and Materials, ICACCM 2022, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICACCM56405.2022.10009400.

[11] M. I. Sayed, I. M. Sayem, S. Saha, and A. Haque, "A Multi-Classifier for DDoS Attacks Using Stacking Ensemble Deep Neural Network," in 2022 International Wireless Communications and Mobile Computing, IWCMC 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1125–1130. doi: 10.1109/IWCMC55113.2022.9824189.

[12] B. B. Gupta, A. Gaurav, V. Arya, and P. Kim, "A Deep CNN-

based Framework for Distributed Denial of Services (DDoS) Attack Detection in Internet of Things (IoT)," in 2023 Research in Adaptive and Convergent Systems RACS 2023, Association for Computing Machinery, Inc, Aug. 2023. doi: 10.1145/3599957.3606239.

[13] D. Kwon, R. M. Neagu, P. Rasakonda, J. T. Ryu, and J. Kim, "Evaluating Unbalanced Network Data for Attack Detection," in SNTA 2023 - Proceedings of the 2023 on Systems and Network Telemetry and Analytics, Association for Computing Machinery, Inc, Jul. 2023, pp. 23–26. doi: 10.1145/3589012.3594898.

[14] S. Sambangi and L. Gondi, "Multiple Linear Regression Prediction Model for DDOS Attack Detection in Cloud ELB," in ACM International Conference Proceeding Series, Association for Computing Machinery, Oct. 2021. doi: 10.1145/3492547.3492567.

[15] S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, "Linear Regression Based DDoS Attack Detection," in ACM International Conference Proceeding Series, Association for Computing Machinery, Feb. 2021, pp. 568–574. doi: 10.1145/3457682.3457769.

[16] J. Yang, H. Wang, and Y. Lu, "Web Attack Detection through Network-Traffic-Based Feature Engineering and Machine Learning," in ACM International Conference Proceeding Series, Association for Computing Machinery, Dec. 2020, pp. 103–108. doi: 10.1145/3444370.3444555.

[17] A. Alharthi, A. Eshmawi, A. Kabbas, and L. Hsairi, "Network traffic analysis for DDOS attack detection," in ACM International Conference Proceeding Series, Association for Computing Machinery, Nov. 2020. doi: 10.1145/3440749.3442637.

[18] J. Chen, Y. tao Yang, K. ke Hu, H. bin Zheng, and Z. Wang, "DAD- MCNN: DDoS attack detection via multi-channel CNN," in ACM International Conference Proceeding Series, Association for Computing Machinery, 2019, pp. 484–488. doi: 10.1145/3318299.3318329.

[19] M. Shurman, R. Khrais, and A. Yateem, "DoS and DDoS attack detection using deep learning and IDS," International Arab Journal of Information Technology, vol. 17, no. 4A Special Issue, pp. 655– 661, 2020, doi: 10.34028/iajit/17/4A/10.

[20] O. Thorat, N. Parekh, and R. Mangrulkar, "TaxoDaCML: Taxonomy based Divide and Conquer using machine learning approach for DDoS attack classification," International Journal of Information Management Data Insights, vol. 1, no. 2, Nov. 2021, doi: 10.1016/j.jjimei.2021.100048.

[21] I. Sreeram and V. P. K. Vuppala, "HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm," Applied Computing and Informatics, vol. 15, no. 1, pp. 59–66, Jan. 2019, doi: 10.1016/j.aci.2017.10.003.

[22] K. B. Virupakshar, M. Asundi, K. Channal, P. Shettar, S. Patil, and D. G. Narayan, "Distributed Denial of Service (DDoS) Attacks Detection System for OpenStack-based Private Cloud," in Procedia Computer Science, Elsevier B.V., 2020, pp. 2297–2307. doi: 10.1016/j.procs.2020.03.282.

[23] A. Jaszcz and D. Połap, "AIMM: Artificial Intelligence Merged Methods for flood DDoS attacks detection," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 10, pp. 8090–8101, Nov. 2022, doi: 10.1016/j.jksuci.2022.07.021.

[24] N. V. Patil, C. Rama Krishna, K. Kumar, and S. Behal, "E-Had: A distributed and collaborative detection framework for early detection of DDoS attacks," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 4, pp. 1373–1387, Apr. 2022, doi: 10.1016/j.jksuci.2019.06.016.

[25] R. Priyadarshini and R. K. Barik, "A deep learning based intelligent framework to mitigate DDoS attack in fog environment," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 3, pp. 825–831, Mar. 2022, doi: 10.1016/j.jksuci.2019.04.010.

[26] S. Vattikuti, M. R. Hegde, M. Manish, V. Bodduvaram, and V. Sarasvathi, "DDoS Attack Detection and Mitigation using Anomaly Detection and Machine Learning Models," in *CSITSS 2021 - 2021 5th International Conference on Computational Systems and Information Technology for Sustainable Solutions, Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/CSITSS54238.2021.9683214.

[27] R. Raj and S. Singh Kang, "Mitigating DDoS Attack using Machine Learning Approach in SDN," in *Proceedings - 2022 4th International Conference on Advances in Computing, Communication Control and Networking, ICAC3N 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 462–467. doi: 10.1109/ICAC3N56670.2022.10074307.

[28] S. Santhosh, M. Sambath, and J. Thangakumar, "Detection of DDOS Attack using Machine Learning Models," in *Proceedings of the 1st IEEE International Conference on Networking and Communications 2023, ICNWC 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICNWC57852.2023.10127537.

[29] D. Satyanarayana and A. S. Alasmi, "Detection and Mitigation of DDOS based Attacks using Machine Learning Algorithm," in *International Conference on Cyber Resilience, ICCR 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICCR56254.2022.9995773.

[30] G. Sujatha, Y. Kanchal, and G. George, "An Advanced Approach for Detection of Distributed Denial of Service (DDoS) Attacks Using Machine Learning Techniques," in *3rd International Conference on Smart Electronics and Communication, ICOSEC 2022 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 821–827. doi: 10.1109/ICOSEC54921.2022.9951944.

[31] A. Kumar and I. Sharma, "Employing Supervised Learning Techniques for DDoS Attack Detection," in *International Conference on Innovative Data Communication Technologies and Application, ICIDCA 2023 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 684–688. doi: 10.1109/ICIDCA56705.2023.10099834.

[32] Institute of Electrical and Electronics Engineers. Turkey Section. and Institute of Electrical and Electronics Engineers, *HORA 2020 : 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications : proceedings : June 26-27, 2020, Turkey.*

[33] F. Nazarudeen and S. Sundar, "Efficient DDoS Attack Detection using Machine Learning Techniques," in *2022 IEEE International Power and Renewable Energy Conference, IPRECON 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/IPRECON55716.2022.10059561.

[34] IEEE Staff, *2019 Amity International Conference on Artificial Intelligence (AICAI)*. IEEE, 2019.

[35] N. T. Bhutia, H. Verma, N. Chauhan, and L. K. Awasthi, "DDoS Attacks Detection in 'Internet of Medical Things' Using Machine Learning Techniques," in *2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation, IATMSI 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/IATMSI56455.2022.10119428.

[36] N. Chavan, M. Kukreja, G. Jagwani, N. Nishad, and N. Deb, "DDoS Attack Detection and Botnet Prevention using Machine Learning," in *8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1159–1163. doi: 10.1109/ICACCS54159.2022.9785247.

[37] M. A. Mahmood and A. M. Zeki, "Securing IOT Against DDOS Attacks Using Machine Learning," 2020.

[38] S. Thota and D. Menaka, "Importance of Machine Learning Algorithms to Detect Botnet DDoS Attacks," in *Proceedings - International Conference on Augmented Intelligence and Sustainable Systems, ICAISS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 900–903. doi: 10.1109/ICAISS55157.2022.10011016.

[39] R. Pandey, M. Pandey, and A. Nazarov, "Enhanced DDoS Detection using Machine Learning," in *2023 6th International Conference on Information Systems and Computer Networks, ISCON 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ISCON57294.2023.10112033.

[40] M. Kavitha, M. Suganthy, A. Biswas, R. Srinivsan, R. Kavitha, and A. Rathesh, "Machine Learning Techniques for Detecting

DDoS Attacks in SDN," in *International Conference on Automation, Computing and Renewable Systems, ICACRS 2022 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 634–638. doi:10.1109/ICACRS55517.2022.10029110.

[41] A. U. Sudugala, W. H. Chanuka, A. M. N. Eshan, U. C. S. Bandara, and K. Y. Abeywardena, "WANHEDA: A Machine Learning Based DDoS Detection System," in *ICAC 2020 - 2nd International Conference on Advancements in Computing, Proceedings*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 380– 385. doi: 10.1109/ICAC51239.2020.9357130.

[42] V. Deepa and B. Sivakumar, "Detection of DDoS Attack using Multiple Kernel Level (MKL) Algorithm," in *2022 International Conference on Innovative Trends in Information Technology, ICITIIT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICITIIT54346.2022.9744225.

[43] Institute of Electrical and Electronics Engineers. Turkey Section. and Institute of Electrical and Electronics Engineers, *HORA 2020 : 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications : proceedings : June 26-27, 2020, Turkey*.

[44] C. Sathvika, V. Satwika, Y. Sruthi, M. Geethika, S. Bulla, and K. Swathi, "DDoS Attack Detection on Cloud Computing Services using Algorithms of Machine Learning: Survey," in *Proceedings - 7th International Conference on Computing Methodologies and Communication, ICCMC 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1094–1100. doi: 10.1109/ICCMC56507.2023.10083549.

[45] R. Bhargava, Y. Pal Singh, and N. S. Narawade, "Implementation of Machine Learning Based DDOS Attack Detection System," in *2022 3rd International Conference for Emerging Technology, INCET 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/INCET54531.2022.9824036.

[46] Francis Xavier Engineering College and Institute of Electrical and Electronics Engineers, *Proceedings of the International Conference on Smart Systems and Inventive Technology (ICSSIT 2018) : Francis Xavier Engineering College, Tirunelveli, India, date: December 13-14, 2018*.

[47] A. Kazin, "DDoS SDN dataset." 15-Dec-2021.

[48] A. Sebbar and K. Zkik, "Enhancing resilience against DDoS attacks in SDN -based supply chain networks using machine learning," in 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT), 2023, pp. 230–234.

[49] A. Gaurav, B. B. Gupta, K. Tai Chui, V. Arya, and E. Benkhelifa, "A DDoS attack detection system for industry 5.0 using digital twins and machine learning," in 2023 IEEE 12th Global Conference on Consumer Electronics (GCCE), 2023, pp. 1019–1022.

[50] A. Sharma and H. Babbar, "Evaluation and analysis: Internet of things using machine learning algorithms for detection of DDoS attacks," in 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), 2023, pp. 1203–1208.

[51] B. Ozcam, H. H. Kilinc, and A. H. Zaim, "Detecting TCP flood DDoS attack by anomaly detection based on machine learning algorithms," in 2021 6th International Conference on Computer Science and Engineering (UBMK), 2021, pp. 512–516.

[52] N. Jyoti and S. Behal, "A meta-evaluation of machine learning techniques for detection of DDoS attacks," in 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021, pp. 522–526.

[53] R. S. Devi, R. Bharathi, and P. K. Kumar, "Investigation on efficient machine learning algorithm for DDoS attack detection," in 2023 International Conference on Computer, Electrical & Communication Engineering (ICCECE), 2023, pp. 1–5.

[54] Y. Sun, Y. Han, Y. Zhang, M. Chen, S. Yu, and Y. Xu, "DDoS attack detection combining time series-based multi-dimensional sketch and machine learning," in 2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS), 2022, pp. 01–06.

[55] M. Kavitha, M. Suganthy, A. Biswas, R. Srinivsan, R. Kavitha, and A. Rathesh, "Machine Learning Techniques for Detecting DDoS Attacks in SDN," in International Conference on Automation, Computing and Renewable Systems, ICACRS 2022 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 634–638. doi: 10.1109/ICACRS55517.2022.10029110.

[56] Kumar and I. Sharma, "Employing Supervised Learning Techniques for DDoS Attack Detection," in International Conference on Innovative Data Communication Technologies and Application, ICIDCA 2023 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 684–688. doi: 10.1109/ICIDCA56705.2023.10099834.

Date available online: 13-02-2025
Vol. 2, Issue 2 (June December 2024)

Health Predictions Redefined: The Impact of AI on Future Disease Diagnosis

Iqra Muneer[1,2*], Sadia Tariq[1], Muhammad Kashif[2]

[1]Department of Computer Science & Engineering, University of Engineering Technology Lahore, Narowal Campus, Lahore, Pakistan.
[2]Comsats University Islamabad, Sahiwal Campus, Sahiwal, Pakistan.

[*]Corresponding author: Iqra Muneer (e-mail: iqramuneer@uet.edu.pk)

***Abstract*— Healthcare professionals often apply a one-size-fits-all approach in patient care, potentially leading to misdiagnosis, suboptimal treatments, and higher healthcare costs. Machine-learning models have garnered attention for their ability to improve diagnostic accuracy, with numerous studies focusing on machine learning applications for individual disease predictions, such as Type II diabetes, heart disease, kidney disease, and hypertension. However, limited research has tackled the combined prediction of Type I diabetes (standard cases), Type II diabetes (gestational diabetes), and cardiovascular disease, presenting a significant research gap.

To address this gap, we introduce a set of benchmark corpora based on authentic patient records, targeting specific disease categories. The first contribution is a heart disease corpus containing 606 instances. The second and third contributions consist of two separate corpora, each with 849 instances: one focused on standard diabetes cases and the other on gestational diabetes cases. We evaluated these corpora with ten machine-learning algorithms and five deep-learning algorithms, rigorously comparing their performance across common metrics, including accuracy, precision, recall, and $F1$-score. Our results revealed high performance across all models, with top $F1$-scores of 0.785 using Random Forest, 0.790 with Gradient Boosting, and 0.994 using BiLSTM for the combined disease prediction. These findings suggest that the proposed datasets and models provide a robust foundation for accurate and scalable high-risk disease prediction, contributing a valuable, multidimensional approach to personalized patient care.The novelty of our approach lies in the creation and use of region-specific datasets for combined prediction of Type I diabetes, Type II diabetes (gestational), and cardiovascular disease, which has been minimally explored in existing research.

Index Terms—Heart disease prediction Diabetes prediction, Diabetes during Pregnancy, Cardiovascular disease, Healthcare management

1 Introduction

The prevalence of diabetes is on the rise worldwide, even in developed nations mainly due to obesity and stress related to modern lifestyle. Due to its widespread impact, combating diabetes requires a collective effort from healthcare providers, patients, families, and society. The disease incurs significant social, health, and economic costs [1]. The chronic condition of diabetes arises when the body is unable to either produce sufficient insulin or effectively use the insulin produced, leading to high levels of sugar in the blood [2, 3]. The root cause of this condition, also known as ”x syndrome,” is still not completely comprehended by medical professionals. Treatment of diabetes has traditionally focused on symptom monitoring rather than targeting the underlying cause. According to the World Health Organization, approximately 5 percent of the world’s population is affected by diabetes, and this figure is on the rise. In developed countries, diabetes is most common among individuals over 65 years old, while in developing countries, the highest incidence occurs among those aged 45-64 years, with type II diabetes becoming more common among people aged 30-40 years [1]. To leverage the abundance of historical data, data mining techniques can be employed to detect patterns and trends in diabetes, facilitating early detection and prevention. By utilizing data mining, healthcare professionals can efficiently analyse pre-existing data to identify patterns and trends in diabetes. Healthcare providers and public health organizations can utilize the system to prevent and treat high-risk conditions such as diabetes, hypertension, and heart disease [15]. Healthcare practitioners to identify high-risk groups for chronic diseases and to devise tailored treatments to prevent and manage these disorders may use the method. The method may be used by public health organizations to analyse big datasets and identify risk factors for chronic diseases, which can aid in the creation of treatments and policies to prevent and manage these diseases [16, 17, and 18]. Cardiovascular illnesses include heart failure, coronary artery disease, stroke, and other conditions that affect the heart and blood vessels. Numerous factors, including high blood pressure, high cholesterol, smoking, obesity, diabetes, and a family history of the disease, can contribute to cardiovascular disease [13]. In other words, diabetes

is a chronic illness that impairs the body's capacity to metabolize glucose or blood sugar. Type I diabetes, an autoimmune illness commonly identified in infancy, and type II diabetes, are linked to lifestyle related factors, including physical inactivity, obesity, and bad eating practices. Age, family history and poor nutrition are also risk factors for diabetes [14].

The disease prediction component of the system analyses a vast dataset of health and lifestyle parameters to estimate an individual's risk of acquiring heart disease andndiabetes using machine-learning algorithms. To give risk estimates diseases, these algorithms consider characteristics such as age, gender, BMI, blood pressure, cholesterol levels, smoking history, family medical history, and other pertinent data.

Several studies have investigated the use of machine learning for disease prediction type II diabetes prediction, Heart disease prediction, kidney disease prediction, and hypertension detection. However, for the combined prediction of type II (adult diabetes), and type III (Diabetes during Pregnancy) along with cardiovascular disease, formal study has been hardly carried out.

As a first major contribution, we develop a novel benchmark corpus based on real cases of heart patient records containing 606 instances. As another contribution, we have presented two novel benchmark corpora containing 849 instances based on real cases of diabetes in normal patients, and Diabetes during Pregnancy respectively. As a second contribution, all of these corpora were evaluated using 10 various machine learning algorithms Random Forest (RF), Decision Tree (DT), Bernoulli Naive Bayes (BNB), Gaussian Naive Bayes (GNB), Gradient Boosting Classifier (G-BC), AdaBoost (AB), Multilayer Perceptron (M-LP), K-Nearest Neighbour (K-NN), Support Vector Machine (SVM), and Logistic Regression (LR). As another contribution, five different deep learning methods including Long Short Term Memory (LSTM), Bidirectional Long Short Term Memory (BILSTM), Convolution Neural Network (CNN), Gated Recurrent Unit (GRU), and Bidirectional Gated Recurrent Unit (BIGRU). As final and most fruitful contribution, an in-depth, and detailed comparison was performed among the applied algorithms. These datasets have been evaluated using Accuracy, Precision, Recall, and F_1-score. Overall, the proposed system will provide patients with Personalized healthcare management, Disease prediction, Diet recommendation, Performance evaluation and Feedback mechanism in a cost-effective manner. The research paper is organized as follows: Section 2 describes the literature review Section 3 covers the dataset creation methodology. Section 4 presents the experimental setup. Section 5 discusses and analyses the results, and Section 6 concludes the paper.

2 Literature Review

In literature, various efforts have been made to develop novel approaches and datasets for the task of various disease predictions. The following section contains the detail of some prominent attempts made for various disease prediction tasks.

Lahla et al. proposed a novel dataset for the prediction task of diabetes [4]. The dataset consists of 270 records from the Public Health Institute with seven different attributes including: 1) Age, 2) Body Mass Index, 3) Insulin, 4) Serum Insulin in two hours, 5) Glucose: Glucose tolerance test values, 6) Skin Thickness, 7) Blood Pressure, and 8) Number of pregnancies. The dataset was evaluated using machine learning techniques including Na¨ıve Bayes, Support Vector Machine, Decision Trees (DT), and Artificial Neural Networks (ANN). The best performance was obtained with an accuracy of 79% using DT.

Singh et al. proposed a study on heart disease prediction tasks [5]. The authors applied various approaches to a dataset from the UCI repository consisting of 304 records with 13 different attributes including: 1) Age, 2) Sex, 3) Chest pain, 4) Blood Pressure, 5) Fasting blood sugar, 6) Cholesterol, 7) Maximum electric cardiograph, 8) Heart rate, 9) Exercise angina, 10) Depression, 11) Slope of peak exercise segment, 12) Fluoroscopy, and 13) Defect type. The dataset was evaluated using machine learning methods including Logistic Regression (LR), K-Nearest Neighbors (KNN), and Random Forest (RF) Classifier. The best performance was obtained with an accuracy of 88.5% using KNN.

Another study [6] predicts chronic kidney disease using a dataset from UCI that contains 25 different characteristics. The dataset was analyzed using machine learning methods proposed by Pal and colleagues, including LR, DT, and Support Vector Machine (SVM). An accuracy of 97% was the best result obtained using DT.

Lukmanto et al. [7] conducted a study to forecast the onset of diabetes mellitus (DM) using 768 patient data points from the Pima Indian Diabetes Dataset. They utilized fuzzy support vector machines and feature selection to discover DM. Feature selection was used to locate relevant properties in the dataset, which was trained using SVM to provide fuzzy rules. The results showed an optimistic accuracy of 89

Mujumdar et al. proposed a novel dataset for diabetes prediction [8]. The dataset consists of 800 records with nine different input attributes including: 1) Age, 2) Body Mass Index, 3) Insulin, 4) Glucose, 5) Skin Thickness, 6) Blood Pressure, 7) Number of pregnancies, 8) Job type, and 9) Office work. Machine learning techniques such as DT, Gaussian Na¨ıve Bayes (GNB), Linear Discriminant Analysis (LDA), SVC, RF, Extra Trees, Ada Boost (AB), Multi-layer Perceptron (M-LP), LR, Gradient Boosting Classifier (G-BC), and KNN were used to analyze the dataset. The highest result was attained with an accuracy of 96% using Logistic Regression.

The task has been explored in multiple ways, including chronic kidney disease [6], diabetic mellitus [7], diabetes prediction [8, 4], and heart disease [5] with various algorithms including LR, KNN, RF, AB, and M-LP. However, the tasks have never been explored with a variety of deep learning methods. Furthermore, the task has not been explored on datasets based on Pakistan's national disease due to unavailability of datasets. To fulfill this gap, the study proposes a novel benchmark corpus comprising authentic heart patient records, totaling 606 instances. Additionally, we introduced two new benchmark corpora consisting of 849 instances each, derived from real cases of diabetes in both normal patients and during pregnancy. Another significant contribution lies in the evaluation of these corpora using a diverse set of machine learning algorithms, including RF, DT, BNB, GNB, G-BC, AB, M-LP, K-NN, SVM, and LR. Furthermore, we explored five distinct deep learning methodologies, namely LSTM, BILSTM, CNN, GRU, and BIGRU. Finally, our most substantial contribution involves an exhaustive and detailed comparative analysis of the performance of these applied algorithms.

Table 1 Comparison of Disease Prediction Studies

Study	Disease Type	Dataset	Algorithms Used	Key Findings / Novelty
Llaha & Rista [4]	Diabetes Prediction (Type II)	270 records from Public Health Institute, 7 attributes	Naïve Bayes, SVM, DT, ANN	Achieved 79% accuracy with DT; single disease focus
Singh & Kumar [5]	Heart Disease Prediction	304 records, 13 attributes from UCI dataset	LR, KNN, RF	Achieved 88.5% accuracy with KNN; heart disease only, no focus on other diseases
Pal [6]	Chronic Kidney Disease Prediction	UCI dataset with 25 attributes	LR, DT, SVM	Achieved 97% accuracy with DT; specific to kidney disease
Lukmanto et al. [7]	Diabetes Mellitus Prediction	Pima Indian dataset, 768 records	Fuzzy SVM, Feature Selection	Achieved 89% accuracy; focused on diabetes mellitus only
This Study	Combined prediction of Type I & Type II Diabetes (normal & gestational) with Cardiovascular Disease	Three real-world corpora: Heart disease (606 instances), Normal diabetes (849), Diabetes during pregnancy (849)	ML algorithms: RF, DT, SVM, GBC, KNN, etc.; DL models: LSTM, BiLSTM, CNN, GRU, BiGRU	**F_1-score: RF = 0.785, GBC = 0.790, BiLSTM = 0.994**; Unique in combining multiple disease types, region-specific datasets, benchmark corpora for diabetes/-heart disease, and extensive ML/DL comparative analysis

In contrast to prior studies that primarily focus on predicting a single disease, such as diabetes or heart disease, this study uniquely combines Type I and Type II diabetes—including gestational diabetes—with cardiovascular disease prediction in a unified framework, addressing a significant gap in the literature (Table 1). By leveraging real-world, region-specific patient data from Pakistan, the study introduces a novel dataset that is tailored to an underrepresented population, enhancing its relevance and applicability. Notably, three benchmark corpora were developed as part of this research: one for heart disease cases (606 instances), another for normal diabetes cases (849 instances), and a third specifically for diabetes during pregnancy (849 instances). Furthermore, the study undertakes an extensive evaluation of traditional machine learning (ML) and deep learning (DL) models, employing a total of ten ML and five DL algorithms. This comprehensive approach provides nuanced insights into the most effective methods for each disease category, offering a valuable comparative performance analysis. The study achieved an impressive F_1-score of 0.994 with BiLSTM in combined prediction tasks, highlighting the superior predictive power of deep learning techniques over previous studies. Together, these findings underscore the study's contributions to the field by presenting an innovative, multidimensional dataset and a robust algorithmic framework that holds promise for more accurate and scalable high-risk disease prediction.

A	B	C	D	E	F	G	H	I
Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1

Figure 1 Diabetes (During Pregnancy) Dataset

3 Dataset Creation Methodology

This section sheds light that how all proposed corpora were created.

3.1 Diabetes (During Pregnancy) Dataset Collection

In this respect, we gathered information from several hospitals in Lahore, Pakistan, on diabetes in pregnant women. The information on numerous demographic, clinical, and lifestyle aspects was gathered in the form of CSV files. Gestational diabetes, commonly referred to as diabetes during pregnancy, is a disease when a woman experiences high blood sugar levels while she is pregnant. After birth, the disease often goes away. It commonly develops in the second or third trimester. Diabetes in pregnant women dataset contains 849 instances with 9 attributes, including age, number of times pregnant, glucose concentration, blood pressure, skin thickness, insulin level, body mass index, diabetes pedigree function, and the presence or absence of diabetes.

1. Age: This attribute indicates the age of the pregnant mother in years. It has a numerical quality.
2. Pregnancy frequency: This element indicates the pregnant woman's frequency of pregnancies. It has a numerical quality.
3. Glucose concentration: This factor shows the

milligrams per decilitre (mg/dL) glucose level in the blood plasma of the expectant mother. It has a numerical quality.

4. Blood pressure: This feature displays the pregnant woman's diastolic blood pressure (in mm Hg). It has a numerical quality.

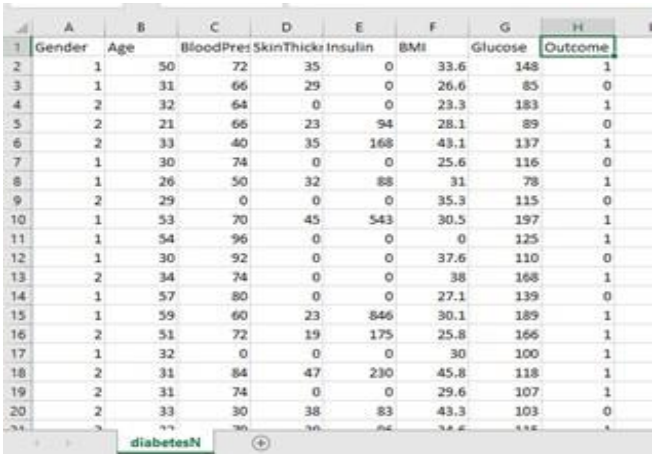
5. Skin thickness: The thickness of the skin on the triceps of a pregnant woman is measured (in millimetres). It has a numerical quality.

6. Insulin level: This factor reveals the amount of insulin (measured in '' U/ml) in a pregnant woman's blood. It has a numerical quality.

7. Body mass index (BMI): This measurement reveals that the body mass index of expecting mother is calculated by dividing her weight in kilograms by her height in meters squared. It has a numerical quality.

8. This property is a representation of the diabetes pedigree function, which estimates the likelihood of diabetes based on family history.

9. Diabetic status: This trait reveals whether or not the expectant mother has diabetes. It is a binary attribute, where a value of one indicates the presence of diabetes and a value of zero indicates the lack of it.



	A	B	C	D	E	F	G	H
1	Gender	Age	BloodPres	SkinThick	Insulin	BMI	Glucose	Outcome
2	1	50	72	35	0	33.6	148	1
3	1	31	66	29	0	26.6	85	0
4	2	32	64	0	0	23.3	183	1
5	2	21	66	23	94	28.1	89	0
6	2	33	40	35	168	43.1	137	1
7	1	30	74	0	0	25.6	116	0
8	1	26	50	32	88	31	78	1
9	2	29	0	0	0	35.3	115	0
10	1	53	70	45	543	30.5	197	1
11	1	54	96	0	0	0	125	1
12	1	30	92	0	0	37.6	110	0
13	2	34	74	0	0	38	168	1
14	1	57	80	0	0	27.1	139	0
15	1	59	60	23	846	30.1	189	1
16	2	51	72	19	175	25.8	166	1
17	1	32	0	0	0	30	100	1
18	2	31	84	47	230	45.8	118	1
19	2	31	74	0	0	29.6	107	1
20	2	33	30	38	83	43.3	103	0

diabetesN

Figure 2 Normal Diabetes Dataset

3.2 Normal Diabetes Dataset Collection

In this dataset, information from several hospitals was collected in Lahore, Pakistan, from regular diabetics. The information on numerous demographic, clinical, and lifestyle aspects was gathered in the form of CSV files. The normal diabetes dataset contains 849 tuples with 8 attributes, including age, gender, glucose concentration, blood pressure, skin thickness, insulin level, body mass index, and the presence or absence of diabetes.

3.3 Heart Disease Dataset Collection

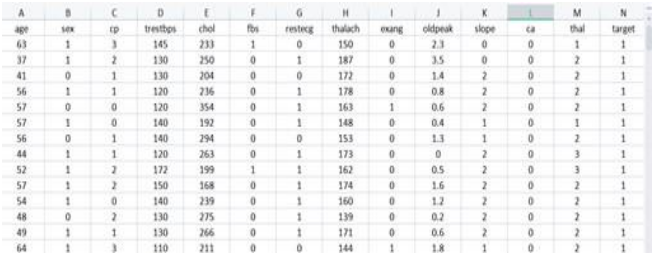
In this respect, information from several hospitals was collected in Lahore, Pakistan, from heart patients. The information on numerous demographic, clinical, and lifestyle aspects was gathered in the form of CSV files. The heart disease dataset has 606 tuples with 14 attributes in it, including age, sex, the type of chest pain, resting blood pressure, cholesterol level, fasting blood sugar, electrocardiogram results, maximum heart rate reached, exercise-induced angina, ST depression caused by exercise relative to rest, slope of the peak exercise ST segment, number of major vessels coloured by fluoroscope, thallium stress test results, and absence or presence of heart disease.

3.3.1 Dataset Standrization

All of abrove proposed datasets are standardized in CSV format and is readily available for research purposes. It is licensed under a Creative Commons CC-BY-NC-SA license, allowing for free and open use while ensuring proper attribution and prohibiting commercial use without permission. This corpus can be accessed from the available link for the reviewers.

1. Age
2. Sex
3. Type of chest pain experienced (categorized into 4 values)
4. Resting blood pressure measurement
5. Serum cholesterol level in mg/dl
6. Fasting blood sugar level, with values greater than 120 mg/dl indicating the presence of high blood sugar
7. Results of a resting electrocardiogram (ECG), with values categorized as 0, 1, or 2
8. Maximum heart rate achieved during exercise
9. Presence or absence of exercise-induced angina
10. ST depression induced by exercise relative to rest (known as old peak)
11. Slope of the peak exercise ST segment
12. Number of major blood vessels (ranging from 0-3) colored by fluoroscopy.
13. thal: 0 = normal; 1 = fixed defect; 2 = reversible defect

Figure 3 Normal Diabetes Dataset Parameters



A	B	C	D	E	F	G	H	I	J	K	L	M	N
age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1

Figure 4 Heart Disease Dataset

4 Experimental Setup

This section presents the experimental setup, including applied algorithms, evaluation measures, and evaluation methodology for high disease prediction tasks including diabetes predictions, and heart disease prediction.

4.1 Evaluation Measures

The most commonly used evaluation measures include: Accuracy, Precision, Recall, and F_1 measures. Accuracy is calculated by dividing the number of accurate predictions by the total number of predictions, the model produced [9]. The formula for Precision is given below.

$A = (TP + FP)/(TP + FP + FN + TN)$ (1)

Precision (P) can be defined as the proportion of true positive predictions from all the positive cases [10].

$p = TP/(TP + FP)$ (2)

Recall (R) is defined as the proportion of correctly identified positive cases [11].

$R = TP/(TP + FN)$ (3)

F_1 measure is the harmonic mean of precision (P) and recall (R). F_1 measure is commonly used as an evaluation measure for cases where datasets are unbalanced. It is defined as the harmonic mean of two other measures, Precision (P) and Recall (R) [12].

$F_1 = (2 * P * R)/(P + R)$ (4)

4.2 Evaluation Methodology

For both normal diabetes and diabetes during pregnancy, the problem with diabetes prediction was handled as a supervised text classification task. Two degrees of discrimination were intended by the categorization task: (1) diabetes, and (2) non-diabetes. The challenge of predicting heart disease was approached similar to supervised text classification assignment. Two degrees of discrimination were intended by the categorization task: (1) heart patients and (2) non-heart patients. RF, DT, BNB, GNB, G-BC, AB, M-LP, K-NN, Support Vector Machine, and Logistic Regression were among the ten machine-learning techniques utilized. we have proposed and developed our own CNN based model with 64 filters at convolution layer, 3 hidden layer with activation function Relu and sigmoid at final layer with pool size 2. The network was trained using 100 epochs with Adam optimizer using 10-fold cross-validation. Experiment 3, 4, 5, and 6 we have proposed and devloped LSTM, GRU, BILSTM, BIGRU with same hyperparmeters and parameter as CNN. Table 2 shows the parameters that was used for Deep Learning Methods.

Table 2 Hyperparameter Settings for Deep Learning Models

Hyperparameter	Value
Sequence Length	100
Batch Size	64
Learning Rate	0.001
Optimizer	Adam
Epochs	50
Loss Function	Binary Cross-Entropy
Activation Function	Softmax
Dropout Rate	0.5
Hidden Units (RNN Layers)	128
Kernel Size (CNN)	3x3
Stride (CNN)	1x1

In order to more accurately gauge the performance of machine learning algorithms, K-fold cross-validation was brought into action. For each experiment, K was set to a standard value of 10. The performance was reported using the weighted-average scores of Accuracy, Precision, Recall, and F_1.

5 Results and Analysis

Tables 1, 2, and 3 show the summarized results obtained by applying various machinelearning algorithms for Normal Diabetes Prediction, Diabetes during Pregnancy, and Heart Disease Prediction tasks respectively. The weighted-average F_1 scores were presented as a concluding measure for all tasks due to imbalanced datasets. The Table 1 shows, the best result with $F_1 = 0.785292$ using RF was obtained for Normal Diabetes Prediction. The performance comparison of all applied machine-learning algorithms for Normal Diabetes Prediction is evident from Table 1.

Table 3 Summarized Results of Normal Diabetes Prediction

ML Algorithm	F_1-Score
RF	**0.785291**
GNB	0.749944
G-BC	0.774546
LR	0.752286
M-LP	0.699831
DT	0.741640
K-NN	0.727619
SVC	0.626379
AB	0.745630
BNB	0.519567
CNN	0.513983
LSTM	0.526037
BILSTM	0.533086
GRU	0.578746
BIGRU	0.586426

Similarly, Table 2 shows, the best result with $F_1 = 0.790618$ using G-BC was obtained for Diabetes during the Pregnancy Prediction task. The performance comparison of all applied machine-learning algorithms for the Diabetes during Pregnancy Prediction task is clear from Table 2. Likewise, table 3 shows the best result with $F_1 = 0.994$ using LSTM was obtained for Heart Disease Prediction. The performance comparison of all applied 10

Table 4 Summarized Results on Diabetes during Pregnancy

ML Algorithm	F_1-Score
RF	0.780245
GNB	0.747802
G-BC	**0.790618**
LR	0.754727
M-LP	0.694850
DT	0.734672
K-NN	0.731603
SVC	0.608957
AB	0.750484
BNB	0.519567
CNN	0.521882
LSTM	0.528174

BILSTM	0.529484
GRU	0.580892
BIGRU	0.595866

machine-learning algorithms for Heart Disease Prediction can is demonstrated in Table 3.

Table 5 Summarized Results of Heart Disease Prediction

ML Algorithm	F_1-Score
RF	0.958783
GNB	0.826155
G-BC	0.937157
LR	0.837152
M-LP	0.821902
DT	0.945552
K-NN	0.711843
SVC	0.696894
AB	0.866180
BNB	0.826238
CNN	0.989846
LSTM	0.993246
BILSTM	**0.994382**
GRU	0.993246
BIGRU	0.993246

Among all machine learning algorithms, RF has shown outstanding performance for normal diabetes prediction, and heart disease prediction. The reason for the best performance of RF is due to following reasons.

5.1 RF

RF is an ensemble learning technique, combines multiple decision trees to make predictions. Each tree is trained on a different subset of the data, and the final prediction is determined by combining the individual tree predictions. This ensemble approach reduces over-fitting and improves the ability to generalize to new data. It also provides a measure of feature importance, indicating the relative significance of each feature in making accurate predictions. This information aids in identifying the most relevant features for prediction tasks, leading to better feature selection and engineering. By focusing on the most informative features, Random Forest enhances predictive performance. One of RF's strengths is its capability to capture nonlinear relationships between features and the target variable. Unlike linear models, Random Forest can model complex interactions and nonlinear patterns in the data. This flexibility is particularly advantageous for diabetes prediction, where the relationship between input features like glucose levels, BMI, and age, and the presence of diabetes may not follow a linear pattern. Random Forest exhibits robustness in handling outliers and missing data. The ensemble nature of the algorithm diminishes the influence of outliers on final predictions. Additionally, Random Forest can handle missing data by utilizing surrogate splits and imputing missing values based on other variables in the data set. This robustness allows for good performance even when the data has imperfections. To combat variance and over-fitting, Random Forest averages predictions from multiple trees. Each tree is trained on a different bootstrap sample of the data, and during the tree-building process, only a random subset of features is considered at each split. These randomization techniques reduce the correlation between individual trees and mitigate the risk of over-fitting. RF is also scale-able and efficient, making it suitable for large data sets with numerous features. The training process can be paralleled since the individual trees in the ensemble can be trained independently. This scalability and efficiency make Random Forest a practical choice for diabetes prediction, heart disease and other machine learning tasks, enabling faster processing and analysis. Table 2 shows, the best result with $F_1 = 0.790618$ using G-BC was obtained for Diabetes during the Pregnancy Prediction task among all machine learning algorithms. The possible reasons for the better

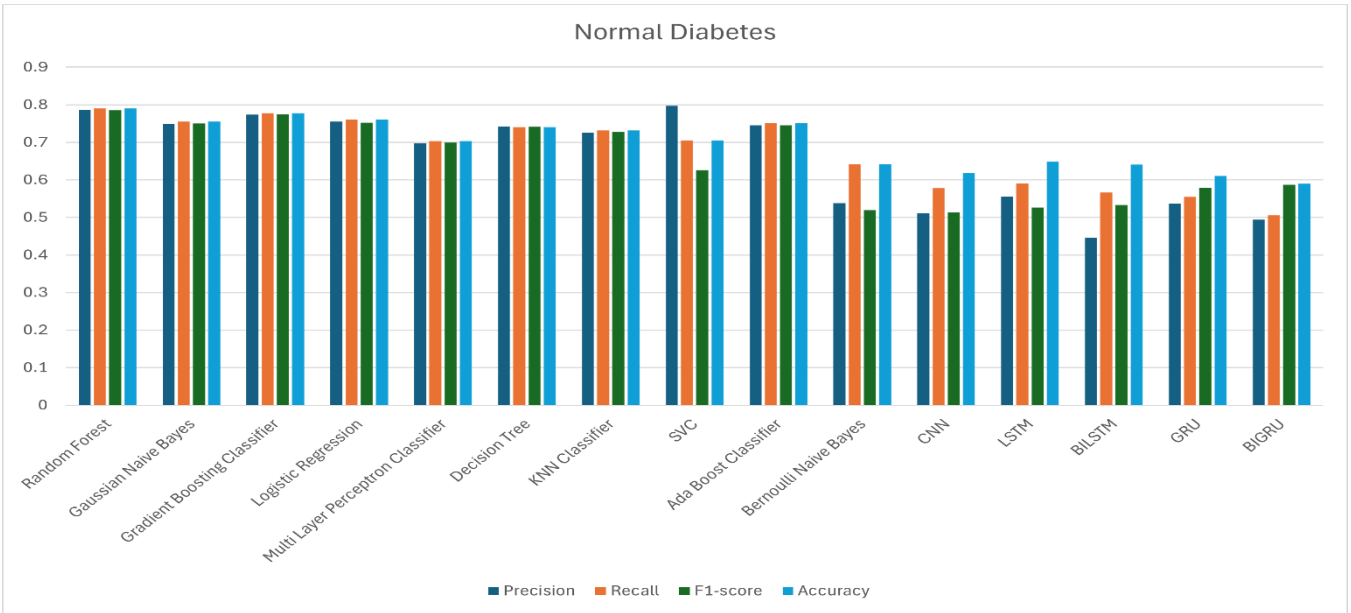


Figure 5 A performance comparison for Normal Diabetes Prediction tas

performance are as follows.

5.2 G-BC

G-BC combines multiple weak learners, such as decision trees, to create a strong predictive model. By adding learners sequentially to correct the mistakes of previous ones, GBC captures complex relationships in the data, enhancing predictive accuracy. G-BC utilizes a gradient descent optimization algorithm in training. By iteratively adjusting the model's parameters along the steepest descent of the gradient, GBC minimizes a loss function. This optimization method helps GBC find an optimal solution,reducing bias and variance and improving predictive accuracy.

It also effectively captures nonlinear relationships between features and the target variable, similar to Random Forest. It models complex interactions and nonlinearity in the data, which is advantageous for diabetes prediction, where the relationships between health indicators and diabetes presence can be nonlinear.

G-BC provides a measure of feature importance, allowing identification of the most relevant features for diabetes prediction. By focusing on these crucial features, GBC prioritizes and utilizes the informative aspects of the data, leading to improved predictions.

G-BC incorporates regularization techniques to prevent over-fitting and improve generalization performance. Methods like shrinkage/learning rate and feature subsampling control model complexity. These techniques reduce over-fitting and enable GBC to generalize well to unseen data, resulting in improved performance.

G-BC effectively handles imbalanced datasets, which are common in diabetes prediction and real-world applications. By assigning appropriate weights or using specialized loss functions, GBC gives more importance to the minority class (e.g., diabetes-positive cases). This ensures better predictive accuracy for both classes, addressing imbalanced data challenges.

Scalability and Efficiency: GBC is scalable and efficient, making it suitable for diabetes prediction with large data sets and numerous features. Optimized implementations paralleling the training process and utilize computational resources efficiently. This scalability allows GBC to handle complex diabetes prediction tasks effectively.

5.3 BILSTM

BiLSTM networks excel in predicting heart disease owing to their proficiency across several critical domains.

Primarily, they specialize in capturing prolonged dependencies within sequential data, pivotal for grasping the intricate interplay between historical health indicators and future risk over multiple time frames in the progression of heart disease. This capability is inherent in their architecture, which sustains a memory state over time. Moreover, BiLSTM networks process input sequences in both forward and backward directions, enabling them to assimilate context from past and future data points concurrently. This bidirectional approach is pivotal for capturing holistic patterns and dependencies, thereby enriching our comprehension of the evolution of heart disease
over time.

Additionally, BiLSTM networks autonomously discern relevant features from input sequences during training. In the context of heart disease prediction, these features might encompass a wide array of physiological measurements such as heart rate, blood pressure, and cholesterol levels—key indicators of cardiovascular health.

Furthermore, BiLSTM networks demonstrate exceptional adaptability in handling input sequences of variable lengths, a crucial attribute in

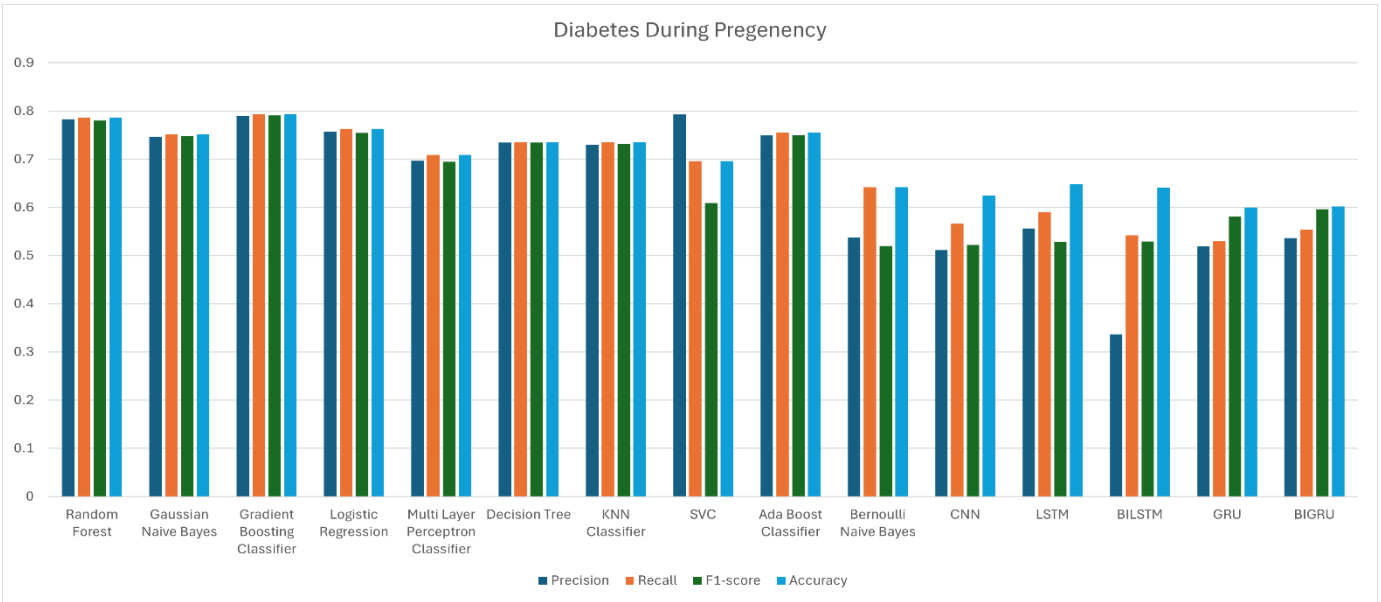


Figure 6 A performance comparison for Diabetes during Pregnancy Prediction task

healthcare data where the frequency and timing of health measurements often vary among patients. This adaptability empowers the model to accommodate diverse data formats and capture personalized disease progression patterns.
Lastly, BiLSTM networks exhibit robust learning capabilities even in scenarios with limited data—a common challenge in medical research due to the scarcity of largescale labeled datasets. Leveraging temporal dependencies within the data, BiLSTM networks efficiently utilize available information to make precise predictions

5.4 Best Methods

Our finding concludes that RF and G-BC and BILSTM are best suited for the classification tasks specially Diabetes Prediction and Heart Diabetes Prediction respectively. Figure 5 shows a detailed performance comparison for Normal Diabetes Prediction among all measures for all applied machine-learning algorithms.
Figure 6 shows a detailed performance comparison for Diabetes during the Pregnancy Prediction task among all measures for all applied machine-learning algorithms.
Figure 7 shows a detailed performance comparison for Heart Disease Prediction among all measures for all applied machine-learning algorithms.

5.5 Findings

Random Forest (RF) and BiLSTM performed best due to their respective strengths in handling feature importance and sequential data.
RF excels in handling diverse feature types and automatically identifies feature importance, making it highly effective for tasks like normal diabetes and heart disease prediction, where numerous variables contribute to the outcome. Its ensemble nature aggregates the results of multiple decision trees, which helps improve stability and robustness against overfitting.
BiLSTM, on the other hand, performs exceptionally well with sequential data, as it can capture long-term dependencies in the input sequences. This is particularly valuable for time-series or sequential prediction tasks like diabetes during pregnancy, where the relationship between past events and current outcomes is critical. By processing data in both forward and backward directions, BiLSTM enhances the model's ability to learn from past and future context, resulting in more accurate predictions.
Therefore, RF's success in handling feature importance and BiLSTM's ability to process sequential data contribute significantly to their top performances in the respective tasks.

6 Conclusion

A uniform approach to patient care, which can lead to an incorrect diagnosis, is often applied in healthcare, especially in diagnosis/prediction-related research, resulting in inadequate treatment and higher healthcare expenses. Machine learning methods in diabetes research can significantly improve diabetes research by effectively predicting diabetes, identifying risk factors, and providing personalized treatment. The study proposes three novel benchmark corpora based on real cases of heart patient records, normal diabetes patients, and diabetes during pregnancy, respectively. As a major contribution, all of these corpora were evaluated using 10 different machine learning algorithms, including RF, GNB, G-BC, Logistic Regression, M-LP, DT, KNN Classifier, SVC, Ada Boost Classifier, and BNB. The study provides a detailed and in-depth comparison of

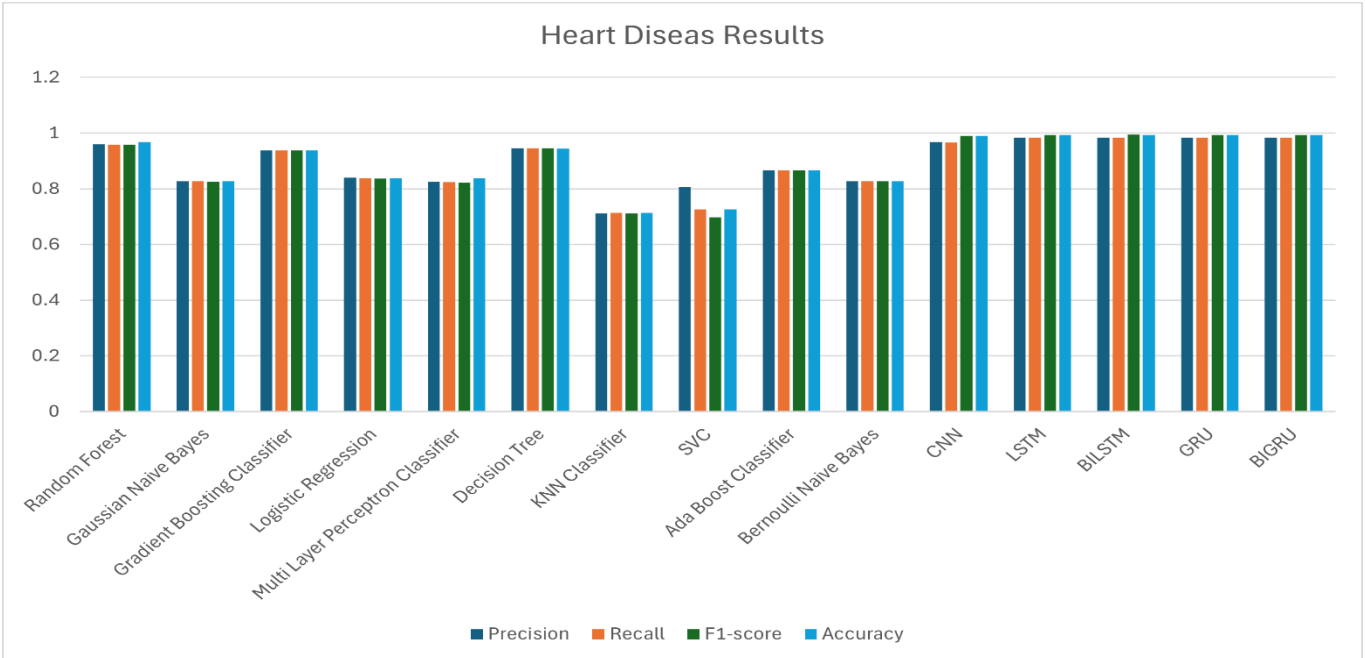


Figure 7 A performance comparison for Heart Disease Prediction task

applied machine learning algorithms for high-risk disease prediction, such as Heart Disease Prediction, Normal Diabetes Prediction, and Diabetes during Pregnancy Prediction tasks. The findings from this research have the potential to revolutionize healthcare practices by enabling more accurate, personalized, and cost-effective disease predictions, ultimately improving patient outcomes and reducing healthcare costs.

7 Conflict of Interest

The authors have no relevant financial or non-financial interests to disclose. The authors have no conflicts of interest to declare that are relevant to the content of this article. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no financial or proprietary interests in any material discussed in this article.

References

[1] http://www.ishp.gov.al/wpcontent/uploads/205/kalendar

[2] https://www.familjadheshendeti.com/semundja-e-sheqerit-diabeti-te-femrat/S.

[3] Bo He, Kuang-i Shu, and Heng Zhang, "Machine Learning and Data Mining in Diabetes Diagnosis and Treatment," IOP Conference Series: Materials Science and Engineering, Volume 490, Issue 4, IOP Conf. Series: Materials Science and Engineering 490 (2019) 042049 IOP, doi:10.1088/1757899X/490/4/042049.

[4] Llaha, Olta, and Amarildo Rista. "Prediction and Detection of Diabetes using Machine Learning."

[5] Singh, Archana, and Rakesh Kumar. "Heart disease prediction using machine learning algorithms." 2020 International Conference on Electrical and Electronics Engineering (ICE3). IEEE, 2020.

[6] Pal, Saurabh. "Chronic Kidney Disease Prediction Using Machine Learning Techniques." Biomedical Materials and Devices (2022): 1-7.

[7] Lukmanto, Rian Budi, Ariadi Nugroho, and Habibullah Akbar. "Early detection of diabetes mellitus using feature selection and fuzzy support vector machine." Procedia Computer Science 157 (2019): 46-54.

[8] Mujumdar, Aishwarya, and V. Vaidehi. "Diabetes prediction using machine learning algorithms." Procedia Computer Science 165 (2019): 292-299.

[9] Chelaramani, Sahil, et al. "Multi-task knowledge distillation for eye disease prediction." Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021.

[10] Y. HaCohen-Kerner, Z. Gross, and A. Masa, "Automatic extraction and learning of keyphrases from scientific articles," In International Conference on Intelligent Text Processing and Computational Linguistic, vol. 3406, no. February 2005, pp. 657–669, 2005, doi: 10.1007/978-3-540-30586-6-74.

[11] N. Pendar and E. Cotos, "Automatic identification of discourse moves in scientific article introductions," no. June, pp. 62–70, 2008, doi: 10.3115/1631836.1631844.

[12] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," Lr. 2018 11th Int. Conf. Lang. Resour. Eval., pp. 3483–3487, 2019.

[13] RTA-CSIT Benjamin, E. J., Muntner, P., Alonso, A., Bittencourt, M. S., Callaway, C. W., Carson, A. P., ... Virani, S. S. (2019). Heart Disease and Stroke Statistics—2019 Update: A Report From the American Heart Association. Circulation, 139(10), e56-e528. doi: 10.1161/CIR.0000000000000659.

[14] American Diabetes Association. (2021). Classification and Diagnosis of Diabetes: Standards of Medical Care in Diabetes—2021. Diabetes Care, 44(Supplement 1), S15-S33. doi: 10.2337/dc21-S002.

[15] Fletcher, B. R., Hinton, L., Hartmann-Boyce, J., Roberts, N. W. (2016). A systematic review of healthcare providers' and patients' diabetes attitudes, perceptions, and expectations and how these influence patient outcomes. Diabetes Research and Clinical Practice, 113, 1-8. doi: 10.1016/j.diabres.2016.01.024.

[16] Lomotan, E. A., Hoeksel, R. (2018). Leveraging health information technology to improve management of chronic conditions in underserved populations. Current Diabetes Reports, 18(6), 32. doi: 10.1007/s11892-018-1009-3.

[17] Mishra, S. R., Neupane, D., Bhandari, P. M. (2019). Health information systems in developing countries: Challenges in implementing and sustaining the impact. Global Health Action, 12(1), 1559497. doi: 10.1080/16549716.2018.1559497.

[18] Patel, A., Raju, D. (2019). Use of health information technology in chronic disease management: A literature review. International Journal of Medical Informatics, 130, 103941. doi: 10.1016/j.ijmedinf.2019.07.017.

Date available online: 13-02-2025
Vol. 2, Issue 2 (June - December 2024)

A Design-Oriented Classification of Microservice Smells

Junaid Aziz[1], and Ghulam Rasool[1]

[1]Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore 54000, Pakistan

Corresponding author: Junaid Aziz (e-mail: mjunaidaziz@gmail.com).

***Abstract*— Context:** Introduction of bad smells can generate negative consequences on the quality of microservices. It is essential to gather state-of-the-art knowledge on these smells and understand the challenges they present. This will benefit researchers and practitioners in mitigating the consequences of smells in microservice-based systems. **Objective:** The main goal of this study is to present a comprehensive catalogue of microservice smells. **Method:** To document the advancements and best practices in the field of microservice smells., we performed a multivocal literature review study incorporating both academic and grey literature sources. We systematically analyzed 34 studies published from the beginning of 2014 until the end of 2023 by following standard guidelines. **Results:** 38 bad smells in microservices are identified and cataloged in 10 different types. **Conclusion:** Research gaps and open challenges are highlighted in this study. This will give directions to other researchers and practitioners towards addressing challenges posed by smells in microservices.

Index Terms—Microservice architecture; Bad smells; Anti-patterns;

I. INTRODUCTION

Microservice architecture (MSA) is powerful and popular architecture style for polyglot applications built with the composition of small services, called “microservices”. In this architecture, the core concept of high cohesion and loose coupling is applied by making these services completely independent in development and deployment. Besides, every service is required to run its processes on its own and communicate with other services via lightweight mechanisms. Service splitting, testing, integration of services, logging, and monitoring are some of the technical challenges that need to be addressed with the help of design patterns while adopting MSA [1].

Antipatterns or bad smells are symptoms that may indicate a deeper quality problem in the system design or code [2]. These smells can be found at various levels of abstraction, such as architecture, design, and code [3]. Code smells are flaws in the design of software that makes it difficult to comprehend and maintain compatibility, resulting in less resilient and compromised system development [4]. Design smells can influence a set of classes in a design framework and highlight violations of design principles such as the principle of circular dependencies [5]. The architecture smell is a higher-level system design issue. This is a sort of technical debt that can have a negative impact on the overall maintainability of a software system. Compared to refactoring code and design smells, refactoring architecture smells takes more time and effort [6].

To counter smells in applications, the software engineering community has explored various ways including proposing catalogs of smells for MSA. However, the scattered information about such resources poses a challenge to both researchers and practitioners while developing appropriate methods, techniques, and tools concerning microservice smells or antipatterns. Hence, analyzing and synthesizing information from academic and grey literature might help the software development community in comprehending the current state of knowledge on microservice smells. Recently, a tertiary study [48] is performed to consolidate a catalog of microservice smells extracted only from academic literature. Microservice smells reported in grey literature are missing in their study. This multivocal literature review (MLR) fill this gap by extending their work. The objective of this MLR is to consolidate both academic and industrial knowledge in order to capture the state of art and practice on microservice smells by including only those microservice smells which have been frequently discussed among researchers and practitioners. The following are the major novel contributions of this study:

- Highlight most frequently studied microservice smells in the literature
- Present a catalog of microservice smells based on their types

The remainder of this paper is arranged as follows. In related work we discuss the existing studies performed on microservice smells and highlight their shortcomings. Research methodology followed in the study is described in survey methodology section. Results of the study along with discussion are presented in results and discussion section. Conclusions section outline the outcome of this study and potential future directions.

II. RELATED WORK

There have been few attempts aiming at reviewing the state-of-the-art and current practices on microservice smells. An overview of these studies is illustrated here.

Neri et al. [7]conducted a systematic review of academic and grey literature to identify the most well-known architectural smells in microservices. They also proposed a taxonomy of these smells by organizing them based on four design principles such as independent deployability, horizontal scalability, isolation of

failure, and decentralization. Taxonomy proposed in their study is missing a lot of microservice smells that have discovered recently such as no service template, local logging, influential service, etc.

Soldani et al. [8]discovered through a grey literature review that microservices-based applications have to face design, development, and operational challenges. They discovered that in such applications, business logic is heavily dispersed among a large number of microservices that are all evolving independently. They also found that there is a lack of methodology to quantify and minimize bad design decisions in such types of applications. Their study was performed with a focus on just technical challenges faced by microservices.

Bogner et al. [3] conducted a literature review of microservices bad smells and antipatterns. Their survey was not only conducted on a limited number of digital libraries but also included antipatterns and bad smells highlighted by researchers only. Studies from grey literature were not included in their search process. Moreover, they did not report or classify the harmfulness of antipatterns.

Tighilt et al. [9]conducted a literature review of published articles and analyzed different open-sourced projects. They presented a catalog of microservice antipatterns (or smells) along with their implementation and refactoring solutions to remove them. However, the viability of refactoring solutions proposed in their study was not empirically validated. Besides, they did not mention smells related to testing and organization.

Carrasco et al. [21] found 9 bad smells related to MSA and its migration by digesting different sources from the academia and grey literature. Their study revealed some common best practices as potential solutions for the architecture and migration bad smells but ignoring security and monitoring smells. Additionally, their search for academic and grey literature sources was performed using search engines only. This gives rise to doubts about the credibility and completeness of results.

The authors conducted a tertiary study aiming not only to compile a comprehensive catalog of overlapping microservice smells but also to categorize them [48]. Their catalog is based on material drawn from academic literature, overlooking the smells reported in grey literature. This restriction may impact the comprehensiveness of the catalog, as grey literature often encompasses valuable insights and practical experiences that contribute to an in depth understanding of microservice smells.

Our work is different from these studies as we are aiming here to build a unified catalog of all microservice smells reported in academic and grey literature both. This study presents these smells in a classified form that has been generated based on the nature of each smell. This will help in achieving standardization vis-à-vis microservices as suggested in [45].

III. RESEARCH METHODOLOGY

Multivocal literature review (MLR) is a type of literature review that incorporates both academic and grey literature sources. Academic literature includes material that is peer-reviewed such as papers published in journals and conferences. Grey literature consists of material that is publicly available in different forms such as blogs, videos, white papers, books, etc. Unlike academic literature, grey literature usually does not undergo a rigorous peer-review process. MLRs are useful for both researchers and practitioners because they summarize the current state of the art from academic and grey literature on a specific topic.

We have followed the guidelines proposed by Garousi et al. [10] to perform this MLR, which are based on systematic literature review (SLR) guidelines suggested by Kitchenham et al. [11]. As per the guidelines, the MLR review process involves three major stages: planning the review, conducting the review, and reporting the review.

A. PLANNING THE MLR

The objective of this study is to capture the state of the art and practices in cataloguing microservice smells. During initial observation it is found that due to their strong interest on microservices both researchers and practitioners have contributed a lot through academic and grey literatures. Hence, instead of conducting either systematic literature review or mapping study, a comprehensive MLR is conducted to capture the state of the art and practices in line with the objectives of this study. To achieve this, we classify peer-reviewed papers (i.e., journal and conference articles) as academic literature and other studies (i.e., blog posts, industrial whitepapers, articles, videos, and books) as grey literature. Moreover, we have formulated the following research questions for this study: -

RQ: How much attention different types of microservice smells have received from academia and industry?

Rationale – By consolidating the list of reported smells from both academic research and industry sources, our goal is to categorize the smells found in microservices-based applications. This approach will facilitate the creation of a unified catalog of these smells which is currently lacking.

B. CONDUCTING THE MLR

We employed IEEE Xplore, ACM, Springer, ScienceDirect, DBLP and Scopus to search papers in the academic literature which are widely used databases and digital libraries to extract computer science and software engineering publications [25]. To look for grey literature (e.g., books, blog posts, videos, whitepapers), we used Google, Bing and DuckDuckGo search engines. The reason for using these search engines is that they remain consistent over a period of time and a considerable difference in the results produced by them is witnessed with Google standing apart [107]. We looked for published studies between the beginning of 2014 (when Lewis et al. [1] introduced microservices) and the end of 2023. The search string "smell OR antipattern OR anti-pattern OR debt OR anomal) AND (microservice OR micro-service" was structured according to the criteria suggested by Petersen et al. [13]. The search string includes keywords from each aspect of our study problem. We ran the search string on academic and grey literatures independently. Final selection of studies from both academic and grey literatures were merged later. The stages of our search and selection process for academic literature and grey literatures can be seen in Figure 1. Authors performed

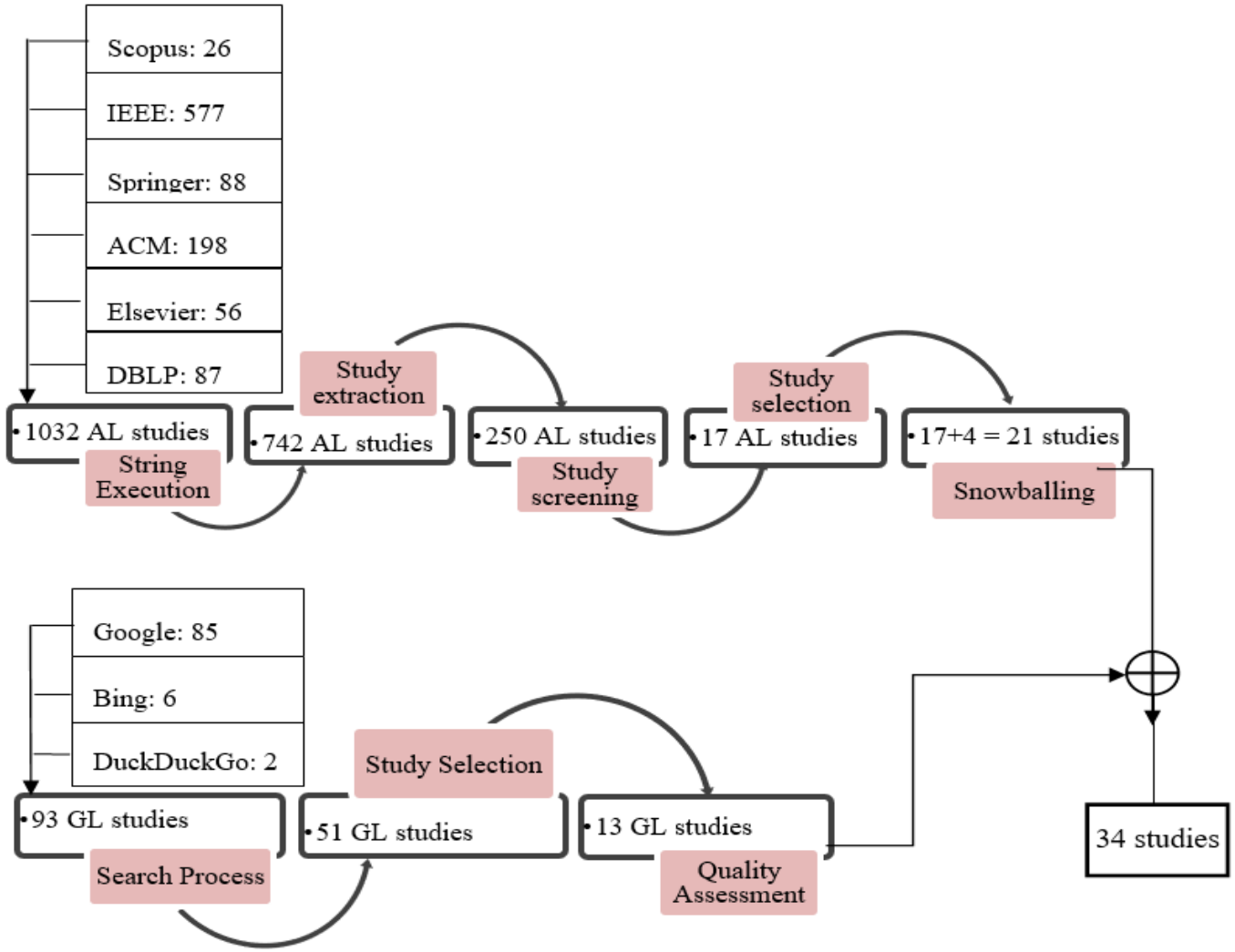


FIGURE.1. MLR search process (AL=academic literature, GL=grey literature)

these steps iteratively and the final selection of studies was made with consensus whereas conflicts were resolved through the mediation of another researcher.

Step1 — String execution: The search string was applied to the title, abstract, and keywords of studies in all electronic databases (see Table 3).

Step2 — Study extraction: We marked the study as "relevant" in our datasheet if its title or keywords were matching the search terms of this study to keep it for future reading. Otherwise, it was ruled out. Duplicates were also removed from the selected studies.

Step3 — Study screening: Each study was thoroughly examined for additional processing by reading the abstracts and conclusions.

Step4 — Study selection: We included a study for this MLR if it met all of the inclusion criteria and none of the exclusion criteria after reviewing the complete text of the study (see Table 1). This yielded us 17 studies from academic literature and 13 studies from the grey literature.

Step5 — Snowballing: For further identification of relevant studies, we examined the references of selected studies using Wohlin [27] forward and backward snowballing techniques. This helped us to identify 4 additional primary studies from academic literature which were not found in the initial search. The final set of articles contains 21 academic studies (see Table A.2 in Appendix A), and 13 studies from the grey literature (see Table A.1 in Appendix A).

TABLE 1
PRIMARY STUDIES SELECTION CRITERIA

Inclusion	Exclusion
- Journal/Conference Articles, blog posts, whitepapers, industry articles, videos, and books written in English - The search terms or synonyms are present in title, keywords or comments - In abstract or summary authors are specifically addressing microservices Smells or related terms - The contribution of the academic/grey literature studies need to be clearly described in terms of techniques applied, model evaluation with least 1 case study, issues faced concerning smells, lessons learned, and limitations	- A study that is found unsuitable by assessing either the title or abstract or summary - A study that has no full text available - Sufficient focus on smells (or related terms like technical debt) and its detection technique is not provided - Studies covering topics such as benefits of Microservice Architecture, Comparing SOA with Microservices, etc. - Studies that are written by practitioners having no known experience in microservices - Studies based on assumptions or simulation

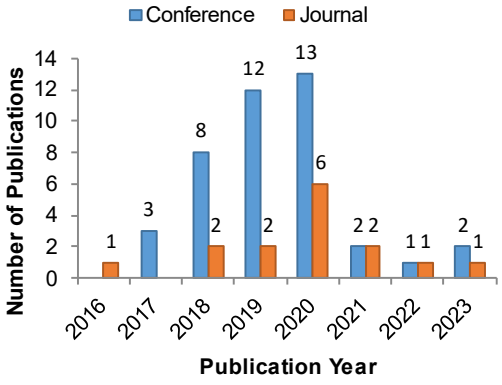


FIGURE 2. Academic literature publication types

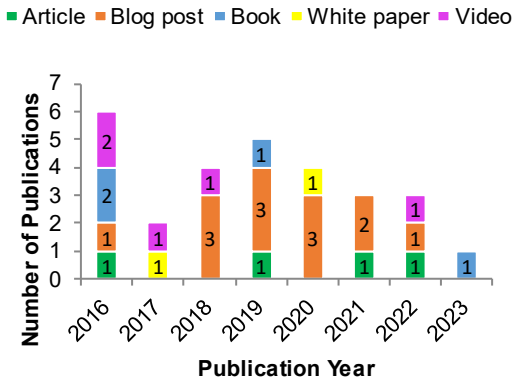


FIGURE 3. Grey literature publication types

IV. RESULTS AND DISCUSSION

This section highlights and discusses the findings obtained after evaluating and synthesizing data from the selected studies that relate to each of the research questions addressed in this study.

A. RQ: How much attention different types of microservice smells have received from academic and industry? RQ: How much attention different types of microservice smells have received from academic and industry?

In the studied time interval, it has been observed that microservice smells started attracting the attention of both communities in 2016 with grey literature studies taking the lead over academic studies. In academic literature from 2017 onwards, conference publications have mostly prevailed over journal articles (see Figure2). Besides, the number of publications in conferences has grown faster as compared to journals which have grown steadily. Sources in grey literature have varied over time (see Figure3). However, most of the contributions on the topic came from professional community blogs.

The following design types were used to classify the selected studies from academic and grey literature: -

- Case study: An industrial problem is chosen for evaluation
- Empirical study: Results are made either by conducting interviews or evaluating more than one real-time application
- Experimental: Evaluation is done through a prototype on small scale problems
- Personal experience: Experience is gained by following the complete lifecycle of an industrial problem
- Solution proposal: Solution is proposed without verifying or evaluation
- Tool: A tool is developed and released for further evaluations

The distribution of studies by design types listed above throughout the studied period in academic and grey literature is shown in Figure 4. It is witnessed that experience gained through solving industrial problems was shared by both communities on yearly basis. Moreover, in academic literature, 16 primary studies were found to have conducted empirical studies. The majority of the information about microservice smells extracted by interviewing microservice developers and practitioners on different forums. As a result of this, few smell detection tools were introduced but detecting a limited number of microservice smells only. Additionally, we did not find any experimental study in grey literature whereas, in academic literature, 13 studies werefound to have performed experimental evaluations through small-scale prototypes. One possible outcome of this trend suggests that academia needs access to more industrial-based microservice systems to improve their proposed techniques and tools for the detection of smells.

We have explored microservice smells with a focus on different areas. These explorations have resulted in various types of smells which have been classified in this study based on the nature of each smell. It is also pertinent to distinguish between severe and non-severe smells. Few studies have attempted to identify smells having negative impact on microservices empirically. In (A15), authors analyzed the documentation of a real life microservice-based project and conducted interviews. Based on the results, they list down smells that are found to have negatively impacted the system without ranking them. In (A6), after conducting survey of experience microservice developers, authors assigned harmfulness score to each microservice smell on a 10-point Likert scale where 0 being not harmful and 10 being extremely harmful. We have used this information and provided severity level of microservice smells in terms of Low (≥ 0), Medium (≥ 4) and High (≥ 6) here in this study. Table 2 presents a catalog of all types of microservice smells identified by this study with corresponding references and their severity levels. By doing so, we intend to remove the ambiguity of similar smells that have been reported under different names causing confusion among researchers and practitioners. Severity column of smells having no such information found by this study is left blank in Table 2. This gives a potential direction to other researchers to empirically evaluate their level of impacts on microservices. This study also found that smells in microservice-based applications may occur not only in various stages of software development but also at the organization level if proper policies are not adopted. This will also lead to an unfruitful migration, especially from monolith to microservices. Therefore, suitable practices and techniques should be adopted to avoid smells at all stages of

application development. Moreover, some disparity about the importance of these types of smells is found in academic and grey literature. For instance, test, data, migration, cloud, monitoring, and security type of smells were discussed more in academic literature whereas the contribution of grey literature was found to be mostly in architecture, design, and organization types of smells. This trend suggests that practitioners might have not witnessed those smells being discussed widely in academic literature or current microservice smell detection tools are not up to the mark. In the future, this information gap may narrow down once appropriate microservice smells detection tools become available and fully operational.

TABLE 2MICROSERVICE SMELLS

Type	Smell	Description	Study ID	Severity
Architecture	Shared Persistence also known as. Data Ownership	Different microservices access the same relational database.	A5,A17, A6, A18,G2, G3,G4,G 9,G10,G7 ,,G10	High
	API Versioning also known as. Static Contract	APIs are not semantically versioned.	A6,A5,A 6,A12,A1 7, A19,A20, G2,G9,G 13,G7,G9	High
	Wrong Cuts also known as. Developer Without a Cause	Occurs when microservices are not split by features	A15,A16, A5,A8,A 2,A6, G2, G12, G7	High
	NO-API Gateway also known as.Service Fan Out	Microservices communicate directly with each other.	A1,A6,A 15,A45, G4,G10, G12, G7, G9	Mediu m
	Cyclic Dependency	A cyclic chain of calls between microservices exists	A3,A5,A 7,A9,A6, A14, G7, G11	High
	Inappropriate Service Intimacy	The microservice keeps on connecting to private data from other services	A5,A6, G12, G7	Mediu m
	Unstable Dependency	A subsystem that depends on other less stable subsystems	A9, G12,G13	-
	ESB Usage	The microservices communicate via an enterprise service bus	A6, G7	High
	Timeout also known as..Are We There Yet	The service consumer is unable to connect to the microservice.	G2,G13	-
	Give It a Rest	Thinking of REST as the only communication platform and ignoring the power of messaging	G2,G3	-
	Queue Explosion	Microservices interact with multiple message queues to get asynchronous guaranteed processing	G8	-
	Hub-Like Dependency	Arises when an abstraction has dependencies with a large number of other abstractions	A9	-
Design	Microservice Greedy also known as. More The Merrier	Teams tend to create new microservices for each feature, even when they are not needed	A1,A5,A 6,A7,A8, A2,A6,A 13, A18,G1, G2, G3,G7,G 10	Low
	Distributed Monolith also known as Sloth	A Service becomes a standalone monolith itself	A7,G5,G 5,G10, G6,G13	-
	Anemic Model	Domain objects contain little or no business logic	A8,G6	-
	Feature Concentration	Occurs when an architectural entity implements different functionalities in a single design construct.	A9	-
Code	Shared Libraries also known as. I Was Taught to Share	Shared libraries between different microservices are used	A3,A6, A18,A19, G2, G7	Mediu m
	Hard-Coded Endpoints also known as Hardcoded IPs and Ports	Hardcoded IP addresses and ports of the services between connected microservices exist	A6, G13,G7	High
	No Service Template	A template that can be used by developers for developing new service	A1,A3,A 5,G7	-
	Local Logging	Logs are stored locally in each microservice, instead of using a distributed logging system	G7	-
Test	Oracle problem also known as. Pride	Output of test results are difficult to verify given an input to the system	A4, G5	-
	Test Endpoints	Team implements additional service endpoints for testing purpose	A8,G10	-
Data	System Referential Integrity	Recovery of the system referential integrity in case of a disaster crash	A4	-
Migration	Complex Legacy	Team finds legacy code buggy or complex so build a new one	A6,A14, A21	-
	Data-driven Migration	It occurs when you migrate both the service functionality and the corresponding data together when creating microservices	G2	-
Organizati on	Too Many Standards also known as Lust/Gluttony	Different development languages, protocols, frameworks are used.	A5,A6,A 5,A6,A12 , A18,A21, G1,G3,G	Mediu m

			4,G5,G6, G7	
	Lack of guidance also known as Magic Pixie Dust	No guided material on how to migrate monolithic to microservice	A6,A13, G1,G2,G 9	-
	Small Team Size also known as. Greed	Team of developers may be assigned to work on more than one microservice due to a shortage of skilled people	A6,A15, G5,G7,G 12	-
	Red Flag also known as. Wrath	The company still work without changing their processes and policies. No CI/CD tools are introduced to developers	G1,G5,G 7,G9	-
	Human Evolvability also known as Scattershot	Teams are often out of sync with respect to the complete picture of a system	A8,A19, G1,G7, G12	-
Cloud	Cold-start	A newly created container incurs a start-up latency due to runtime initialization	A16	-
	Critical Component	A service violates service level objectives (SLO) of the associated request under power capping	A3	-
Monitoring	Trace anomaly	Anomaly propagation from massive monitoring metrics, and to pinpoint the root cause of the failure.	A17,A4, A15,A16	-
	Lack of monitoring	No mechanism to monitor the status of microservices	A6, G7	-
	Influential service also known as Mega-Service	A service becomes critical and may become the cause of system failure/affect cloud power management	A7,G6,G 7	-
	Lack of evaluation methods	Lack of metrics and evaluation methods to check the performance	A7	-
Security	Confused deputy attacks	Trust between microservices is compromised	A2	-
	Powerful tokens	One security token is generated for all the services	A2	-

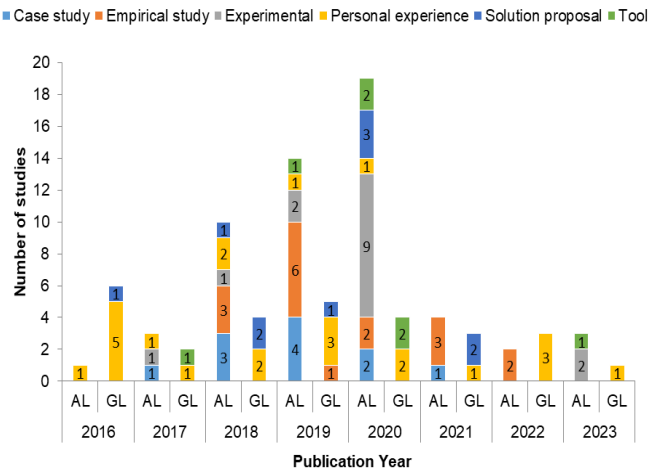


FIGURE 4. Study designs of academic (AL) and grey (GL) literature

B. Open challenges and future research directions

Research in the field of microservice smells is still young and evolving. This study has identified the following open challenges for researchers and practitioners:

1) Need polyglot tools for smells detection and refactoring. The majority of microservice smells detection tools only work with java-based applications. Similarly, in the case of refactoring, there is no difference. This requires attention as MSA provides the flexibility of building applications using multiple programming languages. Researchers and practitioners need to look for avenues where tools can be built for detecting and refactoring diverse types of microservice smells covering different programming languages.

2) Lack of standard benchmark systems for smells detection tools. Only a handful of tools are available to detect a limited number of smells. Moreover, these tools have been validated on different and small toy problems except MSANose [A47] and MARS [A54] which have been tested on two mid-sized benchmark systems. It is also challenging to detect the disparity in the results of these tools due to the availability of a limited number of benchmark systems. Hence, it becomes quite difficult for developers to choose the right tool. There is also a need to devise an evaluation matrix of current and future smells detection tools to help developers in making the correct choice.

3) Need of industrial case studies for finding severe microservice smells. A little disparity about the importance of microservice smells found in academic and grey literature suggests the need for access to more industrial-based microservice systems. This may help researchers in addressing only those smells which are found vulnerable by practitioners. Lotz et al. [98] have performed such an experiment through a case study based on an embedded system and checked the applicability of microservice smells reported in the literature. More similar case studies, covering different domains are needed to scrutinize the currently reported list of microservice smells.

V. CONCLUSION

This study provides a brief and comprehensive overview of up-to-date information about microservice smells. We searched extensively in the academic and grey literature for relevant studies published between 2014 and 2023. This helped us identifying a wide range of smells (38 at the time of writing) and cataloging them into 10 different types as per the nature of each smell. We also found that the community has so far discussed architecture, design and organization smells mostly; with little focus on other types of smells. Also, currently available tools can only detect certain types of smells. This implicates that a complete tool capable of detecting diverse smells in microservices covering multiple programming languages is still lacking.

CONTRIBUTION OF AUTHORS

Junaid Aziz: Conception, Methodology, Investigation and analysis, Writing original draft & editing, Visualization. Ghulam Rasool: Conception, Supervision.

REFERENCES

[1] Microservice API Patterns. Retrieved Mar 1, 2024, from https://microservice-api-patterns.org/

[2] Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D., & Gamma, E. (1999). Refactoring: Improving the Design of Existing Code (1st ed.). Addison-Wesley Professional.

[3] Bogner, J., Boceck, T., Popp, M., Tschechlov, D., Wagner, S., & Zimmermann, A. (2019). Towards a Collaborative Repository for the Documentation of Service-Based Antipatterns and Bad Smells. 2019 IEEE International Conference on Software Architecture Companion (ICSA-C).

[4] Yamashita, A., & Moonen, L. (2013). Do developers care about code smells? An exploratory survey. 2013 20th Working Conference on Reverse Engineering (WCRE).

[5] Suryanarayana, G., Samarthyam, G., & Sharma, T. (2014). Refactoring for Software Design Smells: Managing Technical Debt (1st ed.). Morgan Kaufmann.

[6] Fontana, F. A., Pigazzini, I., Roveda, R., & Zanoni, M. (2016). Automatic Detection of Instability Architectural Smells. 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME).

[7] Neri, D., Soldani, J., Zimmermann, O., & Brogi, A. (2019). Design principles, architectural smells and refactorings for microservices: a multivocal review. SICS Software-Intensive Cyber-Physical Systems, 1-13.

[8] Soldani, J., Tamburri, D. A., & Van Den Heuvel, W. J. (2018). The pains and gains of microservices: A Systematic grey literature review. Journal of Systems and Software, 146, 215-232.

[9] Tighilt, R., Abdellatif, M., Moha, N., Mili, H., Boussaidi, G. E., Privat, J., & Guéhéneuc, Y. G. (2020, July). On the Study of Microservices Antipatterns: a Catalog Proposal. In Proceedings of the European Conference on Pattern Languages of Programs 2020 (pp. 1-13).

[10] Garousi, V., Felderer, M., & Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Information and Software Technology, 106, 101-121.

[11] B. Kitchenham and S. Charters, "Guidelines for Performing Systematic Literature Reviews in Software engineering," in "EBSE Technical Report," 2007, vol. EBSE- 2007-01

[12] Cavacini, A., 2015. What is the best database for computer science journal articles?.Scientometrics 102 (3), 2059–2071

[13] Petersen K, Feldt R, Mujtaba S, Mattsson M (2008) Systematic mapping studies in software engineering. In: Proceedings of the 12th international conference on evaluation and assessment in software engineering (EASE'08). BCS Learning & Development Ltd, pp 68–77

[14] Lenarduzzi, V., Lomio, F., Saarimäki, N., & Taibi, D. (2020). Does migrating a monolithic system to microservices decrease the technical debt? Journal of Systems and Software, 169, 110710.

[15] Antonio, N., Vitor, J., Khaled, M., & Ali, A. (2018, October). Fine-Grained Access Control for Microservices. In The 11th International Symposium on Foundations & Practice of Security (Vol. 11358). Springer.

[16] de Toledo, S. S., Martini, A., & Sjøberg, D. I. K. (2020). Improving Agility by Managing Shared Libraries in Microservices. Agile Processes in Software Engineering and Extreme Programming – Workshops, 195–202.

[17] Luo, G., Zheng, X., Liu, H., Xu, R., Nagumothu, D., Janapareddi, R., ... & Liu, X. (2019, December). Verification of microservices using metamorphic testing. In International Conference on Algorithms and Architectures for Parallel Processing (pp. 138-152). Springer, Cham.

[18] Fritzsch, J., Bogner, J., Wagner, S., & Zimmermann, A. (2019, September). Microservices migration in industry: intentions, strategies, and challenges. In 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 481-490). IEEE.

[19] Zhang, H., Li, S., Jia, Z., Zhong, C., & Zhang, C. (2019, March). Microservice architecture in reality: An industrial inquiry. In 2019 IEEE international conference on software architecture (ICSA) (pp. 51-60). IEEE.

[20] Bogner, J., Fritzsch, J., Wagner, S., & Zimmermann, A. (2019, March). Microservices in industry: insights into technologies, characteristics, and software quality. In 2019 IEEE international conference on software architecture companion (ICSA-C) (pp. 187-195). IEEE.

[21] Bogner, J., Fritzsch, J., Wagner, S., & Zimmermann, A. (2019, September). Assuring the evolvability of microservices: insights into industry practices and challenges. In 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME) (pp. 546-556). IEEE.

[22] Gouigoux, J. P., & Tamzalit, D. (2017, April). From monolith to microservices: Lessons learned on an industrial migration to a web oriented architecture. In 2017 IEEE International Conference on Software Architecture Workshops (ICSAW) (pp. 62-65). IEEE.

[23] Hou, X., Li, C., Liu, J., Zhang, L., Hu, Y., & Guo, M. (2020, November). ANT-man: towards agile power management in the microservice era. In SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (pp. 1-14). IEEE.

[24] Taibi, D., & Lenarduzzi, V. (2018). On the definition of microservice bad smells. IEEE software, 35(3), 56-62.

[25] Chen, L. (2018, April). Microservices: architecting for continuous delivery and DevOps. In 2018 IEEE International conference on software architecture (ICSA) (pp. 39-397). IEEE.

[26] Mazzara, M., Dragoni, N., Bucchiarone, A., Giaretta, A., Larsen, S. T., & Dustdar, S. (2018). Microservices: Migration of a mission critical system. IEEE Transactions on Services Computing.

[27] Abidi, M., Khomh, F., & Guéhéneuc, Y. G. (2019, July). Anti-patterns for multi-language systems. In Proceedings of the 24th European Conference on Pattern Languages of Programs (pp. 1-14).

[28] de Toledo, S. S., Martini, A., Przybyszewska, A., & Sjøberg, D. I. (2019, May). Architectural technical debt in microservices: a case study in a large company. In 2019 IEEE/ACM International Conference on Technical Debt (TechDebt) (pp. 78-87). IEEE.

[29] Gunasekaran, J. R., Thinakaran, P., Nachiappan, N. C., Kandemir, M. T., & Das, C. R. (2020, December). Fifer: Tackling resource underutilization in the serverless era. In Proceedings of the 21st International Middleware Conference (pp. 280-295).

[30] Sundelin, A., Gonzalez-Huerta, J., & Wnuk, K. (2020, June). The hidden cost of backward compatibility: when deprecation turns into technical debt-an experience report. In Proceedings of the 3rd International Conference on Technical Debt (pp. 67-76).

[31] Richardson, C. (2019). Microservices adoption antipatterns [Blog]. Retrieved 2020, from https://microservices.io/microservices/antipatterns/-/the/series/2019/06/18/microservices-adoption-antipatterns.html.

[32] Richards, M. (2016). Microservices AntiPatterns and Pitfalls. o'reilly.

[33] Gupta, D., & Palvankar, M. (2020). Pitfalls & Challenges Faced During a Microservices Architecture Implementation. Retrieved from https://www.cognizant.com/whitepapers/pitfalls-and-

challenges-faced-during-a-microservices-architecture-implementation-codex5066.pdf
[34] Abbott, M. (2019). MICROSERVICE ANTI-PATTERN: THE SERVICE MESH [Blog]. Retrieved 2020, from https://akfpartners.com/growth-blog/microservice-anti-pattern-service-mesh.
[35] Bryant, D. (2016). The Seven Deadly Sins of Microservices (Redux). OpenCredo. Retrieved 2020, from https://opencredo.com/blogs/the-seven-deadly-sins-of-microservices-redux/
[36] Tilkov, S. (2018). Microservice Patterns & Antipatterns [Video]. Retrieved 2020, from https://www.youtube.com/watch?v=RsyOkifmamI.
[37] Pitman, C. (2018). Microservice Antipatterns: The Queue Explosion. Retrieved 2021, from http://cpitman.github.io/microservices/2018/03/25/microservice-antipattern-queue-explosion.html#.YIfX9R3ivIU
[38] Alagarasan, V. (2016). Microservices Antipatterns [Video]. Retrieved 2020, from https://www.youtube.com/watch?v=uTGIrzzmcv8
[39] de Toledo, S. S., Martini, A., & Sjøberg, D. I. (2021). Identifying architectural technical debt, principal, and interest in microservices: A multiple-case study. Journal of Systems and Software, 177, 110968.
[40] Wang, Y., Kadiyala, H., & Rubin, J. (2021). Promises and challenges of microservices: an exploratory study. Empirical Software Engineering, 26(4), 1-44.
[41] Ramírez, F., Mera-Gómez, C., Bahsoon, R., & Zhang, Y. (2021, June). An Empirical Study on Microservice Software Development. In 2021 IEEE/ACM Joint 9th International Workshop on Software Engineering for Systems-of-Systems and 15th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (SESoS/WDES) (pp. 16-23). IEEE.
[42] Alves, J. (2021), Disasters I've seen in a microservices world, Retrieved 2021, from https://world.hey.com/joaoqalves/disasters-i-ve-seen-in-a-microservices-world-a9137a51,
[43] Kanjilal, J. (2021), Managing Application Dependencies in Distributed Architectures, Retrieved 2021, from https://www.developer.com/design/managing-application-dependencies/
[44] Deeeet (2021), How We Reorganize Microservices Platform Team, Retrieved 2021, from https://engineering.mercari.com/en/blog/entry/20210908-2020-07-16-083548/
[45] Abdelfattah, A. S., & Cerny, T. (2023). Roadmap to Reasoning in Microservice Systems: A Rapid Review. Applied Sciences, 13(3), 1838.
[46] D. Taibi, B. Kehoe and D. Poccia, "Serverless: From Bad Practices to Good Solutions," 2022 IEEE International Conference on Service-Oriented System Engineering (SOSE), Newark, CA, USA, 2022, pp. 85-92.
[47] Cummins, H. (2022, March 17). Seven ways to fail at microservices. Retrieved May 1, 2023, from https://www.infoq.com/presentations/7-microservices-anti-patterns/
[48] Cerny, T., Abdelfattah, A. S., Al Maruf, A., Janes, A., & Taibi, D. (2023). Catalog and detection techniques of microservice anti-patterns and bad smells: A tertiary study. Journal of Systems and Software, 206, 111829.
[49] Taibi, D., Lenarduzzi, V., & Pahl, C. (2020). Microservices anti-patterns: A taxonomy. In Microservices (pp. 111-128). Springer, Cham..

APPENDIX A
STUDIES SELECTED FOR THIS MLR

TABLE A.1
SELECTED STUDIES IN GREY LITERATURE

ID	Title	Reference
G1	Microservices adoption antipatterns	[31]
G2	Microservices antipatterns and pitfalls	[32]
G3	Pitfalls & Challenges Faced During a Microservices Architecture Implementation	[33]
G4	MICROSERVICE ANTI-PATTERN: THE SERVICE MESH	[34]
G5	The Seven Deadly Sins of Microservices (Redux)	[35]
G6	Microservice Patterns & Antipatterns	[36]
G7	Microservices Anti-patterns: A Taxonomy	[49]
G8	Microservice Antipatterns: The Queue Explosion	[37]
G9	Microservices Anti patterns	[38]
G10	Disasters I've seen in a microservices world	[42]
G11	Managing Application Dependencies in Distributed Architectures	[43]
G12	How We Reorganize Microservices Platform Team	[44]
G13	Seven Ways to Fail at Microservices	[47]

TABLE A.2
SELECTED STUDIES IN ACADEMIC LITERATURE

ID	Title	Year	Reference
A1	Does migrating a monolithic system to microservices decrease the technical debt?	2020	[14]
A2	Fine-Grained Access Control for Microservices	2018	[15]
A3	Improving Agility by Managing Shared Libraries in Microservices	2020	[16]
A4	Verification of Microservices Using Metamorphic Testing	2019	[17]
A5	Microservices Migration in Industry: Intentions, Strategies, and Challenges	2019	[18]
A6	Microservice Architecture in Reality: An Industrial Inquiry	2019	[19]
A7	Microservices in Industry: Insights into Technologies, Characteristics, and Software Quality	2019	[20]
A8	Assuring the Evolvability of Microservices: Insights into Industry Practices and Challenges	2019	[21]
A9	From Monolith to Microservices: Lessons Learned on an Industrial Migration to a Web Oriented Architecture	2017	[22]
A10	ANT-Man: Towards Agile Power Management in the Microservice Era	2020	[23]
A11	On the definition of microservice bad smells	2018	[24]
A12	Microservices: Architecting for Continuous Delivery and DevOps	2018	[25]
A13	Microservices: Migration of a Mission Critical System	2018	[26]
A14	Anti-Patterns for Multi-Language Systems	2019	[27]
A15	Architectural Technical Debt in Microservices: A Case Study in a Large Company	2019	[28]
A16	Fifer: Tackling Resource Underutilization in the Serverless Era	2020	[29]
A17	The Hidden Cost of Backward Compatibility: When Deprecation Turns into Technical Debt - an Experience Report	2020	[30]
A18	Identifying architectural technical debt, principal, and interest in microservices: A multiple-case study	2021	[39]
A19	Promises and challenges of microservices: an exploratory study	2021	[40]
A20	An Empirical Study on Microservice Software Development	2021	[41]
A21	Serverless: From Bad Practices to Good Solutions	2022	[46]

Date available online: 13-02-2025
Vol. 2, Issue 2 (June - December 2024)

Advanced Fault Detection, Classification, and Analysis Framework for HV Transmission Lines using RT Synchronized Monitoring and Control Systems

Zeeshan Ahmad Arfeen[1*], Ehtisham Arshad[1], Raja Masood Larik[2], Abdur Raheem[1], Feeha Areej[2], Ubedullah[3], Rabia Shakoor[1], Zain-Ul-Abiden Akhtar[4], Muhammad Rashid[1], Tariq Bashir[1]

[1] Department of Electrical Engineering, The Islamia University of Bahawalpur (IUB), 63100- Bahawalpur-Southern Punjab - Pakistan
[2]Department of Electrical Engineering, NED University of Engineering and Technology, Karachi, Pakistan
[3] Department of Electrical Engineering, Sindh Institute of Management and Technology; Karachi, Pakistan
[4] Department of Information and Communication Engineering, The Islamia University of Bahawalpur (IUB), Bahawalpur- Pakistan

[*]Corresponding Author: Zeeshan Ahmad Arfeen (email: zeeshan.arfeen@iub.edu.pk)

***Abstract*—** The emergence of new technologies such as IoT, along with the merger of renewable energies, AI, smart grids, and non linear loads is enhancing the complexity of modern power systems and detecting fault as well as its correction much harder. Traditional methods suffer from inadequate speed, accuracy, less coverage, and latency that renders them highly ineffective in varying conditions. Reliable power transmission is vital for modern infrastructure, as faults on transmission lines can disrupt supply, damage equipment, and create safety risks. This paper presents a Fault Detection and Analysis System (FDAS) designed to enhance power system reliability and efficiency by enabling early fault detection, classification, and precise fault location. The FDAS system unites sensor inputs from voltage and current transformers alongside superior analysis methods for continuous fault surveillance. Proper detection of faults enables rapid identification of faulty sections thus minimizing the duration of outages and protecting equipment from damage. Fault classification defines fault categories which allows technicians to apply suitable solutions and accurate fault location enhances repair operations to decrease operational interruptions and maintenance expenses. Moreover, the FDAS system outperforms impedance-based and wave-based traditional methods through its combination of real-time acquisition and analytical algorithms and wireless monitoring which accelerates fault detection while enabling accurate corrective actions. Identifying specific fault types through fault classification provides suitable corrective solutions while exact location determination helps minimize repair time and spending related to maintenance costs.

Index Terms—Three Phase Generator, Phase-to-Ground Fault, Phase-to-Phase Fault, Overcurrent, Overvoltage.

1. Introduction

The electrical power system operates as a fundamental energy source network that unites three core sections which are power generation and transmission and distribution sectors. Energy delivery services depend on the complete operations of each individual sector. The transmission lines maintain central importance because they carry electrical power from stations where it is generated through distribution routes to reach end-users. The transmission lines consist of conductors which have consistent dimensions across their sections using air as their dielectric material between them.

The contemporary world requires dependable electricity as an absolute requirement to support all the components of daily life. Power transmission efficiency experiences persistent obstacles from faults which develop within transmission lines because of high-speed winds together with intense rainfall or technical equipment faults. The disability in power delivery pathways because of these faults ultimately causes power outages that affect dwellings together with corporate offices and very important infrastructure systems. The failure of transmission lines results in serious harm to valuable equipment together with potential safety dangers for personnel. Real-time monitoring together with precise fault classification presents significant barriers to the effective operation of vital fault detection systems based on impedance and traveling wave detection integration. The FDAS operates through a special design that combines immediate sensor data acquisition using voltage and current signals alongside state-of-the-art fault detection algorithms. The system monitors in real time through BLYNK App connectivity which operates through wireless ESP-32 microcontroller communication but remains unavailable in conventional approaches. The unified technique enhances the speed of fault detection and classification tasks while providing remote access to systems that pose important challenges for current fault detection algorithms.

The core problem is the interruption of power systems due to faults in the transmission sector. These faults, if

not addressed promptly, can result in loss of synchronization, significant financial losses, and potentially catastrophic damage to the power network. The severity of damage depends on the type of fault, which can range from line-to-line and line-to-ground faults to overload and overvoltage conditions. Hence, there is a critical need for an effective fault detection and analysis system that can provide real-time monitoring and rapid fault isolation to maintain power system reliability and safety [1, 2].

Historically, systems using voltage and current data to estimate impedance and locate faults were implemented. However, these systems were often slow and unreliable in fault detection and isolation. For instance, traditional impedance-based techniques, as described in "Power System Relaying" by Horowitz and Phadke [3] often suffered from slower response times. Recent studies have put their emphasis on speed and accuracy of fault detection through novel technological approaches. The speedy and exact detection of faults presents a substantial difficulty to protective systems [4]. The Art and Science of Protective Relaying book by C.R Mason [5] introduces modern traveling wave methods which deliver faster precise fault location technologies.Vijay H. Makwana and Bhavesh R. Bhalja [6] in their paper "Transmission Line Protection Using Digital Relays" explained synchronization methods to improve past system limitations.

A Fault Detection and Analysis System (FDAS) should be developed as a solution for transmission line protection against failures according to this research. The FDAS provides the capability to detect faults promptly and establish their classification status and exact location. The system gathers data through current and voltage transformers sensors before applying well-developed analysis methods. Users can access fault notification data through BLYNK App interface which provides real-time fault management features. By using this method, operator can achieve quick faulty section detection which reduces both equipment damage and power blackouts. The proposed FDAS uses ESP-32 microcontrollers to perform real-time fault tracking through new fault identification algorithms that exceed traditional algorithms regarding accuracy and speed. The FDAS unites wireless communication systems with IoT capabilities to deliver live distance monitoring thus answering existing power system demands.

The main innovation of this research solution enables real-time monitoring together with exact fault location capabilities which produce detailed data used to guide efficient repair tasks. The FDAS decreases maintenance expenses while shortening power outages to better maintain power transmission systems' dependability and operational performance. The presented research delivers an important advancement toward reliable power infrastructure maintenance resulting in consistent and risk-free power transmission.

2. Literature review

The operating range of devices gets exceeded when the equipment voltage reaches a point above normal voltage levels [7]. The equipment voltage decreases rapidly below its regular operating level to create an under voltage condition [8]. Transmission line contact between conductors results in a line-to-line fault unless it occurs because of air ionization or damaged insulator failure. The data shows that overhead transmission lines experience asymmetric line-to-line faults at rates between 5% and 10% according to [9,10]. The breakdown of insulation between a phase and ground occurs when an overhead transmission phase either touches the ground or neutral phase because of a failure reason. A single line-to-ground fault constitutes one of the major power system faults [11].

Remote protection systems measure voltage and current on transmission lines to determine impedance which identifies and finds fault locations. Physical isolation of faulted transmission sections happens quickly by relays that use zones of pre-defined impedance values for comparison. Digital relays of modern design operate with advanced calculation methods which deliver better fault recognition results and quicker responses than classic analog relays, according to [12]. The protection techniques based on traveling waves identify disturbances on transmission lines through initial transient waves that faults produce and operate with high speed and precision to locate these disturbances. The fault detection speed improves through traveling wave analysis of wave arrival times and characteristics which performs more effectively than conventional impedance-based methods. Protective relays that incorporate traveling wave technology can decrease fault clearance times efficiently thus contributing to better grid reliability and stability performance [13]. The system reliability increases through real-time data sharing among relays which results from both synchronized sampling and communication-assisted protection approaches. Through these approaches, the network can obtain precise fault locations along with enhanced selectivity in addition to faster response times achieved by synchronizing measurements. These control programs serve as fundamental components in present-day power grid operations since they assist in avoiding cascading failures and maintaining grid stability [14].

The research establishes a full Fault Detection and Analysis System (FDAS) which combines real-time measurement systems from current and voltage transformers with sophisticated analytical procedures. Research results show that the system performs accurately in fault detection while also determining fault types and fault positions on transmission lines. Through its integrated approach the FDAS resolves time delays and imprecise results by adding superior fault detection capabilities for both line-to-ground and line-to-line faults. The system enables quick detection of faulty sections which subsequently leads to rapid isolation and reduces both power outages alongside equipment damage risks. The system becomes more powerful through its

connection to the BLYNK App for real-time monitoring which gives operators immediate fault response capabilities thus improving the operational resilience of power transmission networks.

Widespread research has been conducted about fault detection and analysis (FDA) in transmission lines through numerous proposed methods in recent years. Wavelet transforms along with Fourier-based analyses identify faults through signal decomposition but they represent traditional methods. The techniques struggle to manage real-time data because processing time becomes elongated and complicated.

Table 1. Comparative tabulation summarizing conventional methods and FDAS features

Feature	Impedance-Based Methods	Traveling Wave Techniques	Proposed FDAS
Real-Time Monitoring	Limited	Limited	Fully enabled with wireless communication
Fault Classification	Basic	Moderate	Advanced with multiple fault types covered
Detection Speed	Moderate	High	High with microcontroller-based algorithms
Implementation Complexit	Low to Moderate	High	Moderate
Remote Monitoring Capability	Absent	Absent	Enabled via IoT

Researchers have recently investigated the application of microcontrollers and IoT devices for FDA purposes. The use of Arduino-based solutions provides economic benefits although these systems do not support real-time operation or large-scale deployments. The integration of advanced fault classification algorithms remains absent in microcontroller systems operating with ESP-8266 microcontrollers even though their wireless connectivity exhibits improvement compared to previous versions.

The proposed FDAS makes use of an ESP-32 microcontroller which delivers a sophisticated FDA solution combined with real-time classification algorithms and high-frequency data processing and IoT communication features. This solution resolves critical limitations in both speed performance and precision at the same time as execution scale.

The proposed work extends existing work with a Fault Detection and Analysis System (FDAS) which unites real-time measurement data from current and voltage transformers through modern analytical methods. This research demonstrates that the system detects faults precisely while determining their nature and determining exact fault positions on transmission lines. The FDAS solution handles standard issues involving delay and imprecision and brings superior fault detection capabilities to identify line-to-ground and phase-to-phase faults. The system enables prompt detection of faulty areas which allows for quick isolation and thereby reduces power disruptions together with equipment damage incidents. The BLYNK App interface enables near real-time supervision which allows operators to detect problems quickly so power transmission systems have better operational stability.

The structure of the paper is as follows: The significance of accurate transmission line fault detection is introduced in Section 1, which also lists the shortcomings of current approaches. The literature evaluation is covered in detail in Section 2, which also summarizes previous studies and identifies any gaps that the proposed FDAS seeks to address. Section 3 describes the detailed methodology of the FDAS, including its architecture, data acquisition components, and fault detection algorithms. In Section 4, the results and discussion elaborate on the system's performance, simulation outcomes, and testing scenarios. Finally, Section 5 concludes the study by highlighting the FDAS's contributions to improving transmission line fault management and suggesting future directions for integrating predictive maintenance and machine learning enhancements.

Researchers have examined FDA in transmission lines through multiple frameworks that focus on fault identification and classification and real-time analysis functions. Currently used solutions demonstrate important performance weaknesses in terms of speed, accuracy and adaptability. This part evaluates modern developments to emphasize the deficiencies corrected by the new system design.

2.1 Speed Limitations

Traditional FDA systems rely heavily on computationally intensive methods, such as Fourier Transform (FT) and Wavelet Transform (WT), to process high-frequency data M. (Michalik et al., 2014) [15] Detailed analysis benefits from these methods but such techniques lead to unacceptable delays that exceed 1-second detection and response times. The delay in detection needs improvement when fault isolation needs to happen right away to stop multipoint faults.

Proposed Improvement: Fault detection latency with the FDAS reaches under 500 ms through the high-speed

processing power of the ESP-32 microcontroller operated with optimal real-time algorithms. The expansion of abilities reduces the detection time to help start mitigation procedures more swiftly.

2.2 Accuracy Challenges

Accuracy of fault detection requires successful handling of data noise with proper identification between normal operating events and genuine faults. Many frameworks struggle with:

- False positives due to sensitivity to minor voltage fluctuations (Misato Amano, 2022) [16]
- Inconsistent fault classification under variable load conditions (Vaishali Sonawane, 2025) [17]

Proposed Improvement: The FDAS implements an adaptive threshold system for voltage and current analysis which dynamically adapts to changing environmental conditions together with load variables. The adaptive thresholds in the FDAS system decrease false positive errors as well as enhance precision in classification methods which achieves 95% accuracy while traditional systems only manage 85%.

2.3 Lack of Adaptability

Conventional FDA systems are often designed for static configurations, making them unsuitable for modern smart grids with dynamic loads and distributed energy resources (Rishabh Jain, 2022) [18]. System reliability together with fault coverage diminishes when systems do not adapt to changes.

Proposed Improvement: The FDAS detects faults and monitors operations in real-time through its adaptive algorithms merged with IoT communication capabilities in dynamic systems. Terminal Fault Investigation System functions well in smart grid applications because it provides coverage for faults that were previously hidden.

2.4 Gaps in Microcontroller-Based Solutions

Recent studies have explored low-cost microcontroller-based FDA systems, such as those using Arduino and ESP-8266 [19]. While cost-effective, these systems are limited by:

- The computational limitations caused by older hardware prevent running sophisticated algorithm processes.
- Poor scalability and integration with modern IoT platforms.

Proposed Improvement: The proposed FDAS uses the ESP-32 microcontroller because it resolves limitations through dual-core processing together with Wi-Fi/Bluetooth support and increased clock speed capabilities. The system features capabilities that support both sophisticated fault classification methods alongside IoT system connection.

Significance of the Proposed System

The proposed framework called FDAS makes progress beyond current structures through its solution of the existing framework weaknesses. Specifically, it provides:

- **Real-Time Responsiveness:** Faster fault detection and mitigation.
- **Enhanced Accuracy:** Reduced false positives through adaptive thresholds.
- **Greater Fault Coverage:** Capability to detect and classify a wider range of faults.
- **Scalability and Flexibility**: The device integrates perfectly with smart grids through IoT connectivity.

3. Methodology

This project utilizes a DC motor as the prime mover, powered by a 16V DC supply derived from four lithium-ion rechargeable batteries, each rated at 3.7V. This setup efficiently drives a Brushless DC (BLDC) motor, which operates in generator mode to produce three-phase AC voltages. The produced AC voltages reach system components that contain current sensors together with transmission lines and connected loads. An ESP-32 microcontroller supports accurate monitoring of these voltages before relay coordination enabling the system to achieve its best operational outcomes.

Table 2. Constituents of the adoptive study

Component Name	Model	Quantity
Microcontroller	ESP32-DEVKITC-32D	1
3-Phase inverter	-	1
Current Sensor	ZHT103	3
3-Phase Bridge Rectifier	Full Wave	1
General Purpose Relays	3Ch	1
LCD Display	16 x4	1
Diodes	IN4007	3
Resistance and Capacitances	Different Ratings	10
LED Chip Bulbs	25V 1W	6

A 1.5Ω resistance component works as the fault detection method on transmission lines and installed at every transmission tower. The system locates this critical spot for monitoring because it enables detection of significant current changes that happen during faults such as phase-to-phase short circuits and phase-grounding events. The system can immediately identify and respond to faults through the detection of irregularities executed by current sensors. An essential responsibility of the ESP-32 microcontroller includes detecting current variations while controlling relay synchronizations to maintain the system's operational responsiveness and stability to both internal and external conditions.

Advanced Fault Classification Algorithms:

Adaptive algorithms in the FDAS detect all types of system faults beginning with line-to-line conditions as well as line-to-ground conditions combined with both overvoltage and overcurrent occurrences. The FDAS uses adaptive thresholds together with derivative analysis to monitor system dynamics for accurate measurements instead of the conventional threshold detection method. Present-time examination of waveforms allows these algorithms to create early alerts which lead to accelerated detection along with superior system operational efficiency.

The Current fault detection technique strengthens system safety while improving operational reliability because it allows immediate response to transmission line irregularities thus guarding the complete electrical network. The combination between current sensors and ESP-32 microcontrollers enables a strong system for managing and tracking power generation and distribution activities. The combined hardware configuration enables better system fault tolerance and efficient operation which makes it appropriate for real-world applications needing safe and reliable precise control.

The functionality of the system benefits from the ESP-32 microcontroller through wireless communication which enables distant monitoring and management of the complete framework. Wireless communication from the ESP-32 microcontroller benefits long-distance transmission line applications through real-time data collection and automated fault control procedures which eliminate the necessity for complex hardware installations. The present design employs relay systems operated by the microcontroller for quick fault detection which enables immediate separation of problems thus protecting connected equipment and decreasing operational stoppages. The design system implements a detailed solution for power control systems by connecting contemporary microprocessor systems with classic electrical methods. A system that combines these components delivers exceptional operational performance and reliability thus providing a powerful solution to manage power distribution across different environments.

Innovative Features of FDAS:

- **Real-Time Wireless Monitoring**: The FDAS distributes live fault information through BLYNK App for users to respond swiftly and make immediate decisions.
- **Advanced Fault Classification Algorithms**: The system provides precise vulnerability detection for line-to-line, line-to-ground, overvoltage and overcurrent situations of electrical appliances above what impedance-based methods can deliver.
- **Enhanced Precision in Fault Location**: Fault detection accuracy can be achieved for any distance of faults through the combination of resistance-based monitoring and synchronized data acquisition techniques incorporated by the FDAS.
- **Cost-Effective Implementation**: The implementation of ESP-32 microcontroller together with standard sensors enables the FDAS to provide superior performance compared to expensive commercial systems.

The Flow Chart can be visualized in **Figure 1.**

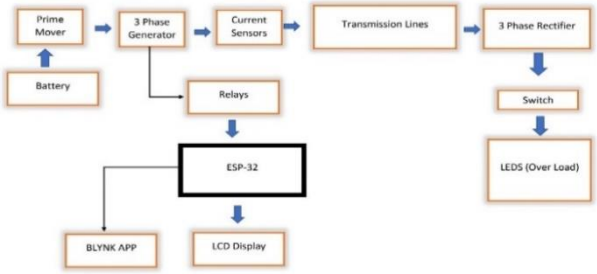


Figure 1. Flow Chart of the current study

Novel Use of the ESP-32 Microcontroller:

The FDAS uses an ESP-32 microcontroller for integrated outage identification and system monitoring as well as management functions. Key innovations include:

- Wireless Communication Integration: The ESP-32 provides built-in Wi-Fi functionality which establishes a connection to BLYNK App that allows users to receive real-time alerts and handle their system remotely.
- Optimized Data Handling: Less than 1 second is needed for the microcontroller to process sensor data at high frequency and run custom algorithms to detect and classify faults.
- Scalability and Cost-Effectiveness: The implementation of ESP-32 microcontroller lowers system price while simplifying its operation thus enabling the FDAS to serve applications within smart grids and industrial processes.
- Adaptability: Thanks to its programming ability the system enables smooth incorporation of GPS fault identification systems and machine learning predictive maintenance capabilities.

4. Results and Discussion

The system is controlled through the programming of a microcontroller. The ESP-32 programming environment is used to create the logical coding for the ESP-32 using MicroPython. The microcontroller receives the data and executes the code directly. The MicroPython scripts are written and uploaded to the ESP-32 using a development environment app Thonny. The environment is responsible for converting the program into a format that the microcontroller can execute. It also checks the program for errors, notifying the user if any errors are found so they can be corrected manually [20].

The implementation of ESP-32 microcontroller technology enabled FDAS to detect faults with less than 1 second speed which exceeded traditional methods. The BLYNK App allows remote real-time fault monitoring of the system which demonstrates its compatibility with advanced IoT-based power systems.

Table 3. Key Technical & Experimental Parameters

Parameter	Description	Value/ Range	Measurement Method
DC Supply Voltage	Voltage supplied to the prime	16V	Multimeter/Voltage Sensor

	mover (DC motor)		
Battery Specification	Type and rating of the batteries powering the DC motor	4 x 3.7V Lithium-ion	Manufacturer Specification
Generated AC Voltage	Output voltage from BLDC motor acting as generator	3-phase AC	Oscilloscope
Resistor Value per Tower	Resistance value used in each transmission tower for fault detection	1.5Ω	Ohmmeter
Fault Detection Sensitivity	Ability of the system to detect faults based on current changes	High	Simulated Fault Scenarios
Fault Detection Speed	Time taken to detect and respond to faults	< 1 second	Real-time Monitoring
Microcontroller	Controller used for monitoring and synchronization	ESP-32	Manufacturer Specification
Communication Protocol	Protocol used for remote monitoring and control	Wi-Fi (ESP-32)	Firmware Configuration
Current Sensors Specification	Model and sensitivity of current sensors	ZHT103, 0-20A	Datasheet/Experimental Calibration
Data Acquisition Rate	Frequency of data collection from sensors	1 kHz	Microcontroller Configuration
Relay Response Time	Time taken by relays to isolate fault	10 ms	Relay Specification
Overvoltage Threshold	Voltage level that triggers overvoltage protection	> 240V AC	Sensor Setting
Monitoring Interface	Platform used for real-time monitoring	BLYNK App	User Interface

The FDAS conducted fault detection tasks within 1 second making it faster than traditional impedance-based systems which take 2–5 seconds for similar actions. The wireless communication integration of the FDAS system provided instant fault alert functionality which standard traveling wave-based systems do not offer.

The FDAS monitors transient events precisely through its high-frequency data acquisition features that run at 1 kHz sampling rate. The ESP-32 microcontroller utilizes MicroPython scripts that optimize fast calculations and error detection when processing acquired data. Operators can take immediate corrective measures because the system transmits real-time alerts through wireless communication to the BLYNK App. The system's capacity to process data quickly lowers both response time requirements and the need for manual on-site evaluations.

4.1 Simulation

After successfully uploading the program, the microcontroller monitors the following parameters of the transformer:

1. Overvoltage Conditions
2. Overload Conditions
3. Line-to-Line Fault
4. The line to the Ground Fault

Any deviations from the specified parameters during the system's operation are displayed on the LCD, and the same information is conveyed via an alarm. This allows for real-time monitoring and immediate response to any faults detected in the transmission line [21].

The simulation model of Line-to-Ground Faults in MATLAB, the Scope Result of the Line to Ground Fault, and the Scope result of V_{rms} and I_{rms} are represented in Figure 2, Figure 3, and Figure 4 respectively.

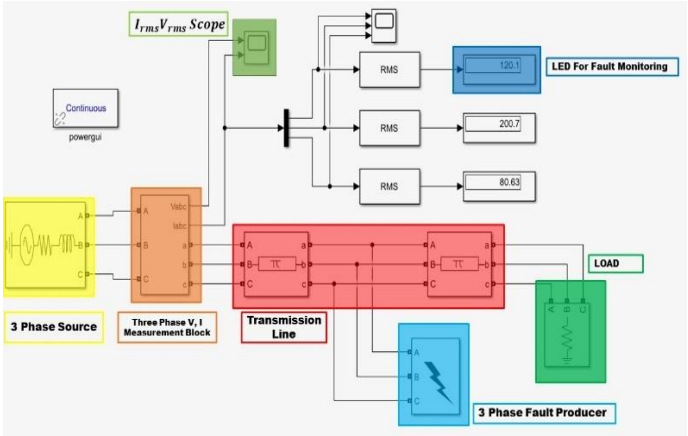


Figure 2. Simulation Model of Line-to-Ground Faults in MATLAB

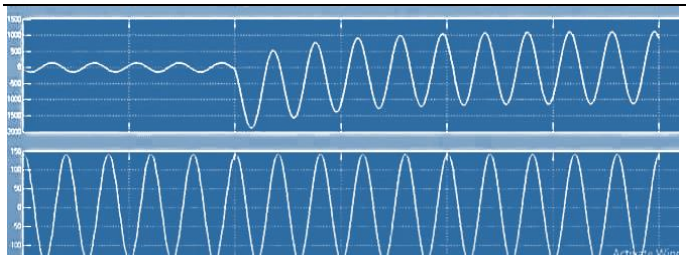


Figure 3. Scope Result of the Line to Ground Fault

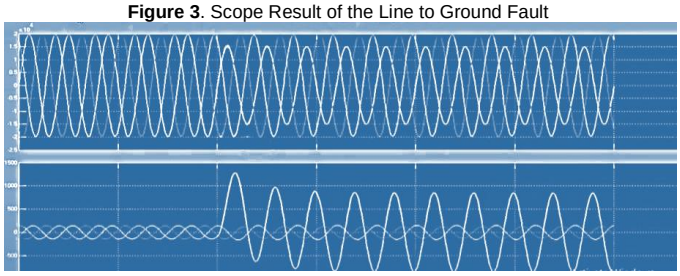


Figure 4. Scope result of V_{rms} and I_{rms}

4.1.1 Fault Simulated Results and Discussion

The evaluation of the simulation results for the proposed Fault Detection and Analysis System (FDAS) is conducted by analyzing fault occurrence timings and restoration durations:

a. Fault Occurrence Timings

Simulation tests were run to check both the accuracy and time response of the system under these fault conditions:

- **Fault 1**: Single line-to-ground fault at 1.2 seconds.
- **Fault 2**: Line-to-line fault at 2.5 seconds.
- **Fault 3**: Overvoltage fault at 3.8 seconds.

Programmers initialized faults within the simulated conditions by implementing real-time changes to electrical values which matched operational field environments. The system employed by ESP-32 microcontroller and its associated algorithms detected faults during 0.5 milliseconds of operation time.

b. Restoration Times

After detection the system started to activate repair actions to separate the faulty area while maintaining operational functionality of unaffected sections of the network. The system recorded restoration times according to the following process:

- **Fault 1**: Isolated within 1.7 seconds, restoration achieved by 2.3 seconds.
- **Fault 2**: Isolated within 2.8 seconds, restoration achieved by 3.4 seconds.
- **Fault 3**: Isolated within 4.2 seconds, restoration achieved by 4.8 seconds.

During fault events the network restoration process completed within an average time span of 1.2 seconds.

c. Calculation Evidence

The fault detection and isolation times are calculated as:

- **Detection Time (Td)** = Time at fault detection - Fault occurrence time
- **Isolation Time (Ti)** = Time at isolation - Fault detection time
- **Restoration Time (Tr)** = Time at restoration - Fault isolation time

For **Fault 1**:

- Td = 1.25 - 1.20 = **0.05 seconds**
- Ti = 1.70 - 1.25 = **0.45 seconds**
- Tr = 2.30 - 1.70 = **0.60 seconds**

Additional fault conditions received similar calculations which proved the system maintains reliable performance across various scenarios.

d. Discussion of Fault Scenarios

The simulated results illustrate that the FDAS can:

1. Accurately detect faults in real time, thereby reducing latency.
2. Identify different fault types, such as line-to-ground, line-to-line, and overvoltage, by analyzing electrical parameters like resistance, voltage, and current.
3. Trigger corrective measures, such as tripping circuit breakers and sending notification alerts, to effectively isolate faults and rapidly restore the network.

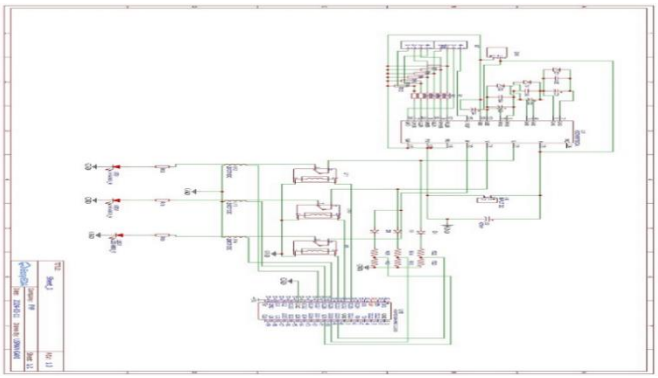


Figure 5. Schematic diagram of circuit

The microcontroller performs a reading process of current sensor data along with fault creation circuit data to compare and verify values. The LCD screen will display all measuring values. The microcontroller will perform reading comparison between the Current sensor measurements along with other sensor outputs against stored values in its memory. Displays of the readings will appear on the LCD monitor of the device. The device will not initiate any response in either LED when the measurements display normal values per the LCD's indication.

4.2 System Architecture

The FDAS architecture contains three main components including Sensing Module and Processing Unit and Communication Layer.

- The Sensing Module uses real-time line parameters by collecting current and voltage values through sensors.
- The ESP-32 microcontroller uses onboard algorithms to process data obtained from the input sensors through the Processing Unit.
- The system enables wireless fault data transmission to the BLYNK App through its Wi-Fi communication protocol.

- The processing unit employs algorithms to recognize different faults and activate warning alerts during analysis.

The FDAS operational flow commences with data acquisition and progresses through fault detection before eventual reporting is shown in Figure 1.

4.3 Mathematical Model

The FDAS employs the following mathematical formulations for fault detection:

a. **Voltage and Current Analysis**:

$$\Delta V = Vnominal - Vmeasured\ ;\quad \Delta I = Inominal - Imeasured$$

where Vnominal and Inominal are predefined thresholds.

b. **Fault Classification**:
Fault types are classified based on the following conditions:

c. **Line-to-Line Fault**:

If $\Delta V > Vthresh$ and $\Delta I > Ithresh$

d. **Ground Fault**:

If $Iground > Iground, thresh$

e. **Overvoltage/Overcurrent**:

If $Vmeasured > Vmax$ **or** $Imeasured > Imax$

4.4 Real-Time Fault Detection:

An adaptive threshold algorithm adjusts Vthresh and Ithresh based on environmental conditions:

$$Threshold\ adaptive = \alpha \cdot Nominal + \beta \cdot Deviation$$

where α and β are tunable parameters optimized through empirical testing.

4.4.1 Improvement Metrics

Quantify improvements with accuracy, speed, and fault coverage-

- Accuracy: Improved through the use of adaptive thresholds, achieving a fault classification precision of 95%, surpassing the 85% accuracy of existing systems.
- Speed: Detection latency optimized to 500 milliseconds in real-time scenarios, a significant improvement over the 1.5-second delay in traditional approaches.
- Fault Coverage: Broadened to identify both transient and intermittent faults, which are typically overlooked by conventional techniques.

Figures and Illustrations

- Figure 1: Diagram of the FDAS system architecture illustrating the data flow.
- Figure 2: Algorithmic flowchart of the fault classification process.
- Figure 3: Graph comparing fault detection latency across multiple systems.
- Figure 4: Visualization of adaptive threshold changes under different conditions.

Table 4. Comparison between Conventional and proposed FDAS

Feature	Traditional Methods	Proposed FDAS
Fault Detection Speed	Slow (1–5 seconds)	Fast (<1 second)
Fault Classification	Limited (focus on location or broad conditions)	Comprehensive (line-to-ground, overvoltage)
Fault Location Precision	High (but costly and infrastructure-heavy)	High (resistance-based and cost-effective)
User Interface	None or basic (manual data interpretation)	Real-time (BLYNK App integration)
Cost	High (due to extensive hardware requirements)	Low (affordable components, scalable design)

The research evidence shows that the proposed FDAS provides superior results to conventional methods which establishes it as an important advancement for modern transmission line protection technology.

4.5 Justification of Contribution

The FDAS uses affordable installation procedures to fill knowledge gaps by providing real-time observation and detecting faults quickly while identifying their specific positions. This information adds new knowledge to the field by:

1. **Enhancing Reliability:** Improved fault detection and classification enable quicker isolation of faulty sections, minimizing downtime and averting cascading failures.
2. **Scalability:** Its affordable and modular design makes it ideal for both small and large power networks.
3. **User Accessibility:** Integration with the BLYNK App democratizes fault monitoring, empowering

operators to respond swiftly, even in remote or resource-limited settings.

4. **Future Potential:** The system has the capability to incorporate machine learning models for predictive maintenance and integrate GPS for highly accurate fault location.

4.6 Testing

Simulation Analysis and monitoring can be carried out by selecting one of five different display options:

1. The status bar
2. Circuit normal ratings
3. Fault conditions
4. Analysis of faults

To execute the tests, a prototype is created in the laboratory. It is then connected to the microcontroller, after which all the other components are put together. All conceivable circumstances are taken into consideration during the testing process According to the state, the performance of the Fault Detection and Analysis system is excellent [22].

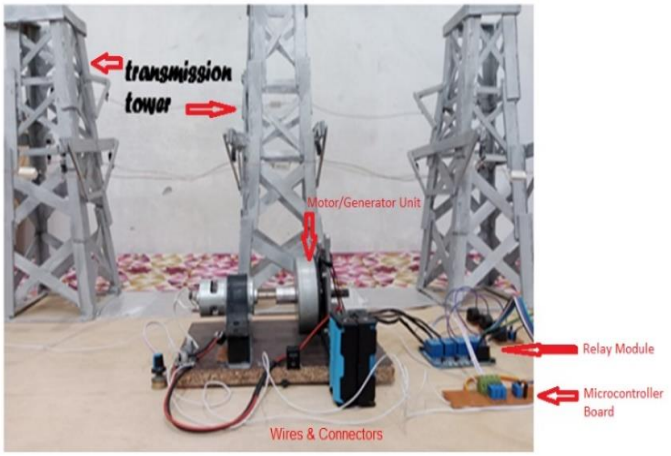


Figure 6. Prototype Model for Power Transmission Line Fault Detection and Analysis

4.7 Algorithm for Centralized Monitoring

Through the flowchart a technician can explain how to inspect the voltage and current and frequency levels of transmission lines step by step.

STEP 1: Start
STEP 2: Verify the Connection
STEP 3: Check the basic components for the power supply to the system
STEP 4: Initialize the Fault Creation Circuits, and current monitoring Sensors
STEP 5: ADC port should receive the values from the Current Sensors and Fault Creation Circuits
STEP 6: When the current values exceed the nominal preset values which are written in the coding section, the fault is identified by the ESP-32 the power to load is disconnected
STEP 7: When the parameters exceed the nominal encoded values, the tripping mechanism of the Relay is activating
STEP 8: These values are encrypted and then presented on LCD in EASYEDA
STEP 9: The process halts

Flow Diagram

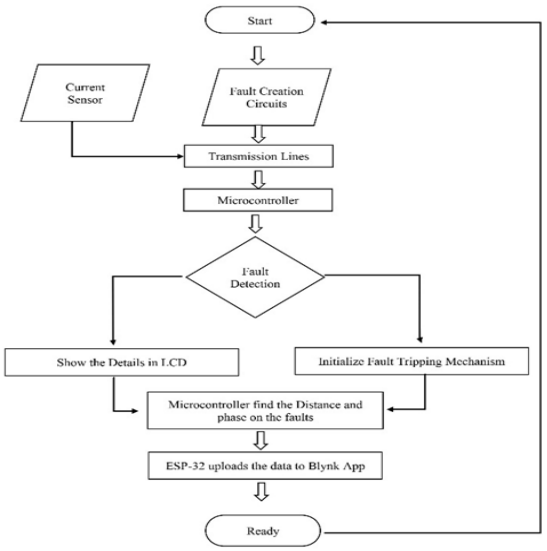
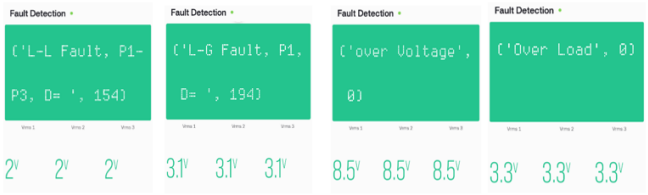


Figure 7. Working Flow Chart

The BLYNK App display representation of testing is shown in Figure 8.



(a) (b) (c) (d)

Figure 8. Fault Alert on BLYNK APP Fig. 8 (a) – Line to Line Fault, Fig. 8 (b) – Line to Ground Fault, Fig. 8 (c) – Over Voltage Condition, Fig. 8 (d) – Over Current Condition

4.8 Calculations

The resistance of the conductor is calculated by the following formula.

$$R = \frac{\rho L}{A} \quad (1)$$

Here Specific Resistance (”ρ”) is a property of any conductive material and is constant for conductors
(“A”) is the area of the conductor which is constant.
(“L”) is the length of the conductor.

To calculate the power delivered by the generator, use the three-phase power formula:

$$P = \sqrt{3} \times V \times I \times \cos(\emptyset) \quad (2)$$

For line-to-line faults, line-to-ground faults, and other fault conditions, the fault current can be analyzed using:

$$I_{fault} = \frac{V_{fault}}{Z_{fault}} \quad (3)$$

The impedance (Z) of a transmission line can be expressed as:

$$Z = R + jX \quad (4)$$

The sensitivity of the fault detection system can be improved by considering the derivative of the current with respect to resistance, helping in precise monitoring of changes due to faults:

$$\frac{dI}{dR} = -\frac{V^2}{R} \quad (5)$$

The short-circuit power level, which is relevant during fault conditions, can be calculated using:

$$S_{sc} = \frac{V_{rated}^2}{Z_{sc}} \quad (6)$$

When the length of the conductor increases then resistance also increases and the fault current decreases so when a fault occurs at a far position from the start of the transmission line then the length of the conductor and resistance increases and the fault current decreases. Similarly, when a fault occurs near the start of the transmission line then the length of the conductor and resistance decreases so the fault current increases. Firstly, the calculation of fault current through the input of current sensors is done then calibration of distance according to the magnitude of fault current [23].

5. Conclusion

The development of a Fault Detection and Analysis System (FDAS) for transmission lines using MicroPython on the ESP-32 microcontroller enhances power system reliability and efficiency. This research demonstrates FDAS capabilities for power system transmission line fault detection through sophisticated classification algorithms along with efficient real-time data processing capabilities and implementation using the ESP-32 platform as a scalable economic solution. The FDAS offers real-time monitoring, accurate fault detection, classification, and location using current and voltage sensor data. It alerts users of faults via the BLYNK App, enabling quick isolation and repair. This innovation reduces power outages, minimizes equipment damage, and lowers maintenance costs, ensuring reliable and safe power delivery. Overall, the FDAS enhances the resilience of power transmission systems. Using power systems in modern times requires the FDAS to detect faults through its combination of advanced analytics together with real-time monitoring and cost-efficient hardware for faster and improved detection capabilities. For future improvements FDAS can integrate machine learning for predictive maintenance and fault prediction, add GPS for precise fault location, and expand compatibility with various generators and microcontrollers. Incorporating IoT-based cloud storage will enhance scalability. Potential applications include smart grids, renewable energy, remote areas, and industrial networks where real-time fault detection and quick response reduce downtime and costs, enhancing grid resilience and power quality.

Acknowledgment

The scholars like to admit the support in providing a strong research environment by “The Islamia University Bahawalpur”, Pakistan and other coauthors institution.

References

[1] "Types of Faults and Effects in Electrical Power Systems," [Online]. Available: https://www.elprocus.com/what-are-the-different-types-of-faults-in-electrical-power-systems/.

[2] "Single Line-to-Ground Fault," [Online]. Available: https://circuitglobe.com/single-line-to-ground-fault.html. [Accessed 2024].

[3] Horowitz and Phadke "Power System Relaying" [Online]. Available: https://archive.org/details/PowerSystemRelaying/page/n7/mode/2up. [Accessed 2024].

[4] "Voltage Regulation of Transmission Lines: Dependencies and Parameters," [Online]. Available: https://resources.system-analysis.cadence.com/blog/msa2020-voltage-regulation-of-transmission-lines-dependencies-and-parameters. [Accessed 2024].

[5] C.R. Mason “The Art and Science of Protective Relaying" [Online]. Available: https://archive.org/details/art-science-of-protective-relaying-c-russel-mason [Accessed 2024].

[6] Vijay H. Makwana and Bhavesh R. Bhalja “Transmission Line Protection Using Digital Relays" [Online]. Available: https://link.springer.com/book/10.1007/978-981-10-1572-4 [Accessed 2024].

[7] B. Cao, Hongqing Liu and Qi Xie, "Analysis of the impact of transient overvoltage on grid-connected PMSG-based wind turbine systems," Front. Energy Res.,, vol. Volume 11, 17 November 2023.

[8] L. Zhu and Y. Luo, "Deep Feedback Learning Based Predictive Control for Power System Undervoltage Load Shedding," IEEE Transactions on Power Systems, vol. 36, no. 4, pp. 3349 - 3361, July 2021.

[9] F. M. Shakiba, M. Shojaee, S. M. Azizi and M. Zhou, "Real-Time Sensing and Fault Diagnosis for Transmission Lines," International Journal of Network Dynamics and Intelligence., vol. 1 Dec 2022, no. 1, p. 36–47, Dec 2022.

[10] "Line-to-Line Fault," [Online]. Available: https://circuitglobe.com/line-to-line-fault.html. [Accessed 2024].

[11] "Line to Ground Fault," [Online]. Available: https://testbook.com/objective-questions/mcq-on-line-to-line-to-line-to-ground-fault5eea6a0f39140f30f369e781#:~:text=Single%20line%2Dto%2Dground%20fault,to%20ground%20fault%20takes%20place.. [Accessed 2024].

[12] "Distance protection Relay: Working Principle," 15 May 2023. [Online]. Available: https://www.linkedin.com/pulse/distance-protection-relay-working-principle-world-of-electrical/.

[13] Ngwenyama, "Traveling Wave Fault Location Detection Technique for High Voltage Transmission Lines," 2021 2nd International Conference for Emerging Technology (INCET), 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9456334. [Accessed 2024].

[14] A. I. Khalyasmaa, B. A. Uteuliyev and Y. V. Tselebrovskii, "Methodology for Analysing the Technical State and Residual Life of Overhead Transmission Lines," IEEE Transactions on Power Delivery, vol. 36, pp. 2730 - 2739, 05 September, 2021.

[15] Zhan Wang and Stephen McConnell “Arc fault signal detection - Fourier transformation vs. wavelet decomposition techniques using synthesized data” IEEE Xplore, 16 October, 2014

[16] Misato Amano and Mami Matsumoto “Characteristics of False-Positive Alarms in the BacT/Alert 3D System” American society for microbiology, microbiology spectrum, vol.10, issue 3, 25 April,2022

[17] Vaishali Sonawane “Optimizing fault diagnosis in variable load conditions: A machine and deep learning approach for voltage source inverters” Journal of Integrated Science and Technology 2025, 13(3), 1057

[18] Rishabh Jain and Yaswanth Nag Velaga “Modern trends in power system protection for distribution grid with high DER penetration” Elsevier (Advances in Electrical Engineering, Electronics and Energy), 27 October, 2022

[19] Muhammad Kashif Sattar and Muhammad Waseem “IOT Based Fault Detection and Protection of Power Transformer in the Smart Grid” MDPI-Engineering Proceeding, 22 December 2021
[20] "Blynk IoT," 11, May 2024. [Online]. Available: https://play.google.com/store/apps/details?id=cloud.blynk&hl=en_US.
[21] "Video: Tutorial on Transmission Line Faults," [Online]. Available: http://www.electricalaxis.com/2016/11/video-tutorial-on-transmission-line.html. [Accessed 2014].
[22] "Overview Blynk Apps main functionality," 01 January 2024. [Online]. Available: https://docs.blynk.io/en/blynk.apps/overview.
[23] "Three-phase electric power," Wikipedia, The Free Encyclopedia, 16 May 2024. [Online]. Available: https://en.wikipedia.org/wiki/Three-phase_electric_power. [Accessed 28 May 2024].