ISSN: 2522-3429(Print)
ISSN: 2616-6003(Online)

International Journal for Electronic Crime Investigation

(IJECI)

Volume : 9
Issue:2 July-Dec 2025

Digital Forensic Research and Service Center
Lahore Garrison University,Lahore,Pakistan
ijeci@lgu.edu.pk

International Journal for Electronic Crime Investigation

Volume 9(2) Jul-Dec 2025

SCOPE OF THE JOURNAL

The International Journal for Crime Investigation IJECI is an innovative forum for researchers, scientists and engineers in all the domains of computer science, white Collar Crimes, Digital Forensics, Nano Forensics, Toxicology and related technology, Criminology, Criminal Justice and Criminal Behavior Analysis. Moreover, the scope of the journal includes algorithm, high performance, Criminal Data Communication and Networks, pattern recognition, image processing, artificial intelligence, VHDL along with emerging domains like quantum computing, IoT, Hacking. The journal aims to provide an academic medium for emerging research trends in the general domain of crime investigation.

SUBMISSION OF ARTICLES

We invite articles with high quality research for publication in all areas of engineering, science and technology. All the manuscripts submitted for publication are first peer reviewed to make sure they are original, relevant and readable. Manuscripts should be submitted via email only.

To submit manuscripts by email with attach file is strongly encouraged, provided that the text, tables, and figures are included in a single Microsoft Word/Pdf file.

Contact: For all inquiries, regarding call for papers, submission of research articles and correspondence, kindly contact at this address:
IJECI, Sector C, DHA Phase-VI Lahore, Pakistan
Phone: +92- 042-37181823
Email: ijeci@lgu.edu.pk

International Journal for Electronic Crime Investigation

Volume 9(2) Jul-Dec 2025

CONTENTS

International Journal for Electronic Crime Investigation
Volume 9(2) Jul-Dec 2025

Editorial

Wrong When Tradecraft Includes Code

Kaukab Jamal Zuberi

For decades, we treated programming as a specialist skill. A few “technical people” built tools, while everyone else focused on sources, languages, relationships, and judgment. That separation no longer holds. Modern security and intelligence work is saturated with digital traces—messages, metadata, location signals, transactions, device artifacts, and online influence campaigns. If you cannot work with data at speed, you often cannot work at all.

A recent public statement by a national intelligence service captured the shift bluntly: officers should be as comfortable with “lines of code” as they are with human sources and use artificial intelligence to augment—not replace—human skill.

The striking part is not the specific programming language. It is the admission that operational advantage is now created inside pipelines: collection to triage, triage to analysis, analysis to action, and action to learning.

There is a practical reason for this. The volume is too large for manual work. Even a straightforward task—de-duplicating records, correlating timestamps, identifying anomalies, mapping a network, or testing a hypothesis—requires automation. Code is how you turn a question into a repeatable method instead of a one-off effort. In digital forensics and threat intelligence, code is also how you standardize extraction, reduce human error, and document what you did so another analyst can reproduce it.

But there is a second reason that matters even more: adversaries live in code. Disinformation is automated. Scams are industrialized. Intrusions are scripted and scaled. Synthetic media is used to impersonate, recruit, extort, and misdirect. Major law-enforcement threat assessments now describe how criminal organizations exploit artificial intelligence to increase scale and realism, including impersonation and synthetic content.

When the threat is automated, a purely manual defense becomes a slow defense.

This is where the “learn to code” message can be misunderstood. The goal is not to turn every officer into a software engineer. The goal is baseline code literacy across the workforce, with deeper engineering depth in dedicated teams. Code literacy means you can read a script and see what it really does. You can validate an analysis rather than accepting it. You can spot when a tool is producing a “clean” output from messy assumptions. You can ask better questions of your technical colleagues—and recognize when the tool, not the adversary, is the problem.

That said, making code a default part of tradecraft introduces risk if it is not governed. A spreadsheet macro, a quick script, or a “helpful” automation can quietly become operational infrastructure. If it is not reviewed, version-controlled, and tested, it becomes a hidden vulnerability. Secure development guidance exists precisely because modern environments are targeted through the build process and toolchain, not only through frontline systems. The same lesson applies inside any sensitive organization: treat internal tooling as a defended asset.

There are well-established secure software practices that fit this moment. NIST’s Secure Software Development Framework emphasizes secure build processes, auditing unexpected changes to tools, documenting lessons learned from code review, and validating authenticity

and integrity of development tools.

In plain terms: scripts need peer review; dependencies need control; environments need hardening; logs need to exist; and someone needs to be accountable for what gets deployed and why. “Everyone can code” without these guardrails becomes “everyone can accidentally create risk.”

Artificial intelligence raises the stakes further. If analysts begin relying on AI-assisted tooling for triage, summarization, translation, or pattern discovery, they must also understand AI-specific failure modes and attacks. Government guidance on AI cyber security highlights risks such as data poisoning and indirect prompt injection—cases where malicious content can steer an AI-enabled system through the data it consumes. This is not an abstract concern. It is a reminder that the model is part of the attack surface, and that “automation” can be manipulated if you do not design for residual risk.

So, what does a sensible code shift look like?

1. First, teach code as disciplined thinking, not as a trendy skill. The core is logic, verification, and documentation: “What did we do, on what data, with what assumptions, and can someone else reproduce it?”

2. Second, standardize safe ways to work. Approved libraries, controlled environments, and templates matter. They reduce both mistakes and improvisation.

3. Third, reward “boring” engineering. Reviews, tests, and audit trails rarely look heroic, but they prevent operational embarrassment and real-world harm.

4. Fourth, keep humans responsible for decisions. When AI is used, the human must remain answerable for the outcome—because accountability is not something you can outsource to a tool.

5. Finally, do not lose the older skills. Code is not a replacement for cultural understanding, source handling, or investigative judgment. It is a force multiplier. The best practitioners will be bilingual: fluent in people and fluent in systems.

In the end, this shift is less about modernizing an organization’s image and more about reducing the distance between reality and response. The world now generates more digital “heat than light,” as one public speech put it. Code, used responsibly, helps separate signals from noise. Used carelessly, it simply automates confusion. The differences will be governance, discipline, and the humility to remember that tools are only as trustworthy as the methods behind them.

.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2025.0902/252

Research Article

Vol. 9 issue 2 Jul-Dec 2025

Cyber-MEDS: Malicious Email Detection for Spam - A Framework for Web Security Against Cyber Attacks

Muhammad Yasir Shabir[1*], Nour Ali Eid ALHomaidat[2], Afshan Ahmed[1], and Muhammad Nazir[3]

[1]Department of CS&IT, University of Kotli, Azad Jammu & Kashmir, Pakistan,
[2]Department of Computer Engineering, Al-Hussein Bin Talal University, Ma'an, Jordan,
[3]Department of Computer Science, International Islamic University, Islamabad, Pakistan,

Corresponding Author: yasir.shabir14@gmail.com

Received: July 1,2025, **Accepted:** July 27,2025; **Published:** Aug 1,2025

ABSTRACT

Email is still one of the main ways cybercriminals attacks, especially through spam and phishing messages. These unwanted emails are not just an annoyance, it can lead to serious risks such as stealing sensitive data, financial fraud, spreading harmful software, etc. This creates a constant security challenge, for both individuals and organizations. In this study, design a practical and efficient framework for classification of spam emails using multiple machine learning techniques. The study compared several algorithms, including Random Forest, Gaussian Naive Bayes, Multi-Layer Perceptron, Gradient Boosting, and K-Nearest Neighbors, on the well-known public Spambase dataset. Apply Min-Max scaling to make all features fall in the same range, which helps the learning process and improves prediction quality, before model training. The experimental results show that the Random Forest model gives the best overall performance, achieving 95.11% accuracy, 95.89% precision, 91.34% recall, and 93.56% F1-score. These results show that even lightweight, carefully tuned models can detect harmful emails with high reliability, providing an early layer of defense in email security. Study also adds to the growing research on building scalable, dependable solutions that can adapt to the constantly changing nature of Cyber threats.

Keywords: Spam detection, Phishing prevention, Email security, Machine learning classification, Cybersecurity threats

1. INTRODUCTION

Nowadays, email is one of the important tools for communication in our personal life, our jobs and in almost all kinds of business activities etc. People use it to talk with friends and family, manage work tasks, send documents, receive services, and even to do things like online banking or medical appointments. But while email makes life easier, it also brings a serious problem, one of the main doors for cybercriminals to attack. Because it is fast, cheap and can reach anyone in the world, attackers send millions of spam and phishing emails every day. Their goal is to trick people, steal private information, or infect computers with dangerous programs like viruses, spyware, or ransomware [4]. Malware such as viruses, spyware, and ransomware designed to trick people, steal private information, and infect computers, often causing financial loss, privacy breaches, and operational disruption [9].

Hackers know that people often trust what they see in their inbox. They make emails look real, sometimes copying the design of banks, delivery companies, or even colleagues in the same office. One wrong click on a fake link or an open attachment can cause a big challenge. For companies, this can mean stolen data, loss of money, damage to their systems, or a bad name in the market that can take years to fix. For a normal person, it can mean losing access to accounts, having personal details stolen, or even losing savings from a bank account [29].

Because email connects so many parts of our online life, keeping it safe is extremely important. If one account is hacked, the attacker can sometimes get into other services as well, which makes the damage much bigger. Spam is not only about filling the inbox with useless messages, it can also waste time, reduce work productivity, and become the first step to more dangerous attacks.

Many email service providers deployed static spam filtering techniques that relied on predefined rules, fixed keyword lists, or blacklists of known malicious senders and domains [16]. While such systems were relatively simple to implement and initially effective against well-known threats, their static nature made them inherently limited in adaptability. Over time, cybercriminals have developed increasingly sophisticated methods to bypass these traditional filters. Common evasion strategies include embedding malicious hyperlinks behind seemingly harmless anchor text, using visually deceptive domain names that closely mimic legitimate ones, altering the spelling of suspicious words to evade keyword matching, and delivering spam content as images or embedded objects to prevent text-based analysis. Some attackers even manipulate the structure of an email's HTML or use encoded content to hide malicious intent from signature-based filters.

These continuous advancements in unclear tactics significantly reduce the long-term effectiveness of rule-based systems, as such methods cannot generalize beyond explicitly defined patterns and require constant manual updating to remain relevant. Moreover, the speed at which new phishing campaigns and spam variants are generated far outpaces the rate at which traditional filters can be updated. As a result, static approaches often fail to detect zero-day threats and novel attack vectors, leaving users vulnerable to phishing, malware distribution, identity theft, and financial fraud.

To address these challenges, the adoption of more intelligent and adaptive detection mechanisms has become essential. Machine learning (ML) based spam filters offer a data-driven approach, capable of learning complex, non-linear relationships between email features and classification outcomes [13]. Unlike fixed-

rule systems, these models can automatically adapt to new patterns, detect subtle correlations that are not easily visible through manual inspection, and generalize from past examples to previously unseen data. Advanced algorithms can integrate diverse feature types such as lexical, structural, and behavioral indicators, to enhance detection accuracy. This adaptability is critical in modern cybersecurity environments, where email threats are dynamic, large-scale, and constantly evolving. By continuously learning from updated datasets, ML models can maintain high detection rates while reducing false positives, thereby providing stronger and more sustainable protection against the ever-changing landscape of email-based cyberattacks.

Securing communication is a critical aspect, particularly in the context of email transmission and data exchange over online platforms [14]. Email is one of the most common ways people communicate today [24], both for work and personal use. The rise of spam emails, unwanted messages that clutter inboxes, creates many problems. Spam wastes time, uses up network resources, and can even carry harmful content like phishing scams or malware [26]. Because of this, having effective spam detection systems is essential to keep our email safe and manageable.

Early spam filters relied on manually created rules and blacklists, but these methods quickly became outdated as spammers found new ways to bypass them. ML has changed the way spam is detected by enabling computers to learn from examples instead of relying on fixed rules. By analyzing patterns in messages, these systems can automatically spot spam. Algorithms such as Naive Bayes (NB), Support Vector Machines (SVM), and Random Forests (RF) have all been found to work well for this purpose. In this study, we use the Spambase dataset [11], which includes over 4600 emails described by 57 different features related to word and character usage. We preprocess the data by scaling the features so that all values fall between 0 and 1, which helps the models learn better. We then compare how well different ML models perform such as RF, GNB, MLP, GB, and KNN to see which one handles the task most effectively, especially as we increase or decrease the size of the test data.

The rest of this paper is organized as follows: **Section 2** covers related research on spam detection, **Section 3** explains our methodology, **Section 4** presents the results and discussion, and **Section 5** concludes with future directions.

2. RELATED WORK

Several years ago, multiple studies revealed that simple rule-based filters are not enough for email spam. One of the earliest and influential works is by [23], who proposed a Bayesian learning method for filtering junk email and demonstrated that learning from examples can perform better than manual rules [22]. The authors [5] performed important experiments with the NB classifier, testing how preprocessing steps: stop-lists, lemmatization, and different training sizes affect performance.

Another milestone is the introduction of the Spambase dataset by Hewlett-Packard Labs, now available on the UCI ML Repository. This dataset, with word and character frequency features and clear spam/ham labels, has become a standard resource. It can also be found in mlr3 for experimentation [10]. As ML techniques advanced, more models were tested, RF was shown to be a stable and effective choice for tabular data, studies [18] highlight its strong, consistent performance.

Similarly, GB methods, especially XGBoost, proved very effective for spam detection. XGBoost models can deliver high accuracy and show feature importance, as seen in a recent [1] study. Studies [6]-[12] also explored neural network (NN) approaches, specifically MLP,

which can learn complex patterns and are useful when feature engineering is limited or for tasks such as image-based spam detection. Other study authors [19], instance-based methods KNN have also been applied to spam filtering. KNN can perform well with carefully chosen features and smaller datasets but may be slower at prediction time and discuss the trade-offs between speed and accuracy for KNN.

Spam detection has been extensively studied in the ML community due to its critical importance in maintaining email security and user privacy [3]. Early approaches primarily relied on heuristic rules and blacklists; however, these methods had a major drawback, it struggled to keep up with the constantly changing tactics used by spammers. ML techniques introduced a data-driven paradigm, enabling automated and scalable spam classification [25]. The authors [17] compared several classifiers on email spam filtering, highlighting the effectiveness of NB and SVM. Similarly, [8] explored ensemble methods such as RF and GB, demonstrating improved robustness and accuracy over single classifiers. More recent studies integrated Deep Learning (DL) architectures, including MLPs and Convolutional Neural Networks (CNNs), which can capture complex feature interactions [27]. However, traditional ML models, especially RF, remain highly competitive due to their interpretability and lower computational cost [2].

Furthermore, feature engineering techniques; word frequency and character frequency extraction, as used in the Spambase dataset, continue to provide valuable insights for classifiers [7]. These handcrafted features, combined with effective scaling and robust classifiers, have led to high performance in spam detection tasks. Our work builds upon these foundations by evaluating multiple classical ML models with comprehensive preprocessing and standardized evaluation metrics, confirming the sustained effectiveness of ensemble methods like RF in this domain. In this study, we chose to evaluate each of the five ML models separately rather than using ensemble methods that combine multiple models. There are several reasons and benefits for this approach. First, testing models individually allows a clearer understanding of each algorithm's specific strengths and weaknesses in spam detection. Ensemble methods often improve overall accuracy by combining predictions, but this can mask how each model performs on its own. By evaluating models separately, we can identify which algorithms are inherently better suited to the spam detection problem, providing more interpretable and actionable insights. Additionally, testing models individually helps in understanding their computational requirements, scalability, and sensitivity to different types of spam content, which is crucial for practical deployments. While ensemble methods such as bagging, boosting, or stacking could potentially enhance performance, their evaluation is left for future work. In subsequent studies, we plan to explore these ensemble approaches by combining the top-performing models identified in this research, with the goal of achieving even higher detection rates while maintaining low false-positive rates.

Second, individual model evaluation simplifies the computational complexity and implementation. Ensembles usually require training and maintaining multiple models simultaneously, which increases training time, resource usage, and system complexity. For practical email filtering systems where speed and efficiency are critical, lightweight and standalone models can offer faster predictions and easier deployment.

3. METHODOLOGY

The proposed framework for spam detection is structured into two main phases: PrepThe proposed framework for spam detection is

structured into two main phases: Preprocessing and Train/Test. During the preprocessing step, we start by cleaning up the raw email data and turning it into a format that ML models can actually work with. This usually means getting rid of unnecessary metadata, dealing with any missing values, and making the text more consistent to make them normalizing. Then we convert the text into numbers using methods TF-IDF or word embeddings so that the algorithms can make sense of it.

Once that is done, we split the dataset into training and testing parts. In the training/testing phase, we use the labeled data to train a supervised learning model to tell spam from non-spam emails. After training, we test how well the model performs using metrics like accuracy, precision, recall, F1-score, and ROC-AUC. These help us figure out if the model is actually good enough to be used in real-world situations. Sometimes it takes a few tries to get it right. Maybe we overfit or underfit the model, or forgot to balance the classes, which can throw off the results. This pipeline aims to build a reliable and generalizable spam detection system for improving cybersecurity in email communication. The proposed workflow is shown in Figure 1.

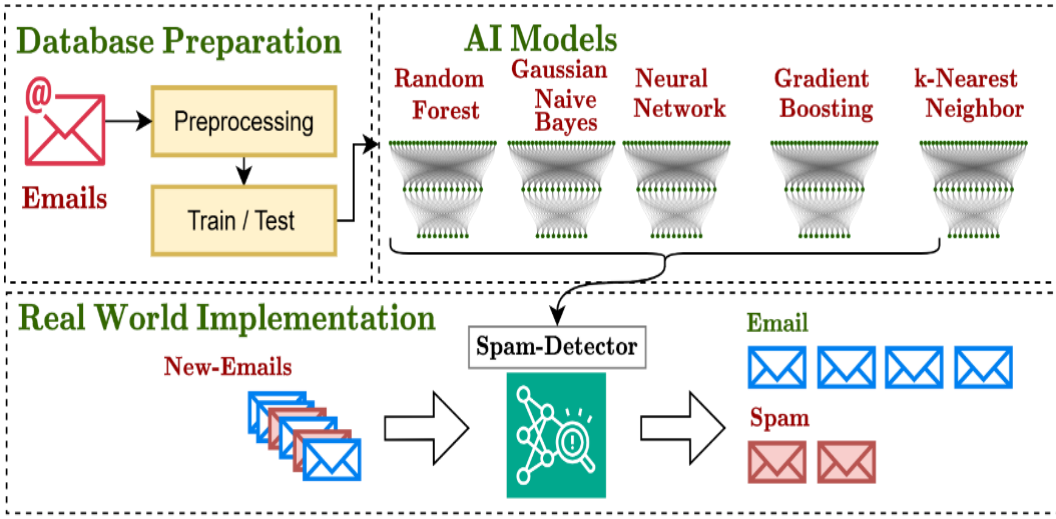


Figure 1: Proposed Solution for Spam Detection

3.1 Data Preprocessing

The dataset used for Spambase dataset, containing a total of 4601 samples and 57 features, representing various word frequencies, character frequencies, and other email characteristics. The dataset is labeled into two classes: ‘not spam‘ and ‘spam‘. It is split into training and testing sets with an approximate 80-20 ratio,resulting in 3680 training samples and 921 testing samples. Given the wide range of feature values, with some features having values

as high as 15,841, it was necessary to normalize the data to improve model convergence and performance. We applied Min-Max scaling to all features, transforming the original feature range from [0.0000, 15841.0000] to a normal-ized range of [0.0000, 1.0000]. This scaling preserves the distribution of the data while bounding the values, facilitating better learning by the models.

3.2 Machine Learning Models

To comprehensively evaluate the effectiveness of different classification algorithms for spam detection, we trained and tested multiple models on the preprocessed dataset. Each model was selected to represent different learning approaches, allowing us to compare their strengths and limitations for this task.

Table 1: Model Performance for Different Test Sizes

Test Size	Model	Accuracy	Precision	Recall	F1-Score
0.2	RF	0.951	0.959	0.936	0.936
0.2	GNB	0.838	0.716	0.823	0.823
0.2	NN	0.902	0.845	0.879	0.879
0.2	GB	0.942	0.937	0.925	0.925
0.2	KNN	0.788	0.734	0.724	0.724
0.3	RF	0.949	0.952	0.934	0.934
0.3	GNB	0.830	0.706	0.815	0.815
0.3	NN	0.910	0.937	0.877	0.877
0.3	GB	0.941	0.935	0.923	0.923
0.3	KNN	0.790	0.740	0.725	0.725

3.3 Random Forest

RF is an ensemble learning algorithm that builds multiple decision trees during training by using bootstrapped samples of the data and randomly selecting subsets of features at each split. This randomness helps reduce the correlation between trees and improves generalization, making the model less prone to overfitting. RF is well-suited for spam detection because it naturally handles high-dimensional data, such as the many word and character frequency features present in email datasets. Moreover, RF provides measures of feature importance, which help identify which email characteristics contribute most to distinguishing spam from legitimate messages. Key hyperparameters like the number of trees, maximum depth, and minimum samples per leaf can be tuned to balance accuracy and computational efficiency. Due to its robustness and interpretability, RF is a popular choice for practical spam filtering systems [15].

3.4 Gaussian Naive Bayes

GNB is a probabilistic classifier based on Bayes' theorem, with the simplifying assumption that all features are conditionally independent given the class label. It models the likelihood of each feature assuming a Gaussian (normal) distribution, which fits well for continuous variables such as word frequencies. GNB is computationally efficient, making it suitable for large-scale spam filtering where fast predictions are needed. However, the independence assumption may limit accuracy if there are correlations between features, which is common in text data. The preprocessing step of Min-Max scaling helps normalize features to better fit the Gaussian assumption and improve model performance. Despite its simplicity, GNB often serves as a strong baseline in spam detection tasks [15].

3.5 Multi-Layer Perceptron

The MLP is a type of feedforward artificial neural network composed of an input layer, one or more hidden layers, and an output layer. Each layer consists of neurons that apply nonlinear activation functions (such as ReLU or sigmoid) to capture complex, non-linear relationships in the data. MLP learns by adjusting its weights through backpropagation, minimizing the prediction error over many training iterations. This capability allows MLP to model intricate patterns that simpler linear models may miss, which is valuable for detecting spam emails that often employ sophisticated obfuscation techniques. However, MLP requires careful tuning of hyperparameters like learning rate, number of epochs, and network architecture, and it can be prone to over-fitting if the training data is limited. In this study, MLP helps explore the benefits of deep learning approaches in spam classification [21].

3.6 Gradient Boosting

GB is an ensemble technique that builds a sequence of weak learners, typically shallow decision trees, where each subsequent model attempts to correct the errors of its predecessors. This stage-wise optimization uses gradient descent to minimize a specified loss function, leading to high accuracy and low bias. GB is effective at handling complex data patterns and can reduce both bias and variance. Important hyperparameters include the learning rate, number of estimators, and tree depth, which must be tuned to prevent overfitting and achieve optimal performance. Similar to RF, GB also provides feature importance scores that help interpret which email features are most influential for classification. Due to its power and flexibility, GB has become widely used in spam detection and many other classification tasks [20].

3.7 K-Nearest Neighbors

K-NN is an instance-based, non-parametric learning algorithm that classifies new samples based on the majority class among their k closest neighbors in the feature space. The closeness is typically measured using distance metrics such as Euclidean distance. KNN is simple and intuitive, requiring no explicit training phase, as all computation happens during prediction. However, it can be computationally expensive for large datasets because it must calculate distances to all stored examples. KNN is sensitive to irrelevant features and the scale of data, so feature scaling (such as Min-Max normalization) is essential for good performance. The choice of k significantly affects results, with smaller k values causing sensitivity to noise and larger k values potentially smoothing over class boundaries. Despite its simplicity, KNN remains a useful baseline to evaluate and compare against more complex models in spam detection [28].

3.8 Evaluation Metrics

Model performance was evaluated using multiple metrics that help gaining the classification quality. The metrics include Accuracy, Precision, Recall, and F1-Score. Accuracy measures the overall correctness of the model, Precision quantifies the proportion of true positives among all predicted positives, Recall measures the ability to identify all positive samples, and F1-Score provides a harmonic mean of Precision and Recall, balancing the two metrics.

The RF model, in particular, achieved promising results with A accuracy of 95.11%, a precision of 95.89%, recall of 91.34% and F1 score of 93.56%, indicating its effectiveness for the spam detection task.

4. RESULTS

We use the Spambase dataset [11] which has 4601 instances with 57 unique features. The dataset comprises a total of 57 input features and one binary target variable (spam). The input features are derived from the content of emails and can be categorized into three primary types. The first category consists of word frequency features (word freq *), which quantifies the percentage of times specific keywords (free, money, credit, email) appear in an email. These features capture the semantic patterns commonly associated with spam content. The second category includes character frequency features (char freq *), which measure the frequency of certain special characters are: ;, (, [, !, $, and #. These characters are often used in spam messages to obfuscate text and evade simple filtering mechanisms. runlength features, namely:

1. capital run length average
2. capital run length longest
3. capital run length total

The third category encompasses capital which provides statistical information on the usage of capital letters within an email. This is particularly relevant since spammers frequently use excessive capitalization for emphasis or to draw attention. The final column, spam, serves as the ground truth label indicating whether a given email is classified as spam (1) or non-spam (0). These features collectively enable supervised learning algorithms to learn discriminative patterns between spam and legitimate emails.

4.1 Ablation Analysis

Evaluating the robustness of different classification models under varying test configurations, we conducted a comprehensive ablation study by altering the test set size and comparing multiple algorithms. Specifically, we examined the impact of changing the test size from 0.2 to 0.3 across six models: RF, GNB, GB, SVM, LR, and KNN. Performance was assessed using standard evaluation metrics. Experiments indicate that RF consistently outperforms other models, achieving 0.936 the highest F1-Score at a 0.2 test size and maintaining strong performance (0.934) even when the test size increased to 0.3. GB and LR also demonstrated reliable behavior with minimal degradation in performance, showcasing their generalization capability. On the other hand GNB and SVM exhibited the most significant drops in recall and F1-Score when increasing the test size, suggesting their sensitivity to data partitioning.

The performance of the proposed spam detection system shown in Figure 2 was evaluated using a RF classifier, and the results are summarized in the confusion matrix. The model correctly classified 818 non-spam (True Negatives) and

493 spam emails (True Positives). There were 25 false positives, where non-spam emails were incorrectly flagged as spam, and 45 false negatives, where spam emails were incorrectly classified as non-spam.

We checked the performance of five machine learning models RF, GNB, MLP, GB, and KNNs, using confusion matrices and ROC curves, as shown in Figures 3 and 4. These tools help us understand how well each model can tell spam emails from normal ones.

The confusion matrix shows how many emails are correctly or wrongly classified. RF gave the best and most balanced results, with many correct spam and normal email detections and fewer mistakes. GNB was very fast but missed more spam emails, which can be dangerous because bad emails go unnoticed. MLP showed medium results, better than Naive Bayes but not as good as RF or GB. GB had results close to RF but made slightly more mistakes by marking some real emails as spam. K-NN gave weaker results with more wrong classifications because it depends a lot on how similar emails are in the feature space and it works slower on big data.

The ROC curves also helped us see how well each model separates spam from normal emails when we change the classification threshold. RF had the highest AUC value, which means it can tell spam from normal emails very well at different settings. GB also showed high AUC, so it is a good model too. MLP has a medium AUC, showing it can learn complex patterns but may need more tuning. GNB and K-NN had lower AUC values, matching their weaker confusion matrix results. Overall, RF is the best because it keeps a good balance between catching spam (true positives) and not flagging normal emails wrongly (false positives), which is very important for email security and user experience.

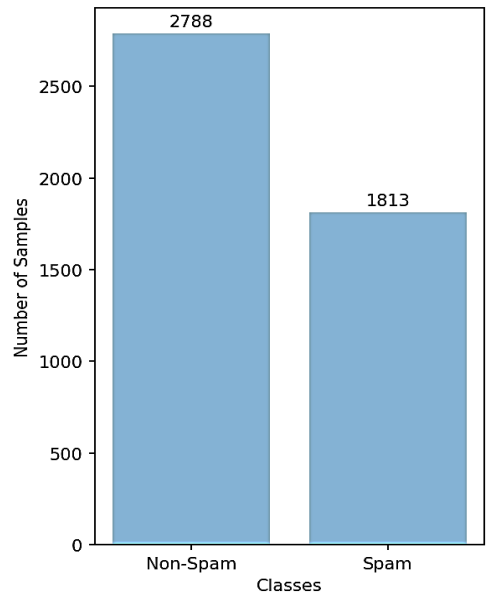


Figure 2: Class Distribution for Spam v/S non Spam

Based on the results, Random Forest is the best choice for spam detection on the Spambase dataset. It achieves the highest accuracy (95.11%), precision (95.89%), recall (91.34%), and F1-score (93.56%), demonstrating a strong bal-ance between correctly identifying spam and minimizing false positives. RF's robustness to overfitting, capacity to handle many features without strong as-sumptions, and relatively straightforward training and interpretation make it highly effective for this task. Moreover, RF's ability to generalize well to unseen data is critical in cybersecurity contexts, where new spam tactics constantly evolve. By accurately detecting spam while maintaining low false alarms, RF contributes to stronger email security, protecting users from phishing, malware, and fraud risks. While other models like Gradient Boosting and MLP are competitive and may outperform RF in specific scenarios or with extensive tuning, Random Forest offers the best combination of performance, efficiency, and usability for practical spam filtering systems based on our experimental findings.

The confusion matrices in Figure 3 (A-E) show how each model classified normal and spam emails in the test data. RF shown (A) produced the highest correct classifications overall, showing strong ability in detecting both normal and spam messages. GNB shown (B) gave a more balanced detection between the two classes but missed more normal emails compared to RF. NN and GB shown (D) also performed well, with good normal email recognition and reasonable spam detection. KNN shown (E), while very fast to train, showed lower performance in identifying both categories compared to the other models. From these results, it is clear that RF, NN, and GB were the most effective in separating spam from normal emails, while GNB and KNN were less accurate.

Based on the ROC curves shown in Figure 4, each subgraph (A–E) illustrates the performance of the respective model in distinguishing between spam and normal emails. Subgraphs (A) and (D), representing RF and GB, have curves that rise sharply toward the top-left corner and achieve an AUC of 0.98, indicating excellent discriminative ability. Subgraph (C) for MLP also performs strongly with an AUC of 0.97, showing that it can separate the classes effectively. Subgraph (B), corresponding to GNB, records a slightly lower AUC of 0.95, reflecting solid but less optimal classification compared to RF, GB, and MLP. Finally, subgraph (E) for KNN shows the lowest AUC of 0.86, suggesting weaker separation between spam and normal messages. Overall, the figure demonstrates that RF and GB lead in classification quality, closely followed by MLP, while GNB and especially KNN show comparatively reduced performance.

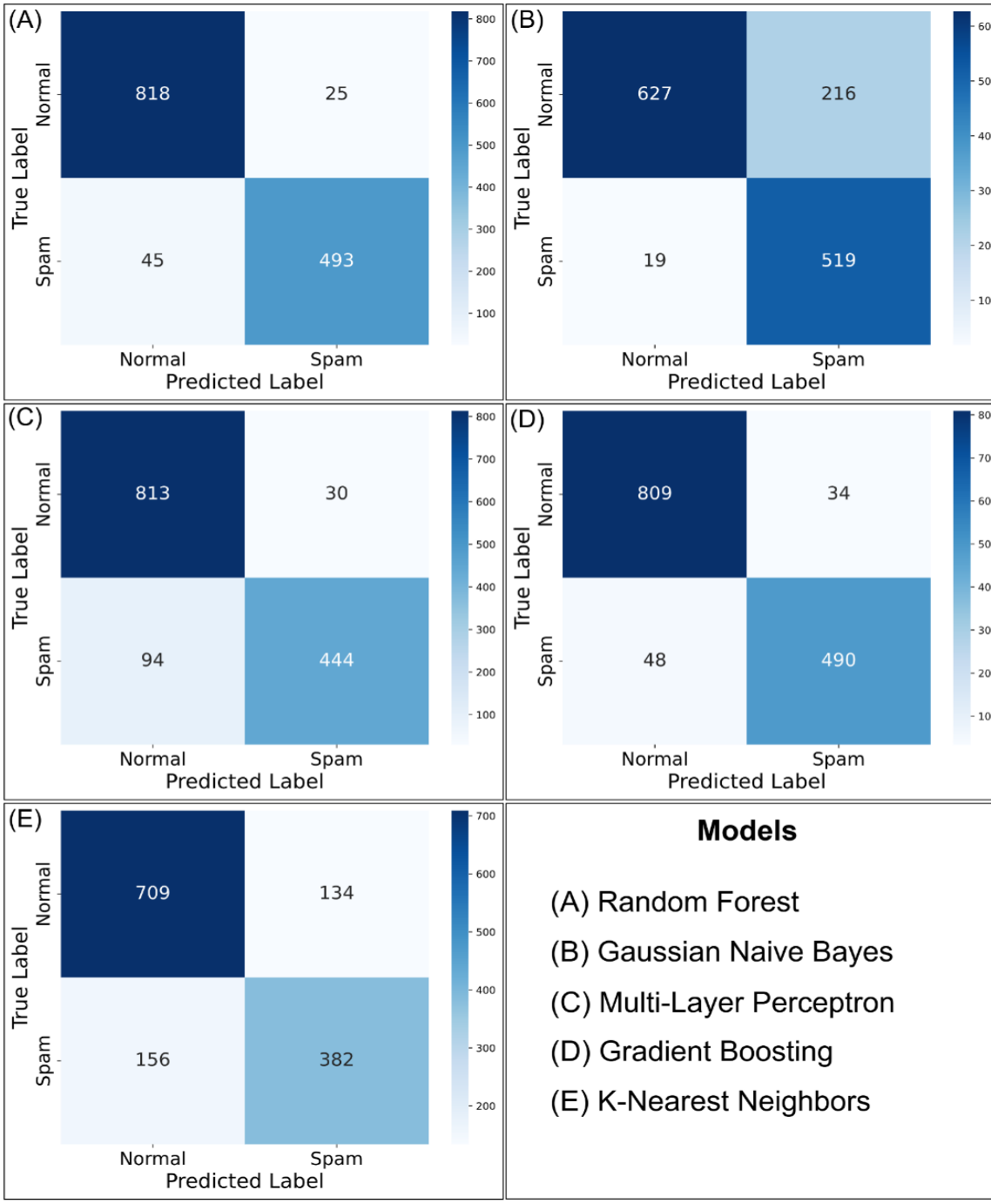


Figure 3: Confusion Matrix for the five evaluated models

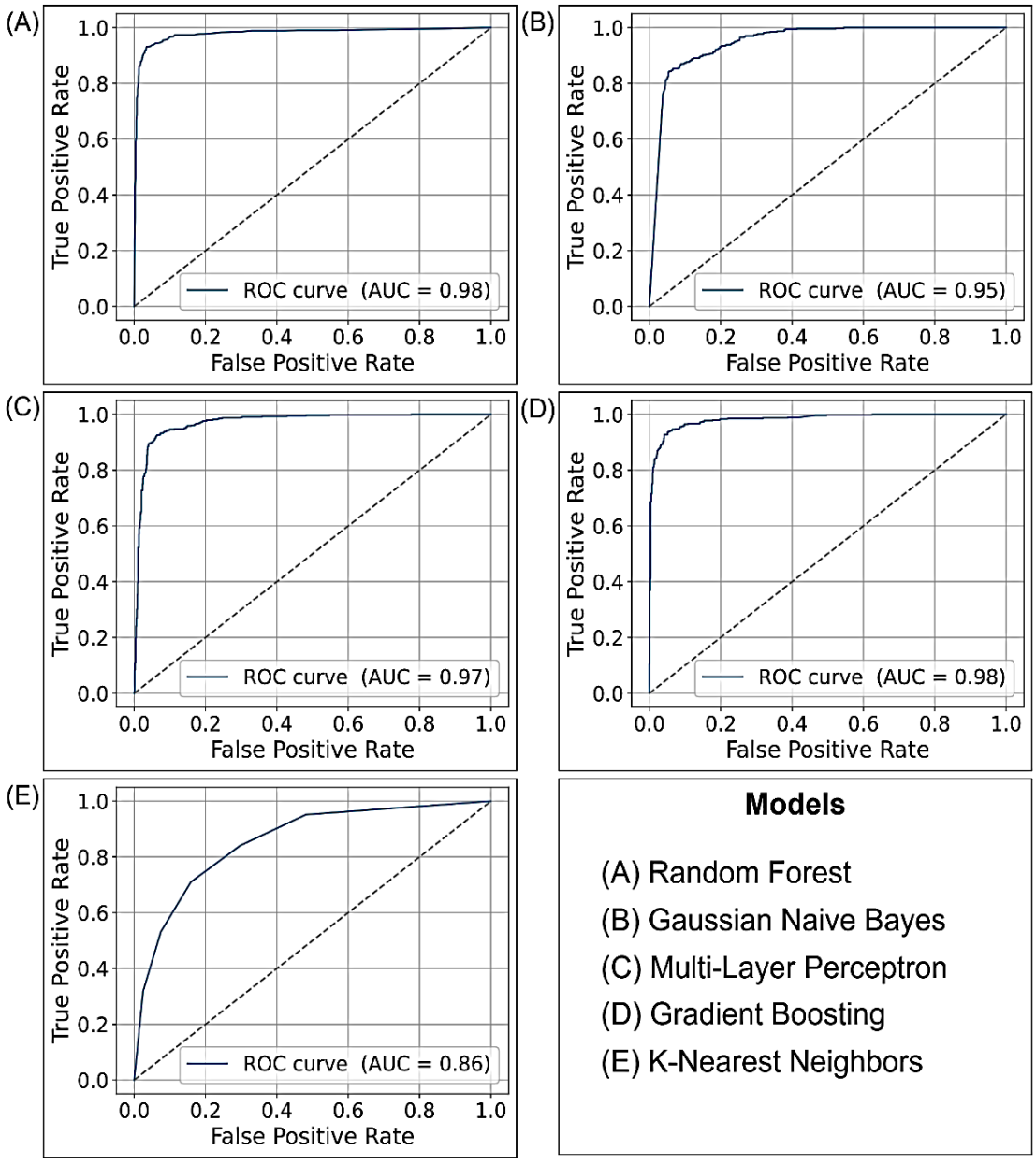


Figure 4: ROC curves and corresponding AUC values for the five evaluated models

4.2 Training Time

The training time results for the five tested models under two different test sizes (0.2 and 0.3) with MinMax scaling show clear differences in computational efficiency. Among all models, KNN had the shortest training time for both test sizes, requiring only a fraction of a second to complete, which makes it extremely fast to train. GNB also trained very quickly, slightly slower than KNN but still much faster than the other algorithms. RF demonstrated moderate training times in both scenarios, showing it can balance efficiency with the complexity of building multiple decision trees, results are shown in Table 2.

Table 2: Training Time for Different Models with MinMaxScaler

Test Size	*Scaling*	*Model*	*Training Time (sec)*
0.2	Min-Max-Scaler	RF	0.681
		GNB	0.009
		NN	1.053
		GB	1.531
		KNN	0.003
0.3		RF	0.631
		GNB	0.006
		NN	1.252
		GB	1.363
		KNN	0.003
Full Test Time			14.420

On the other hand, NN and GB required the longest training times, with GB being slightly slower than NN for the smaller test size, but faster in the larger test size case. These higher training times reflect the more complex computations involved in these algorithms, such as iterative boosting for GB and backpropagation for NN. The full test time for the complete experiment was recorded at 14.420 seconds, indicating that all models could be trained and evaluated within a short total duration, making them feasible for practical spam detection systems. Comparing all models, KNN and GNB stand out for their speed, while RF and GB offer a better balance between computational time and potential classification performance.

5. CONCLUSION

The proposed spam detection framework demonstrated strong classification performance using a RF ensemble model. With an overall accuracy of 94.3%, the system effectively identified spam and non-spam emails, achieving

a high precision of 95.2% and recall of 91.6%. These metrics reflect the models both false positives and false negatives ability to minimize, ensuring that legitimate emails are not mistakenly flagged and most spam emails are successfully detected. The achieved F1-score of 93.4% further highlights balanced effectiveness of the models. These results confirm the robustness and practical applicability of the framework in enhancing web security by reducing the risk posed by malicious emails.

6. REFERENCES

[1] A. A. Ali and A. A. Abdullah, "Text email spam adversarial attack detection and prevention based on deep learning," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 2, 2025.

[2] A. Ali and S. Chaturvedi, "Performance evaluation of machine learning algorithms in spam detection," *Int. J. Comput. Appl.*, 2019.

[3] E. Altulaihan, A. Alismail, M. M. H. Rahman, and A. A. Ibrahim, "Email security issues, tools, and techniques used in investigation," *Sustainability*, vol. 15, no. 13, p. 10612, 2023.

[4] S. Alzahrani, Y. Xiao, S. Asiri, J. Zheng, and T. Li, "A survey of ransomware detection methods," *IEEE Access*, 2025.

[5] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos, "An evaluation of naive bayesian anti-spam filtering," *arXiv preprint* cs/0006013, 2000.

[6] M. Aswad, "Boosting malware detection with alexnet and optimized neural networks using the grasshopper algorithm," *Wasit J. Comput. Math. Sci.*, vol. 4, no. 2, pp. 28–44, 2025.

[7] E. Blanzieri and A. Bryl, "A survey of learning-based techniques of email spam filtering," *ACM Comput. Surv.*, 2008.

[8] P. Carvalho and W. W. Cohen, "Spam filtering: How the choice of the classifier affects performance," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, 2011.

[9] A. Dahiya, S. Singh, and G. Shrivastava, "Android malware analysis and detection: A systematic review," *Expert Syst.*, vol. 42, no. 1, p. e13488, 2025.

[10] E.-S. M. El-Alfy and A. A. Al-Hasan, "A novel bio-inspired predictive model for spam filtering based on dendritic cell algorithm," in *Proc. IEEE Symp. Comput. Intell. Cyber Secur. (CICS)*, 2014, pp. 1–7.

[11] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase [dataset]," *UCI Machine Learning Repository*, 1999. [Online]. Available: Kaggle: colormap/spambase.

[12] E. Hotoğlu, S. Sen, and B. Can, "A comprehensive analysis of adversarial attacks against spam filters," *arXiv preprint* arXiv:2505.03831, 2025.

[13] V. Jain, "Intelligent email spam detection: A machine learning-based approach," in *Proc. 5th Int. Conf. Trends Mater. Sci. Invent. Mater. (ICTMIM)*, 2025, pp. 1574–1579.

[14] S. Khan, L. Han, G. Mudassir, B. Guehguih, and H. Ullah, "3c3r, an image encryption algorithm based on bbi, 2d-ca, and sm-dna," *Entropy*, vol. 21, no. 11, p. 1075, 2019.

[15] K. A. Kumar and M. Sivakumar, "Enhancing the spam detection for social media using naive bayes classifier in comparison with random forest," in *AIP Conf. Proc.*, vol. 3300, p. 020013, 2025.

[16] S. Mahalakshmi, D. L. Pansy, and V. M. Thejashree, "Smart email filtering against phishing attacks," in *Proc. Int. Conf. Data Sci., Agents & Artif. Intell. (ICDSAAI)*, 2025, pp. 1–6.

[17] V. Metsis, I. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes – which naive bayes?," *CEAS*, 2006.

[18] A. Okunola and A. Ahsun, "Comparative analysis of machine learning models for real-time fraud detection," *ResearchGate*, Jan. 2025.

[19] S. Prakash, B. Kalaiselvi, K. Sivachandar, *et al.*, "Recognizing fake documents by instance-based ML algorithm tuning with neighborhood size," *J. Appl. Data Sci.*, vol. 6, no. 2, pp. 1214–1228, 2025.

[20] L. G. A. Putri, S. A. Wicaksono, and B. Rahayudi, "Analisis klasifikasi spam email menggunakan metode extreme gradient boosting (xgboost)," *J. Pengemb. Teknol. Inf. Ilmu Komput.*, vol. 9, no. 2, 2025.
[21] N. R. Rao and G. A. F. Vinodhini, "Measuring the efficiency of random forest, naive bayes, multilayer perceptron and support vector machine in email spam detection," in *AIP Conf. Proc.*, vol. 3270, p. 020052, 2025.
[22] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian approach to filtering junk e-mail," in *Learning for Text Categorization: Papers from the 1998 Workshop*, vol. 62, pp. 98–105, Madison, WI, USA, 1998.
[23] M. Sharabov, G. Tsochev, V. Gancheva, and A. Tasheva, "Filtering and detection of real-time spam mail based on a Bayesian approach in university networks," *Electronics*, vol. 13, no. 2, p. 374, 2024.
[24] K. Soppari, B. Vangapally, S. S. Sohail, and H. Dubba, "Survey on: Voice driven email solutions for visually impaired people," *World J. Adv. Res. Rev.*, vol. 26, no. 1, pp. 032–036, 2025.
[25] E. H. Tusher, M. A. Ismail, and A. F. M. Raffei, "Email spam classification based on deep learning methods: A review," *Iraqi J. Comput. Sci. Math.*, vol. 6, no. 1, p. 2, 2025.
[26] E. H. Tusher, M. A. Ismail, M. A. Rahman, A. H. Alenezi, and M. Uddin, "Email spam: A comprehensive review of optimize detection methods, challenges, and open research problems," *IEEE Access*, 2024.
[27] H. Yin, W. Zhu, C. Fei, and X. He, "Deep learning for spatiotemporal modeling: A survey," *IEEE Trans. Big Data*, 2017.
[28] T. Yin, W. Ding, H. Ju, J. Huang, and Y. Chen, "The fuzzy hypergraph neural network model based on sparse k-nearest neighborhood granules," *Appl. Soft Comput.*, vol. 170, p. 112721, 2025.
[29] T. Yusnanto, F. Fatkhurrochman, M. A. Muin, and K. Mustofa, "Data security analysis on the use of e-commerce to prevent online fraud," *RIGGS: J. Artif. Intell. Digit. Bus.*, vol. 4, no. 1, pp. 50–55, 2025.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2025.0902/255

Research Article Vol. 9 issue 2 Jul-Dec 2025

Big Data Analytics and Machine Learning Techniques for Real-Time Credit Card Fraud Detection

Sadia Abbas Shah[1*], Rabia Javed[2], Fahima Tahir[3], Khansa Aatif[4], and Wajeeha Malik[5]

[1] School of system and technology, Department of Software Engineering, University of management and technology, Lahore, Pakistan,

[2,3,4,5] Department of Computer Science, Lahore College for Women University, Lahore, Pakistan,

Corresponding Author: sadia.abbas@umt.edu.pk

Received: Sep 15,2025; **Accepted:** Sep 27,2025; **Published:** Oct 16,2025

ABSTRACT

Big Data is commonly characterized by the 4 V's: Volume, Variety, Velocity, and Veracity. In today's digital age, data is generated in terabytes and petabytes, far exceeding the storage capabilities of a single machine. With data constantly circulating across cloud platforms, the risk of leakage and fraud has increased significantly, with credit card fraud being one of the most pressing global concerns. As numerous shopping platforms and businesses operate around us, each domain generates vast amounts of data, often reaching into yottabytes. Manually handling, analyzing, or detecting anomalies in such large-scale data is extremely challenging. However, with the advancement of computing and emerging technologies, detecting fraud has become much more efficient and scalable. This study examines the application of big data in analyzing credit card consumer behavior, specifically in the context of online transactions, password creation, age, income, and other relevant factors. The focus is on identifying anomalies in these data points to detect potentially fraudulent activities quantitative approach is employed to identify statistical patterns, and the performance of seven different machine learning algorithms, such as Logistic Regression, K-Nearest Neighbors (KNN), and XGBoost, is evaluated for their effectiveness. As technology advances, factors such as age and increasing reliance on online transactions, e-commerce, and digital banking contribute to rising vulnerabilities, making fraud detection more critical than ever. In the Real-time credit card fraud detection using big data, different algorithms are discussed and implemented so XGBOOST gives better results with 99% accuracy as another ML Algorithm. The impact of compliance on sophisticated data-based security systems will be examined in a later study, which can make use of historical fraud typologies and trends to comprehend potential shifts over time.

Keywords: Anomaly Detection, Machine Learning Algorithms, Big Data, Credit Card Fraud Detection, XGBoost and KNN

1. INTRODUCTION

Over all the world consumers now utilize the internet more frequently for banking, mobile payments, and purchases, which is convenient but also exposes them to hackers. The existing methods for identifying fraud are no longer effective; new, more effective methods that might work for the new customer type are required [1]. Conventional decision-making usually depends on empirical techniques, a specialist's knowledge, or strict, antiquated algorithms that are unable to adjust to complex patterns or new information in contemporary data. Even while the aforementioned techniques might be effective in some fields, they are frequently limited by human cognitive biases and the sheer difficulty or impossibility of processing such massive amounts of incoming data. Delays in response, significantly increased error rates, and a general inability to handle massive datasets to reveal accurate and important information are all possible outcomes of decisions made using traditional methods rather than machine learning techniques [2]. Consumers benefit from credit cards in addition to debit cards since they protect items that may be lost, stolen, or destroyed. Before using their credit card to make any purchases, customers must confirm the transaction with the retailer [3]. Credit card fraud is detected using a variety of techniques, such as statistical, machine learning, and deep learning methods. To find and examine irregularities in credit card transactions, statistical methods, including regression, hypothesis testing, and clustering are used. Machine learning techniques, on the other hand, use algorithms to analyze previous data and identify fraudulent activities in real time. Neural networks are used in deep learning techniques to automatically find complex patterns and features in large, complicated datasets, leading to incredibly accurate fraud detection [4]. We examine the effectiveness of artificial neural networks (ANN-DL), K-nearest neighbors (KNN), Naïve Bayes classifiers, decision trees, random forest classifiers, and logistic regression. To find the best models for identifying fraudulent transactions, we compare important performance parameters like accuracy, sensitivity, specificity, and F1-score. Furthermore, we investigate how various folds in cross-validation affect model performance, offering information on the classifiers' stability and resilience. This study adds to the continuous efforts to create reliable and effective fraud detection systems, providing insightful information to researchers and financial organizations working to successfully battle credit card theft [5].

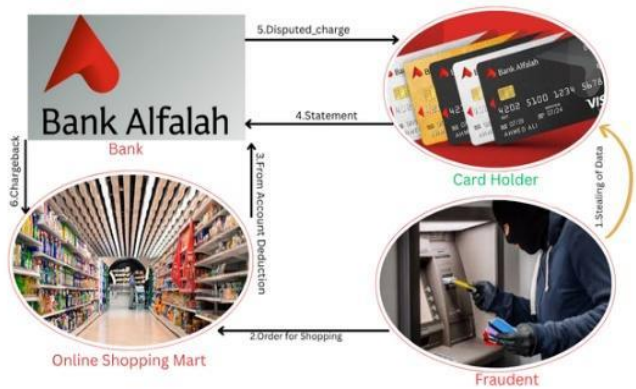


Figure 1. Large-scale real-time credit card fraud

Fig 1 illustrates the process involving a transaction dispute related to fraudulent activity using a credit card issued by Bank Alfalah. Initially, a cardholder engages in an online shopping order, which leads to a deduction from their account. However, the cardholder later receives a statement indicating a disputed charge, prompting them to seek a chargeback from the bank. The flow also highlights the involvement of a fraudulent entity that compromises the cardholder's data, leading to unauthorized transactions. This visualization encapsulates the cycle of online shopping, potential fraud, and the subsequent actions taken by the cardholder and bank to resolve the dispute.

2. RELATED WORK

Machine learning solutions are extremely helpful in various effective spheres in which data have to be processed; one of them is the detection of card fraud. Some of the methods recommended in prior studies have proposed inclusion of methods to detect fraud during the supervised approaches, the unsupervised approaches, and even a hybrid approach; this make sit necessary and important to know some technology in the identification of credit card fraud and understand better the nature of card frauds. Numerous measures were proposed and verified. The following brief will review most of them. Prediction of card fraud has been centered on the interpretation of the card actions during purchase. During the process of identifying card fraud, most of them were put in place, including neural network (NN), genetic algorithm (GA), support vector machine (SVM), frequent itemset mining (FISM), decision tree (DT), optimization algorithm of the migratory birds (MBO) and naive Bayes process (NB). In its performance, the quantitative analysis carried out is an estimation of the logistic regression and naive Bayes analysis alongside assessment of the Bayesian and neural system output on the dataset of credit card fraud [6]. A synopsis of the available works done towards real-time credit card fraud detection is contained in the Table 1.

Kasongo [12] designed a GA-based FS to enhance the performance of ML-based models used in the field of intrusion detection systems. The results demonstrated that the RF classifier performed better when GA was used, with an Area Under the Curve (AUC) of 0.98.

Numerous assessments of earlier studies have been carried out to analyze machine learning applications in detecting fraudulent credit card activity using innovative and stacked architectures [13–16]. Numerous studies have been completed using various data mining approaches; according to reviews [17–21], over 23% of these studies have focused on the SVM methodology, with the naïve Bayes and random forest techniques accounting for 13% of the total number of research papers. Up until now, the research has mostly concentrated on the data. The researchers found that the data was unbalanced, which is likely what caused the models' performance to deteriorate. As a result, they used under-sampling and oversampling, and the under-sampling strategy using logistic regression produced better results [22]. Researchers have successfully experimented with artificial neural networks for fraud detection since they perform well when dealing with complex data [23]. In this paper, additional analysis has been demonstrated to examine more commonly used techniques that may perform better than the earlier findings. After applying 19 resampling techniques to each algorithm, the top three are chosen to be used in the second stag

Table 1. The recent efforts to improve the identification of real-time credit Card Fraud

Reference	Dataset	Method	Pros	Cons	Accuracy
[7]	European card holders	KNN	Mean-squared error is decreased using CFLANN.	KNN takes a lot of time.	97.56% for identifying fraudulent transactions
[8]	Chinese financial institution	Support vector machine, neural network	The CCFD performs better overall.	lengthy procedure	95.20% recall and 99.21% accuracy
[9]	Datasets from financial institutions	SVM, KNN	It facilitates the categorization of real-world interactions.	The algorithm needs in-depth knowledge to make predictions in real-world scenarios.	SVM's 91% accuracy and KNN's 72%
[10]	Banks datasets	SVM	SVM avoids overfitting.	Model training takes a lot of time.	When compared to the hybrid B.P. model, SVM performs well.
[11]	UCSD FICO datasets	VM, KNN, Naive Bayes (NB)	The less significant change has little effect on how the model is implemented.	Model training takes a lot of time. The prediction is not always correct. KNN is susceptible to dataset noise.	SVM achieved 20% accuracy, NB 15%, and KNN 10%.

Our objective is to address three main problems with credit card fraud datasets: substantial class imbalance, inclusion of labelled and unlabeled samples, and increased processing capacity. A variety of supervised and semi-supervised machine learning techniques are used for fraud detection.

3. Materials and Methods

Figure 2 illustrates the proposed workflow for real-time credit card fraud detection is structured in several key stages. The Input Layer involves gathering data from various transaction features, including timestamps and fraud indicators. Following this, the Preprocessing phase applies the feature

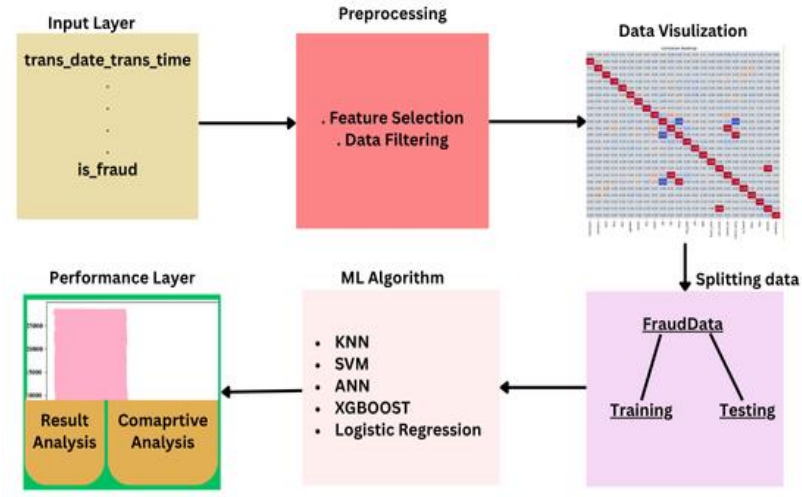


Figure 2. Workflow of Real-Time Credit Card Fraud Detection

Selection and data filtering techniques to refine the dataset, enhancing its quality for analysis. This groundwork is followed by Data Visualization, where the relationships among features are explored using visualization tools to identify patterns and insights. Subsequently, the ML Algorithm step employs various machine learning techniques such as KNN, SVM, ANN, XGBoost, and Logistic Regression to develop models that can predict fraudulent activities. Finally, in the Performance Layer, both result analysis and comparative analysis are conducted to evaluate the effectiveness of the models, as the dataset is split into training and testing subsets for robust performance assessment. **Input Layer** shows the Dataset taken from Kaggle [24] with 22 columns, in which the target column is the 'is fraud' attribute from the Kaggle file with the name of 'fraudTest.csv'. This dataset of simulated credit card transactions includes both authentic and fraudulent transactions. **Data Preprocessing** stage has a big impact on how machine learning models are used later. Some negative data features, like as noise, excessive dimensionality, and outliers, can negatively affect model performance, and many models are unable to handle missing values. Therefore, the dataset is improved in terms of accuracy and completeness by doing data preparation. Figure 3 shows the data handling, filtering, and managing data, calculating statistical values of fraud data implemented with Python on Visual Studio Code. Python is installed on an HP EliteBook equipped with an 11th-generation Intel Core i7 processor and 8 GPU cores

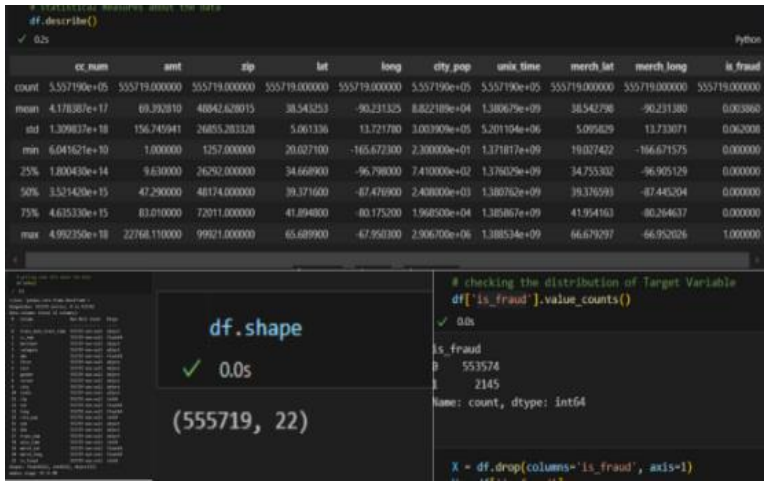


Figure 3. Exploratory Data Analysis for Real-Time Debit Card Fraud Detection

Data Visualization works as a Exploratory Data Analysis (EDA) which is performed on the Fraud Test dataset. Relevant libraries are imported sequentially for data loading, visualization, and normalization. To gain initial insights, we examine the figure 3 shows dataset's column names, shape, and statistical summaries of each feature. Figure 4 shows data distribution as his plot, pair plot and relational plot.

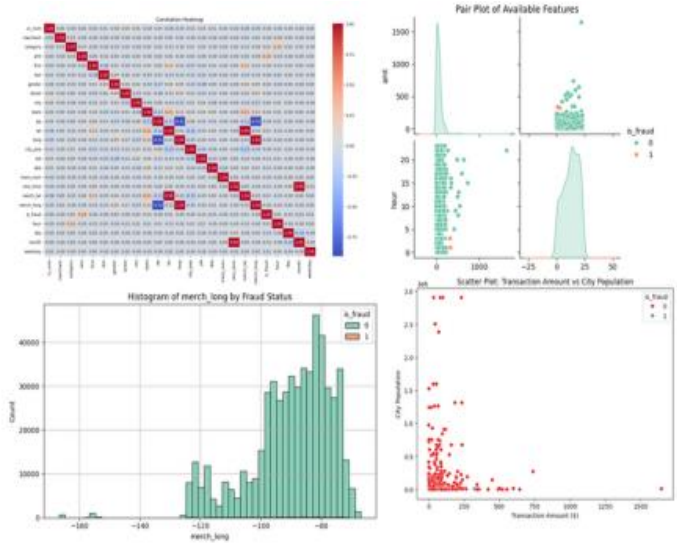


Figure 4. Data Distribution for Real-Time Credit Card Fraud Detection.

4. Results and experiments

The dataset's balanced distribution is depicted in Figure 5. It has been the most crucial issue to address to identify fraudulent behavior. Since there are very few fraudulent situations, any algorithm may assume that any transaction requested from the database will be typical. But in practice, it isn't the case.

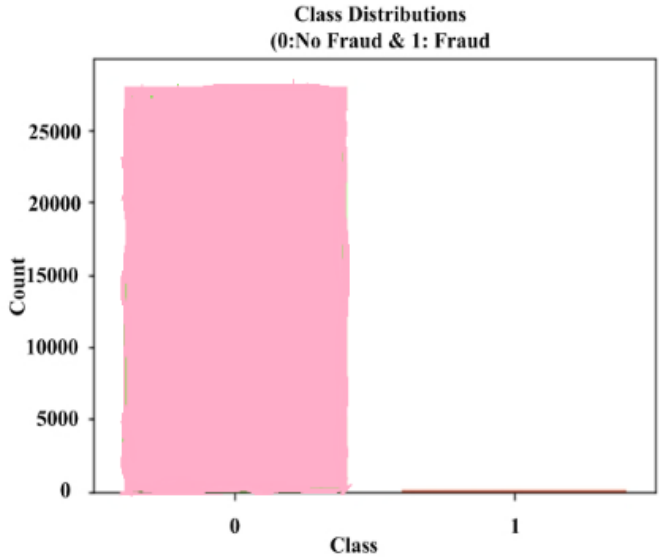


Figure 5. An unbalanced dataset

Table 2. Performance of 7 Techniques for Real Credit Card Fraud Detection

Techniques	**Accuracy**	**Precision**	**MCC**
KNN	0.97	0.97	0.96
SVM	0.97	0.77	0.53
Logistic Regression	0.98	0.88	0.68
XGBoost	0.99	0.98	0.97
ANN	0.93	0.97	0.86
Randomforest	0.96	0.89	0.83
DecisionTree	0.96	0.94	0.84

In the provided Table 2, the performance of seven techniques for real credit card fraud detection is evaluated based on three metrics: Accuracy, Precision, and MCC (Matthews Correlation Coefficient). The KNN (K-Nearest Neighbors) model demonstrates the highest overall effectiveness with an accuracy of 0.99, precision of 0.97, and MCC of 0.96, indicating its reliability in correctly identifying fraudulent transactions. Following closely is the SVM (Support Vector Machine) with an accuracy of 0.9972, though it shows a relatively lower precision of 0.77 and MCC of 0.53, suggesting issues with false positives. Logistic Regression and XGBoost exhibit comparable performance, with Logistic Regression achieving 0.991 accuracy and 0.88 precision, while XGBoost scores an accuracy of 0.99 and precision of 0.98. The ANN (Artificial Neural Networks) technique yields slightly lower results with an accuracy of 0.93 and a precision of 0.97. The Random Forest model shows an accuracy of 0.96 and a precision of 0.89, while the Decision Tree model has the same accuracy of 0.96 but a higher precision at 0.94. This comparison highlights varying levels of effectiveness among the techniques, with KNN and XGBoost emerging as the strongest contenders for accurately detecting fraudulent activities in credit card transactions.

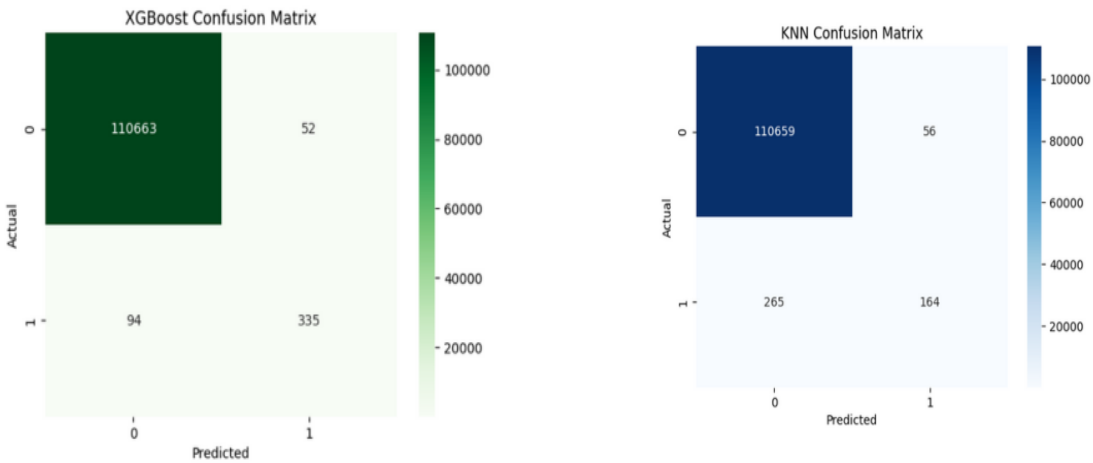


Figure 6. Confusion Matrix for Real-Time Credit Card Fraud Detection

Figure 6. displays comparison confusion matrices for two machine learning models: XGBoost and K-Nearest Neighbors (KNN). Each matrix quantifies the performance of its respective model by categorizing actual and predicted classifications into four distinct areas: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Comparatively, both models exhibit high true positive rates, indicating that they effectively

identify positive cases. However, XGBoost demonstrates slightly better performance, with fewer false positives and false negatives than KNN. This suggests that while both models are competent, the XGBoost model may provide more accurate predictions in this context.

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

$$ACCURACY = \frac{TN+TP}{TN+TP+FN+FP} \quad (2)$$

$$MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (3)$$

5. CONCLUSION

Frequent occurrences of credit card fraud led to significant financial losses. Online credit card transactions account for a significant portion of the vastly increasing number of transactions that take place online. As a result, banks and other financial organizations provide credit card fraud detection software that is highly valued and in high demand. Transactions that are fraudulent can take many different forms and fall under a variety of headings. This essay focuses on four primary instances of fraud in actual transactions. A number of machine learning models are used to tackle each scam, and an evaluation is used to determine which approach works best. This assessment offers a thorough manual for choosing the best algorithm based on the kind of frauds, and we use a suitable performance metric to demonstrate the assessment. Another issue that concerns us in our research is real time credit card fraud detection. With such a question as to whether a particular transaction is genuine or is fraudulent, we resort to predictive analytics that is undertaken by the machine learning models that have been implemented along with an API module. We assess as well a new method which effectively addresses the skewed distribution of the data. A financial institution supplied us with the data used in our experiments under a confidential disclosure accord. KNN and XGBoost are the two most suitable to detect the fraudulentness in real-time, specifically in credit cards, where there is some fraud conduct.

6. REFERENCES

[1] F. T. Zohora, R. Parveen, A. Nishan, M. R. Haque, and S. Rahman, "Optimizing credit card security using consumer behavior data: A big data and machine learning approach to fraud detection," *Frontline Marketing Management Economics Journal*, vol. 4, no. 12, pp. 26–60, 2024.

[2] L. Theodorakopoulos, A. Theodoropoulou, F. Zakka, and C. Halkiopoulos, "Credit card fraud detection with machine learning and big data analytics: A PySpark framework implementation," 2024.

[3] Y. Zou and D. Cheng, "Effective high-order graph representation learning for credit card fraud detection," *arXiv preprint* arXiv:2503.01556, 2025.

[4] A. R. Khalid *et al.*, "Enhancing credit card fraud detection: an ensemble machine learning approach," *Big Data and Cognitive Computing*, vol. 8, no. 1, p. 6, 2024.

[5] S. F. Farabi *et al.*, "Enhancing credit card fraud detection: A comprehensive study of machine learning algorithms and performance evaluation," *Journal of Business and Management Studies*, vol. 6, no. 3, pp. 252–259, 2024.

[6] M. Yousif *et al.*, "Machine learning-based suicide risk assessment and intervention strategies for depression," 2024.

[7] Y. Wang, "A data balancing and ensemble learning approach for credit card fraud

detection," in *Proc. 4th Int. Symp. Computer Applications and Information Technology (ISCAIT)*, IEEE, Mar. 2025, pp. 386–390.

[8] L. Bonde and A. K. Bichanga, "Improving credit card fraud detection with ensemble deep learning-based models: A hybrid approach using SMOTE-ENN," *J. Comput. Theor. Appl.*, vol. 2, no. 3, pp. 383–394, 2025.

[9] E. F. Aminu *et al.*, "A hybridized SMOTE-ENN approach on imbalanced dataset of fraudulent credit-card scenario," *i-Manager's Journal on Data Science & Big Data Analytics (JDS)*, vol. 3, no. 1, 2025.

[10] M. Zhu, Y. Zhang, Y. Gong, C. Xu, and Y. Xiang, "Enhancing credit card fraud detection: A neural network and SMOTE integrated approach," *arXiv preprint* arXiv:2405.00026, 2024.

[11] C. Yu *et al.*, "Credit card fraud detection using advanced transformer model," in *Proc. IEEE Int. Conf. Metaverse Computing, Networking, and Applications (MetaCom)*, Aug. 2024, pp. 343–350.

[12] P. Chatterjee, D. Das, and D. B. Rawat, "Digital twin for credit card fraud detection: Opportunities, challenges, and fraud detection advancements," *Future Generation Computer Systems*, vol. 158, pp. 410–426, 2024.

[13] K. G. Dastidar, O. Caelen, and M. Granitzer, "Machine learning methods for credit card fraud detection: A survey," *IEEE Access*, 2024.

[14] F. D. Data, C. S. Subrahmanyam, N. Deshai, and K. S. J. T. Rajesh, "Algorithms in advanced artificial intelligence," in *Algorithms in Advanced Artificial Intelligence: ICAAAI-2023*, R. N. V. Jagan Mohan et al., Eds., London, U.K.: Taylor & Francis Group, 2024, p. 456.

[15] F. D. Data, C. S. Subrahmanyam, N. Deshai, and K. S. J. T. Rajesh, "Algorithms in advanced artificial intelligence," in *Algorithms in Advanced Artificial Intelligence: ICAAAI-2023*, R. N. V. Jagan Mohan et al., Eds., London, U.K.: Taylor & Francis Group, 2024, p. 456.

[16] J. Liu, X. Zhang, and H. Xiong, "Credit risk prediction based on causal machine learning: Bayesian network learning, default inference, and interpretation," *Journal of Forecasting*, vol. 43, no. 5, pp. 1625–1660, 2024.

[17] Z. Wang, Q. Shen, S. Bi, and C. Fu, "AI empowers data mining models for financial fraud detection and prevention systems," *Procedia Computer Science*, vol. 243, pp. 891–899, 2024.

[18] H. Chen *et al.*, "The role of blockchain in finance beyond cryptocurrency: trust, data management, and automation," *IEEE Access*, vol. 12, pp. 64861–64885, 2024.

[19] I. Y. Hafez *et al.*, "A systematic review of AI-enhanced techniques in credit card fraud detection," *Journal of Big Data*, vol. 12, no. 1, p. 6, 2025.

[20] T. A. Gaav, H. U. Adoga, and T. Moses, "Recent advances in credit card fraud detection: An analytical review of frameworks, methodologies, datasets, and challenges," *J. Future Artif. Intell. Technol.*, vol. 2, no. 3, pp. 343–369, 2025.

[21] F. Moradi, M. Tarif, and M. Homaei, "A systematic review of machine learning in credit card fraud detection," *MDPI Preprints*, 2025.

[22] L. Theodorakopoulos, A. Theodoropoulou, A. Tsimakis, and C. Halkiopoulos, "Big data-driven distributed machine learning for scalable credit card fraud detection using PySpark, XGBoost, and CatBoost," *Electronics*, vol. 14, no. 9, p. 1754, 2025.

[23] J. Wang, J. Liu, W. Zheng, and Y. Ge, "Temporal heterogeneous graph contrastive learning for fraud detection in credit card transactions," *IEEE Access*, 2025

[24] K. Shenoy, "Credit card transactions fraud detection dataset," *Kaggle*, 2020.

International Journal for
Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2025.0902/255

Research Article Vol. 9 issue 2 Jul-Dec 2025

Securing 5G Network Infrastructure Against DDoS Attacks Using ML-Based Anomaly Detection

Shahzaib Hassan[1], Alishba Tabassum[1], Lubna Nadeem[1], Yasar Amin[1], Tariq Mahmood[23]

[1]Department of Telecommunication Engineering, University of Engineering and Technology, Taxila, 47050, Pakistan,

[2]Department of Information Science, University of Education, Lahore, Pakistan,

[3]Artificial Intelligence and Data Analytics (AIDA) Lab, CCIS Prince Sultan University, Riyadh, 11586, Kingdom of Saudi Arabia

Corresponding Author: lubna.nadeem@uettaxila.edu.pk

Received: Sep 18,2025; **Accepted:** Sep 30,2025; **Published:** Oct 17,2025

ABSTRACT

Today, millions of people and devices use the Internet to carry out daily activities, but the growing reliance on the Internet comes with major security concerns. Older security systems and traditional detection techniques are out of date because attackers continue to find new and smarter ways of penetrating networks. They are just not precise enough to stay in the race. This research discusses how that gap can be filled by machine learning (ML). Although in cybersecurity, ML has demonstrated potential, accuracy remains reliant on the selection of the appropriate models and the concentration on the most important parts of the data. Although ML has already shown its potential, our work aims at refining the approach to increase detection accuracy. The most promising among the techniques tested was the Random Forest (RF) algorithm, which had an impressive accuracy rate of 99.84%. This clearly indicates that our proposed system is far better than the previous methods, showing its capability to detect malicious activities.

Keywords: 5G networks, security threat, distributed denial of service (DDoS), machine learning

1. INTRODUCTION

Internet networks continue to expand globally due to technological advances that include smartphones, computers, communication systems, and IoT devices [1]. Research indicates that there exist more than 5 billion smart devices worldwide, while 3 billion users actively use the internet [2]. The widespread use of Internet networks produces enormous amounts of data second by second, which presents major challenges in protecting information against cyber threats [3]. Computer systems and their networks are dependent on cybersecurity to protect them from unauthorized access [4]. Data protection, along with privacy assurance, functions as a fundamental structure that protects organizations and states as well as individual users. Data transmitted on the Internet remains exposed to hacking and manipulation attempts by cybercriminals [5]. The 2017 cyberattack damages reached $5 billion, and analysts predict that this amount will increase to $6 trillion yearly starting in 2021 [6]. Distributed denial-of-service (DDoS) attacks represent one of the most common cybersecurity threats that cause servers and networks to crash when flooded with excessive data packets [7]. The number of distributed denial-of-service at tacks has increased significantly in recent times. A major DDoS attack on Amazon Web Services' (AWS) Amazon Simple Storage Service (S3) and other platforms generated a severe service disruption in February 2020, which lasted approximately eight hours [8]. The recorded attack, which stood out as one of the largest, reached its peak performance level at 2.3 terabytes per second. The research reported by Security Week shows that DDoS attacks occur 28,700 times a day on the Internet [9]. The rising need for strong cybersecurity systems able to detect cyber-attacks effectively drives the current market demand. The goal of cybersecurity professionals is to create IDS (Intrusion Detection Systems) that detect known threats and new attacks without producing false alerts [10]. Modern cyber-attacks especially DDoS attacks require intelligent detection methods because multiple existing intrusion detection approaches exist. Modern intrusion attempts have rendered traditional IDS solutions ineffective, according to research [11]. Artificial intelligence techniques have become necessary for cybersecurity practice and have achieved great success in all fields. The practice has proven successful in all areas since it became mandatory. The capability of big data exploration 1, through hidden models reaches tremendous heights because of their ability to discover patterns in data. Through ML techniques, organizations can detect and monitor network-based attacks [12]. Various studies used different ML techniques for intrusion detection. Some deficiencies remain in this approach, including the determination process. The re searchers have chosen basic and effective features to enhance the performance of ML techniques [13].

A. CONTRIBUTION

The main contributions are as follows.

- To build an innovative detection framework that will detect DDoS attacks accurately.
- Select important features that would boost the accuracy and efficiency of DDoS attack detection.
- Testing and comparing different machine learning models to see which could detect threats with the highest accuracy.
- Comparison of different machine learning models to see which one could detect threats more accurately.
- Selecting the optimal model, which can then be tested using open-source datasets through standard performance measurements.

2. PAPER ORGANIZATION

Section III discusses the Literature Survey, and Section IV depicts the Gaps in research and the Motivation of our work. Then Section V is about Proposed Methodology. Section VI explains Key Performance Indicators. Section VII provides details about Dataset. Then Section VIII is about our main research contributions. Section IX explains in detail the proposed system model. Further, Section X demonstrates the Workflow of our research. Section XI is related to simulation results and discussion. Finally, Section XII is on

Conclusion and Future Recommendations.

3. LITERATURE SURVEY

The rapid technological advancements and widespread Inter net of Things (IoT) devices have created a situation where people increasingly depend on internet networks, so robust security must protect user privacy and data. Relevant research shows that artificial intelligence presents itself as an efficient method for handling cybersecurity threats. Research through multiple studies has investigated the usage of Intrusion Detection Systems (IDS) that employ both Machine Learning (ML) and Deep Learning (DL) methods. This section discusses multiple research approaches that use ML and DL methods to identify cyber-attacks. The research team of Bindra and Sood evaluated six ML techniques including Logistic Regression (LR), K-Nearest Neighbors (KNN), Random Forest (RF), Naïve Bayes (NB), Linear Support Vector Machine (SVM) along with Linear Discriminant Analysis (LDA) to identify the best method for DDoS attack detection [14]. RF demonstrated the best accuracy rate of 96.5% according to tests conducted on the CIC IDS dataset which outranked all other analysis methods. Chavan et al. evaluated DDoS attack detection by analyzing KNN, SVM, Decision Tree (DT), and LR as four ML techniques in their work [15]. The LR model showed superior accuracy results by reaching 90.4%. Combined methods used in decision-making produce better accuracy levels than single classification systems. Das, Saikat along with coauthors developed an ensemble model which unites Multilayer Perceptron (MLP) with SVM and KNN and DT as base ML techniques [16]. The researchers performed experiments on the NSL-KDD dataset, which demonstrated that their ensemble classifier achieved superior results than standalone models in the study. The Auto Encoding (AE) method proposed by Kasim serves both to reduce features and boost traffic classification efficiency [17]. Kasim proposed the application of Auto-Encoding (AE) for both feature selection and dimension reduction to achieve traffic classification. AE methods help reduce dimensions for effective traffic classification according to [18]. The researchers conducted performance tests using both CICIDS2017 and NSL-KDD data sets. The model achieved successful classification results in testing datasets according to the performance studies. Bhardwaj et al. [19] presented a method that merges well-stacked sparse AE. The approach uses Deep Neural Networks together with Deep Neural Network (DNN) feature learning to detect possible DDoS attacks. Highly efficient DL techniques discovered big data thanks to their exceptional discovery capabilities. Multiple groups have made systematic attempts to utilize this topic for cybersecurity research. Al-Emadi et al. [20] analyzed how DL techniques function within CNN and RNN systems for network intrusion detection. Table 1 shows the comparison of the proposed work with existing works.

4. RESEARCH GAPS &MOTIVATION

As 5G networks develop, telecommunications have reached new levels of speed, responsibility, and connectivity. Such improvements are used to facilitate critical services and extensive device communication. This improvement also en tails an increase in security threats, especially DDoS attacks, that can hamper vital operations. Creating secure 5G systems is more relevant than ever, and increases the necessity for smarter, adaptive security solutions that will ensure reliable and uninterrupted connectivity. The existing DDoS detection techniques can't effectively deal with the scale and complex ity of a 5Genvironment. They are usually not flexible enough to cope with rapidly changing patterns of traffic and advanced threats. To fill this gap, our research suggests using a machine learning–based detection framework. It focuses on feature se lection to improve detection accuracy, tries different models, and determines the most successful approach to employment based on open datasets and conventional evaluation metrics.

5. PROPOSED METHODOLOGY

The proposed work develops an effective network intrusion detection system by combining ML

algorithms with feature selection strategies. The study conducts performance assessments on the intrusion detection of four ML methods RF, KNN, SVM, and DT. The system utilizes feature selection techniques to detect important features.

Table 1. Comparison of Proposed work with Existing works

Study	Year	Model	Dataset	Best ML Accuracy
[18]	2021	CNN, LSTM, and CLSTMNet	NSL-KDD	CLSTMNet (99.28%)
[14]	2020	CNN and RNN	NSL-KDD	CNN (97.01%)
[10]	2019	RF, LR, NB, KNN, Linear SVM, and LDA	CIC IDS	RF -96.50%
[15]	2020	AE+ SVM	CICIDS2017 & NSL-KDD	AE+ SVM (96.36%)
[6]	2020	CNN	NSL-KDD	CNN -99.30%
[5]	2019	Ensemble model, MLP, SVM, KNN, and DT	NSL-KDD	Ensemble model -99.10%
Proposed Work	2025	RF, KNN, SVM, and DT	NSL-KDD	RF -99.84%

Table 2. NSL-KDD Dataset Features

Index	Feature	Index	Feature
1	src_bytes	22	dst_host_same_src_port_rate
2	dst_host_srv_count	23	same_srv_rate
3	num_access_files	24	dst_host_count
4	logged_in	25	dst_bytes
5	serror_rate	26	dst_host_srv_serror_rate
6	su_attempted	27	srv_rerror_rate
7	num_access_files	28	num_file_creations
8	root_shell	29	num_compromised
9	is_host_login	30	protocol_type
10	count	31	num_shells
11	duration	32	diff_srv_rate
12	srv_serror_rate	33	dst_host_srv_sror_rate
13	num_root	34	srv_count
14	land	35	service
15	dst_host_diff_srv_rate	36	urgent
16	wrong_fragment	37	hot
17	is_guest_login	38	dst_host_srv_count
18	serror_rate	39	flag
19	num_failed_logins	40	class
20	rerror_rate	41	num_outbound_cmds
21	dst_host_same_srv_rate	42	srv_diff_host_rate

5.1. DATASET

The research utilized NSL-KDD dataset because it represents a clean and refined version. The KDD data set received the traffic information from which it was built. The NSL KDD data set contains 148,517 structured samples with 42 characteristics (Table 2 provides their list)

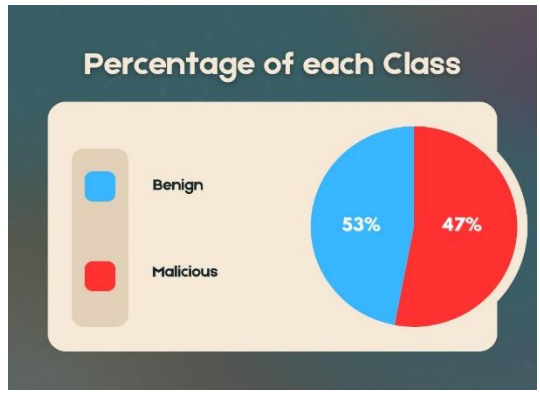


Figure 1. Pie Chart of Classification

5.2. TRADITIONAL METHOD USED FORANOMALY DETECTION

5.2.1 Decision Tree (DT)

Decision Trees represent one of the most applied non parametric supervised learning methods, which functions for both classification and regression tasks. The system arranges itself into a branching pattern that extends from the root node through decisions based on established rules.

5.2.2. Random Forest (RF)

The supervised learning method Random Forest creates various decision trees as an ensemble approach to maximize both regression and classification success rates. Combining multiple trees into one model improves overall model performance.

5.2.3. Support Vector Machine (SVM)

SVM operates as a supervised machine learning model that mainly provides classification functions. SVM operates through identifying the optimum division (or hyperplane) that distinguifies different classification categories in a data set.

5.2.4. K-Nearest Neighbors (KNN)

The simple KNN algorithm serves as a solution for classification and regression problems.

6. PROPOSED MODEL

This model diagram depicts the DDoS attack structure within IoT-enabled 5G networks. The architecture implements several connected layers beginning with the IoT gateway and extending through the backhaul network as well as operator network cloud and external network services that face DDoS attack risks which lead to operational disruptions.

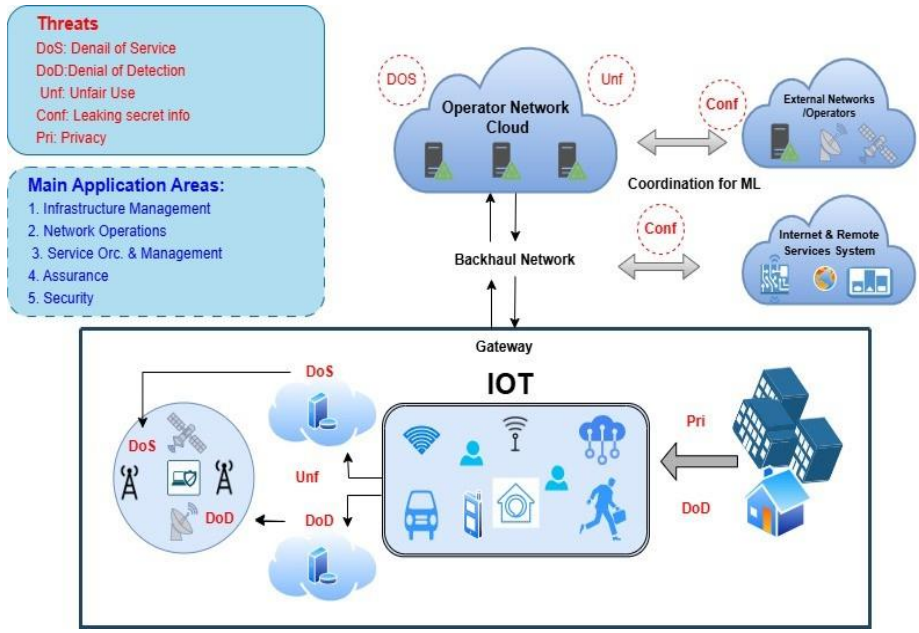


Figure 2. System Model

6.1. DDOS ATTACK THREATS IN IOT-5G NETWORKS

A Distributed Denial of Service attack uses network re- sources to an overwhelming point thus preventing legitimate users from accessing services. The model demonstrates the essential locations which DDoS attacks launch from and spread between.

1) **IoT Devices & Gateway:** Unsecure IoT devices allow attackers to create botnets which send large amounts of traffic to IoT gateways thus causing service degradation.
2) **Backhaul Network:** The traffic overload traverses through the operator network cloud until it reaches its maximum processing capacity and bandwidth thresh- old.
3) **Operator Network Cloud:** The cloud infrastructure in Operator Network Cloud suffers from network congestion along with service unavailability triggered by malicious requests.
4) **External Networks & Remote Services:** Cyber at- tackers can strike external networks and remote ser- vices to cause extensive service breakdowns among linked systems.

6.2. DDOS ATTACK PROPAGATION & IMPACT

The propagation of DDoS attacks happens through com- promised IoT nodes that serve as botnet members to send excessive request floods. The impact includes:

1) Excessive traffic on gateway servers as well as cloud systems result in both longer communication response times and data transmission failures.
2) The excessive request flow during DDoS attacks uses up all available CPU power together with memory capacity and network bandwidth thus blocking access to services.

3) Iot attacks against critical infrastructure result in permanent breakdowns of medical care delivery and residential automation systems and industrial automation networks.

6.3. WORKFLOW

1) **Dataset Selection & Preparation:** We used the NSL- KDD dataset, which is well-suited for detecting DDoS attacks due to its labeled network traffic data.
2) **Feature Selection:** From the dataset, we extracted the most relevant features that significantly contribute to DDoS detection.
3) **Algorithm Selection:** We plan to use machine learning algorithms like DT, SVM, RF, and KNN. These algorithms are selected based on their proven effectiveness in detecting DDoS attacks.
4) **Model Training & Validation:** We split the dataset into training and testing sets. We trained the models using Python libraries like Scikit-learn, ensuring their reliability through cross-validation techniques.
5) **Evaluation Metrics:** To evaluate model performance, we used metrics like accuracy, precision, recall, F1- score, and specificity. A comparison of these results helped us select the best algorithm for real-time application.

6.4. KEY PERFORMANCE INDICATORS

Five quality measures evaluated the performance of ML techniques for evaluation purposes. The evaluation of ML techniques depends on five measures including Accuracy, Precision, Sensitivity, Specificity and F1-Score. The classification value of '1' represents positive outcomes in this analysis. While benign samples are considered negative and represented by '0'. All performance measure formulas are represented below:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = 2 \times (\frac{Precision \times Recall}{Precision + Recall})$$

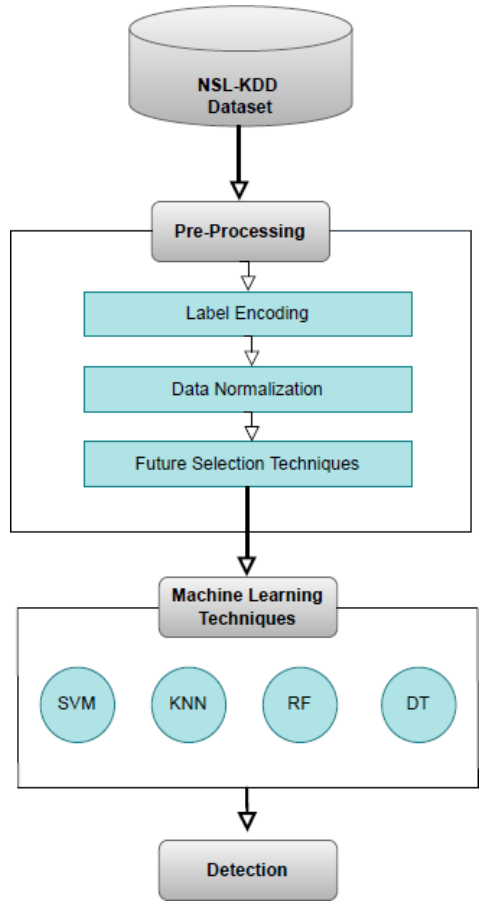


Figure 3. System WorkFlow

TP	**FN**
FP	**TN**

Figure 4. Parameters Table

- TP refers to cases where malicious samples receive proper detection as malicious.
- The detection of benign samples as benign falls under True Negatives category.
- The system identifies benign samples incorrectly as malicious through its FP output.
- The detection system marks malicious samples as benign when they are already identified as harmful.

7. RESULTS AND DISCUSSION

7.1 PERFORMANCE EVALUATION

8. The four Machine Learning (ML) techniques (RF, DT, KNN, and SVM) exist within Scikit Learn library which functions as a robust Python library for implementing ML development. One of the most potent libraries for building and

implementing ML techniques is Scikit Learn.

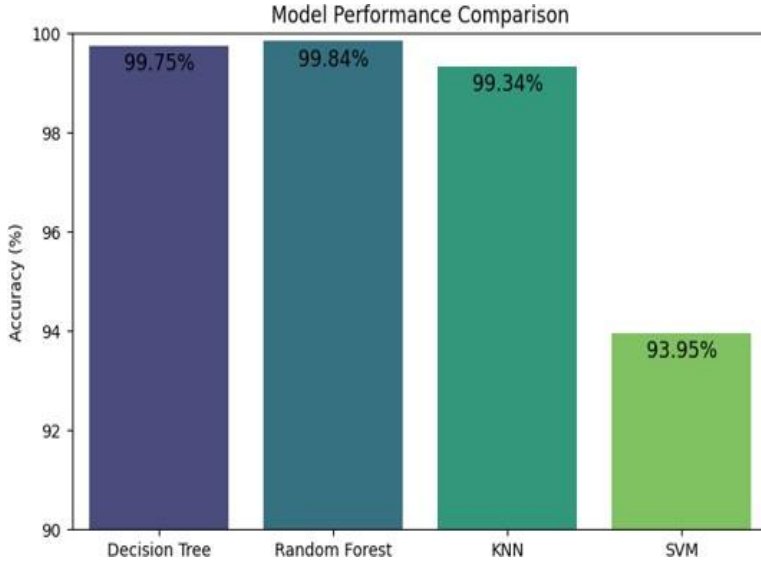


Figure 5. Accuracy Result

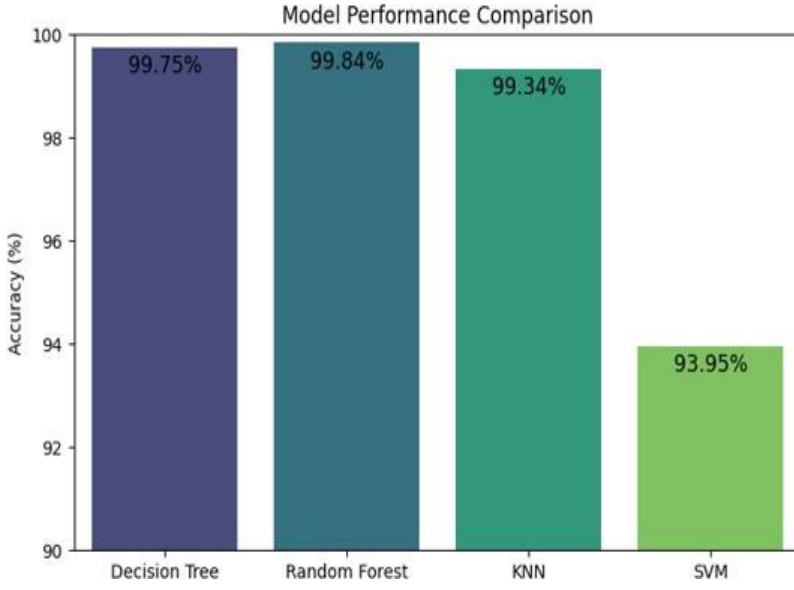


Figure 6. Other Parameters Result

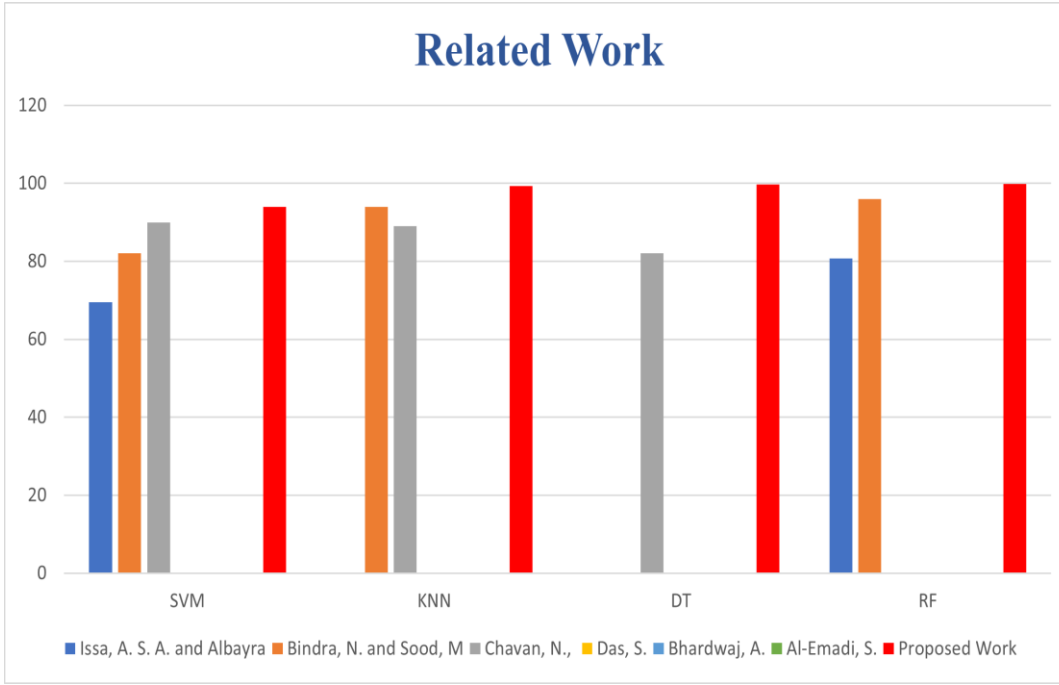


Figure 7. Related Work

Figure 7 depicts in detail the comparison of different ML techniques with the proposed work. It is observed that the accuracy of the proposed technique is better than all the existing techniques in literature.

The performance metrics of the four ML techniques are presented in Table 3.

Table 3. Results

Technique	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Support Vector Machine	93.95%	94.50%	94.50%	94.50%
Random Forest	99.84%	99.90%	99.90%	99.90%
K-Nearest Neighbors	99.34%	99.00%	99.00%	99.00%
Decision Tree	99.75%	99.90%	99.90%	99.90%

The research dataset split its content into 80% training material and 20% testing material. A comparison of the four techniques occurred on just twenty-five available features. The selected dataset features underwent feature selection before usage in this study. The RF method demonstrated in Figure 5 the best accuracy performance among all techniques. 99.84%, superior to the rest of the techniques. The KNN technique obtained accuracy results of 99.34% while the second-best accuracy rate went to the DT technique with 99.75%. The accuracy rate for SVM technique came in as the lowest at 93.95% but the RF technique led with 99.84%. Both the DT technique and KNN technique performed with respective accuracies of 99.75% and 99.34%.

Figure 6 demonstrates the comparison of ML techniques with the other parameters like Precision, Recall F1 score. The confusion matrix of the four ML techniques also appears in Figure 8 to Figure 11. Each confusion matrix represents model's performance regarding the separation of various classes. The application of pre-processing techniques alongside feature selection methods creates necessary steps for implementing ML methods. A model demonstrates superior data preprocessing implementation. The performance accuracy would rise after conducting critical feature testing and pre- processing. Figure 8 illustrates that Random Forest (RF) model had correctly assigned most of the samples with few misclassifications. It has strong class prediction abilities of all classes. From figure 9 it can be observed that the Decision Tree (DT) model was good; however, it committed a bit of errors when compared with the Random Forest (RF) especially on differentiating similar threat types. From Figure 10, the K- Nearest Neighbors (KNN) model had difficulties with class boundaries and thus it performed worse than other models with more misclassifications. It is more sensitive to data imbalance. In Figure 11, the Support Vector Machine (SVM) had good performance though some of the classes were not well separated which was as a result of overlapping feature distributions.

The research utilized RF technique which produced superior results than other RF studies. The RF technique employed here exceeded previous applications of RF technique in other research studies. In addition, the KNN, SVM, and DT. The KNN along with SVM and DT techniques from this study achieved superior performance results compared to other KNN and SVM and DT techniques. An RF model which utilized feature selection methods achieved the recommendations. The recommended model based on feature selection techniques and RF classifications shows promising and reliable future performance.The performance of this model exceeded every measurement in this study. The pro- posed model study draws data from the mentioned research papers.

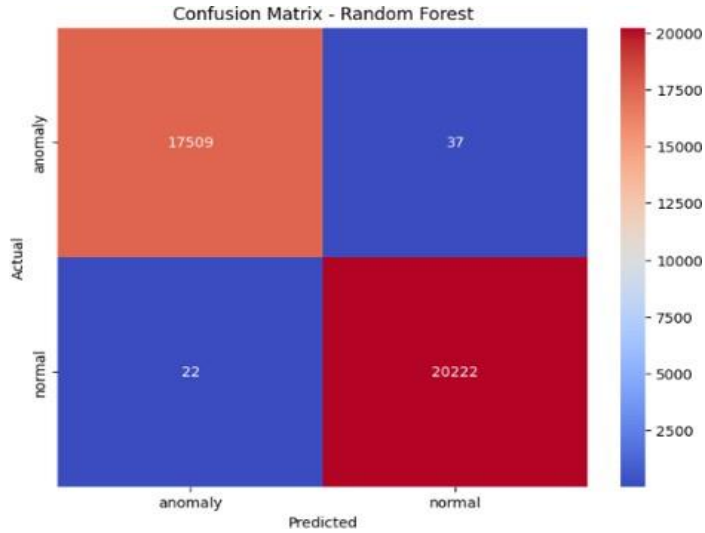


Figure 8. Confusion Matrix of RF

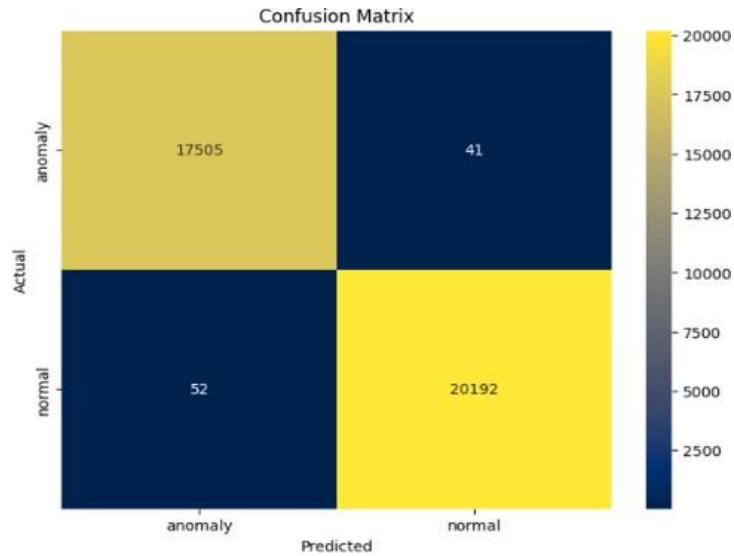


Figure 9. Confusion Matrix of DT

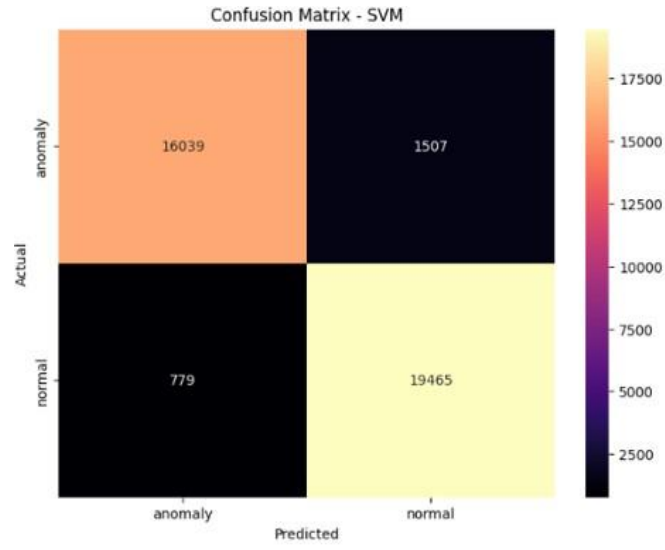


Figure 10. Confusion Matrix of SVM

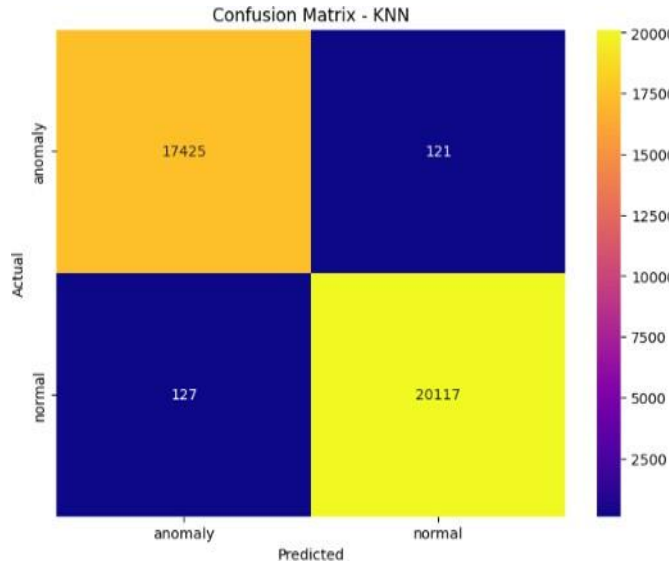


Figure 11. Confusion Matrix of DT

7.2. FUTURE RECOMMENDATIONS

Future work needs to employ different machine learning methods to enhance detection accuracy together with error reduction. Real-time detection technology protected by fast response systems serves as an essential measure to enhance 5G network protection. Deep learning techniques

combined with novel methods of choosing detection features would improve threats detection capabilities. New security systems which present the capability to broadcast threat information between different networks will result in better overall security responses.

9. CONCLUSION

The detection of DDoS attacks in 5G networks becomes effective through the utilization of Machine Learning (ML) algorithms that include Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Tree (DT). Random Forest yielded the best accuracy rate at 99.84% among the applied Machine Learning algorithms which demonstrates the potential of ML for network protection. The combination of ML algorithms with feature selection methods delivers better detection results by converting traditional IDS shortcomings into effective solutions when it comes to highly complex 5G cyberattacks.

10. REFERENCES

[1] D. Dasgupta, Z. Akhtar, and S. Sen, "Machine learning in cybersecurity: A comprehensive survey," *The Journal of Defense Modeling and Simulation*, vol. 19, no. 1, pp. 57–106, 2022.

[2] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[3] A. Salih, S. T. Zeebaree, S. Ameen, A. Alkhyyat, and H. M. Shukur, "A survey on the role of artificial intelligence, machine learning and deep learning for cybersecurity attack detection," in *Proc. 7th Int. Eng. Conf. "Research & Innovation amid Global Pandemic" (IEC)*, 2021, pp. 61–66.

[4] S. R. Zeebaree, K. Jacksi, and R. R. Zebari, "Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 1, pp. 510–517, 2020.

[5] I. Avci and M. Koca, "Cybersecurity attack detection model using machine learning techniques," *Acta Polytechnica Hungarica*, vol. 20, no. 7, pp. 29–44, 2023.

[6] W. Tong, L. Lu, Z. Li, J. Lin, and X. Jin, "A survey on intrusion detection system for advanced metering infrastructure," in *Proc. 2016 6th Int. Conf. on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, 2016, pp. 33–37.

[7] R. Cohen, "Cloud attack: Economic denial of sustainability (EDoS)," 2009. [Online].

[8] C. Cimpanu, "AWS said it mitigated a 2.3 Tbps DDoS attack, the largest ever," *ZDNet*, 2020. [Online].

[9] J. Reo, "Academic research reports nearly 30,000 DoS attacks per day," 2021. [Online].

[10] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine learning and deep learning approaches for cybersecurity: A review," *IEEE Access*, vol. 10, pp. 19 572–19 585, 2022.

[11] A. S. A. Issa and Z. Albayrak, "CLSTMNet: A deep learning model for intrusion detection," *Journal of Physics: Conference Series*, vol. 1973, no. 1, p. 012244, 2021.

[12] M. Abdullahi, Y. Baashar, H. Alhussian, A. Alwadain, N. Aziz, L. F. Capretz, and S. J. Abdulkadir, "Detecting cybersecurity attacks in Internet of Things using artificial intelligence methods: A systematic literature review," *Electronics*, vol. 11, no. 2, p. 198, 2022.

[13] A. A. Salih and M. B. Abdulrazaq, "Combining best features selection using three classifiers in intrusion detection system," in *Proc. 2019 Int. Conf. on Advanced Science and Engineering (ICOASE)*, 2019, pp. 94–99.

[14] Kaggle, "NSL-KDD Dataset," 2019. [Online]. Available: https://www.kaggle.com/datasets/hassan06/n

slkdd. Accessed: Dec. 16, 2019.

[15] N. Bindra and M. S. Sood, "Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset," *Automatic Control and Computer Sciences*, vol. 53, no. 5, pp. 419–428, 2019.

[16] N. Chavan, M. Kukreja, G. Jagwani, N. Nishad, and N. Deb, "DDoS attack detection and botnet prevention using machine learning," in *Proc. 2022 8th Int. Conf. on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, 2022, pp. 1159–1163.

[17] S. Das, A. M. Mahfouz, D. Venugopal, and S. Shiva, "DDoS intrusion detection through machine learning ensemble," in *Proc. 2019 IEEE 19th Int. Conf. on Software Quality, Reliability and Security Companion (QRS-C)*, 2019, pp. 471–477.

[18] Ö. Kasim, "An efficient and robust deep learning-based network anomaly detection against distributed denial of service attacks," *Computer Networks*, vol. 180, p. 107390, 2020.

[19] A. Bhardwaj, V. Mangat, and R. Vig, "Hyperband tuned deep neural network with well-posed stacked sparse autoencoder for detection of DDoS attacks in cloud," *IEEE Access*, vol. 8, pp. 181 916–181 929, 2020.

[20] S. Al-Emadi, A. Al-Mohannadi, and F. Al-Senaid, "Using deep learning techniques for network intrusion detection," in *Proc. 2020 IEEE Int. Conf. on Informatics, IoT, and Enabling Technologies (ICIoT)*, 2020, pp. 171–176.

[21]

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2025.0902/256

Research Article Vol. 9 issue 2 Jul-Dec 2025

Efficient Blind Multi-Receiver Signcryption of Secure Multicast in IoT and Beyond.

Nizam ud Din[1], Zahid Mahmood[2], Muhammad Yasir Shabir[3], Asif Kabir[2], [4]Kusar Perveen

[1]Department of Computer Science, University of Chitral, Pakistan,

[2]Department of CS&IT University of Kotli Azad Jammu & Kashmir, Pakistan.

[3]Department of Computer Science, University of Turin, Italy.

[4]Department of Computer Sciences, National College of Business Administration & Economics, Lahore, Pakistan

Corresponding Author: zahidmahmood575@uokajk.edu.pk

Received: Jul 20,2025; **Accepted:** Aug 12,2025; **Published:** Oct 28,2025

ABSTRACT

This research introduced Blind Multi-Receiver Signcryption (BMRSC) scheme that is designed upon Elliptic Curve Cryptography (ECC) to improve security and privacy in netwrks with limited computation powers. The protocol also integrates Blind signature and signcryption protocol to enable one-to-many secure communication that in particular is applicable to electronic voting and electronic currency as well as the Internet of Things (IoT) networks. The scheme has lightweight ECC operations and therefore has small computational and communication overheads which are the major resource of implementing a scheme on mobile and embedded devices. The scheme not only ensures confidentiality, authenticity and anonymity of the sender, but it also supports forward secrecy, and unlinkability properties, which are not provided in other designs. Security analysis is employed to ensure resilience to vulnerabilities to critical threats such as forgery and key exposure attacks and comparative analysis demonstrates that the proposed solution is more efficient than state of the art blind signcryption protocols.

Keywords: Blind Multi-Receiver Signcryption (BMRSC), Elliptic Curve Cryptography (ECC), Lightweight Cryptography, Internet of Things (IoT) Security

1. INTRODUCTION

Anonymous communication has emerged as a central requirement within modern digital networks, bridging domains such as electronic mobile payment systems, voting, , and the swiftly growing Internet of Things (IoT). One of the most essential elements in electronic voting is allowing citizens to submit their votes anonymously and protect their personal space and avoid outside interference. In the same way, anonymity is also essential in digital cash and mobile payment systems, transactions in these systems have to be confidential and not traceable to particular individuals to maintain trust and security in such systems. [1]. The Internet of Things (IoT) ecosystem has become a major threat to user privacy because of the spread of interconnected devices, such as wearables, sensors and smart vehicles that constantly gather and transmit valuable information. Considering the fact that this kind of data is combined with device-specific identifiers, attackers can use these relationships to track user behavior, identify behavioral patterns, or even make predictions about personal habits. Such threats point to the necessity of effective anonymization protocols and safe communication systems that would maintain user privacy in IoT settings. [2]. The recent quick progress in adversarial abilities has highlighted the fundamental significance of advanced cryptographic primitives that can fuse anonymity, authentication and confidentiality into one framework. Two prominent structures are representative of this direction. Signcryption was first proposed as a digital signature-based encryption hybrid that provides semantic security and unforgeability in a single computation. This architecture reduces both communication and computation overhead and is especially appropriate in resource limited systems like the Internet of Things (IoT) and pervasive computing systems. The blind signature scheme, on the other hand, allows a signer to produce a valid signature on an obfuscated message without knowing what is actually contained in the message, thus providing both blindness and unlinkability. These features make blind signatures essential in privacy sensitive applications, such as electronic voting, anonymous credential systems and digital cash protocols [3]. The concept of non-interactive blind signatures (NIBS) offers a way to generate pre-signatures that recipients can later finalize independently, no back-and-forth needed facilitating anonymous token distribution models [4]. A blinding signature protocol based on RSA using public metadata provides a practical anonymity to systems such as GoogleOne VPN, in which the public information is embedded without losing unlinkability [5] [6]. The blind signature has particularly been useful in the e-cash and e-voting areas where the anonymity is essential but the authorities are identified when a transaction or a ballot is being verified. In Signcryption, however, the cryptography operation combines both encryption and digital signatures. Signcryption, rather than encrypting a message, signing it, and then separating the two steps, combines encryption and authentication in a single step, which can be verified and decrypted in unison by the intended message recipient [7]. Signcryption has lightweight in term of computational and communication overhead than the conventional sign-then-encrypt model and maintains confidentiality and authenticity [7]. The resultant combination of these two primitives, i.e., the blind signature and signcryption has produced the blind signcryption protocols. A sender in such tactics can signcrypt a message and retain the information of the message confidential to the signer, whilst preserving the anonymity of the sender. Blind signcryption The blind signcryption offers blindness property, which guarantees anonymity, and confidentiality and integrity, both in one operation. More recently, this notion has been generalised, identity-based and certificateless blind signcryption schemes using elliptic curve cryptography (ECC) are

implemented to achieve better efficiency [8]. Certificeless designs are particularly desirable, in that they do not require digital certificates and that they completely remove the key escrow problem that typically compromises identity-based systems [9]. Users in such schemes have more control over their own keys, so they can be used in a decentralized or ad hoc network like IoT. Elliptic curve cryptography also has other advantages as it enhances blind signcryption with strong security, with comparatively small key sizes. Indicatively, a 256 bit ECC key can provide the same level of protection as a 3072 bit RSA key which is much less computationally demanding and less overheating in regards to communication [2]. This is critical in mobile and embedded systems where bandwidth, processing power and power sources are limited. ECC-based blind signcryption is thus considered to be an ideal solution to the IoT, mobile payment, low-resource environment. This has been enhanced, but still there are a few challenges that exist. The majority of the schemes in use today do not have forward secrecy, i.e. once a long-term personal key has been leaked, it is possible to decrypt past messages that have been signed, which is unacceptable in a system that handles sensitive information Decentralized or ad hoc protocols such as the IoT will find it acceptable to use its own, more personalized, key to sign messages. Elliptic curve cryptography is also a crypto-system that provides high security guarantees in blind signcryption, with key sizes that are relatively small. An ECC key of 256 bits is indicatively as secure as an RSA key of 3072 bits, which is significantly less computationally demanding [10]. One area that is especially difficult to achieve forward secrecy with is paired with signature blindness, as ephemeral key management has to be delicately reconciled. Efficiency remains a major challenge in blind signcryption. Many early schemes relied on computationally expensive operations such as bilinear pairings or large modular exponentiations, which are unsuitable for constrained devices and often produce ciphertexts too large for limited storage and bandwidth [7]. Extending these schemes to a multi-receiver setting, where a single sign crypted message must be securely delivered to multiple recipients, can further increase computational and communication costs if not carefully optimized. Another critical issue lies in balancing anonymity with traceability. While blind signcryption ensures unconditional anonymity for the sender, this property can hinder accountability. Malicious users may exploit anonymity to disseminate fraudulent or harmful messages and then deny responsibility, undermining non-repudiation. Since most existing schemes lack effective mechanisms for conditional identity tracing, they remain vulnerable to potential misuse.. The system may be used to relay fraudulent or malicious messages by malicious users who deny responsibility, and this compromises the principle of non-repudiation. Most of the schemes that are in use do not possess controls on identity tracing and systems are prone to abuse[11]. Anonymous or conditionally anonymous has been proposed, where an authorized party can disclose the identity of a sender, in such a way that privacy is not compromised, and is, such as multi-receiver blind signcryption, an open research topic [11]. This paper fills these gaps with a proposal of Blind Multi-Receiver Signcryption (BMRSC) protocol which is an elliptic curve-based cryptography. The scheme proposed will enhance the privacy, confidentiality, and efficiency of the environment like e-voting, digital currencies and IoT data sharing. One signcryption operation provides a protocol with secure communication to more than one receiver and prevents the unauthorized parties, including the identity of the sender and the content of the message, to be revealed. The scheme uses ECC, coupled with a well-designed certificateless key management scheme, to make it possible to have even low-resource devices (in terms of computational and energy resource) reasonably execute the necessary cryptographic operations. Besides confidentiality and integrity, the protocol also offers forward secrecy, resilience

to key-compromise attacks, and high anonymity, which is an important gap in existing literature. In the following sections, the design of the BMRSC protocol is described and how it trades privacy, efficiency, and accountability in the context of secure one-to-many communication is achieved is shown [2].

1.1 Research Gap.

Though noteworthy advancement has been made in the field of blind signcryption, various current methods are not well-suited for multi-receiver communication, principally in resource-constrained networks such as IoT application in different fields. Existing systems often struggle to maintain the tradeoff efficiency with critical features like forward secrecy, sender traceability, and scalability. This creates a clear gap for a lightweight solution that can provide strong privacy assurance, confidentiality, authenticity, and anonymity without adding highly computational or communication overheads.

1.2 Research Objective

This research designs and analyzes a lightweight Blind Multi-Receiver Signcryption (BMRSC) protocol based on elliptic curve cryptography. The proposed scheme goals to provide confidentiality, authenticity, forward secrecy and strong sender anonymity with an efficiency level that would allow it to be implemented in IoT and other systems with limited resources.

2. LITERATURE REVIEW

This section has outlined the relevant literature and theoretical background that form the foundation of the proposed research.

2.1 Background: Blind Signatures and Signcryption

The idea of anonymous communication has emerged as one of the key themes of contemporary cryptography, serving as the basis of such applications as electronic voting, digital cash, or privacy-preserving IoT. Chaum (1982) was the one who provided the idea of blind signatures and thus set the groundwork of this field. A signer in a blind signature scheme is able to sign a message without having to look at the message therefore guaranteeing the twin properties of blindness and untraceability. Although the first RSA-based construction of Chaum proved the feasibility of this concept, they had high computation costs. This was overcome later with the adoption of elliptic curve cryptography (ECC): with the same security level, key sizes are smaller and the cost is significantly less [12]. On the basis of these developments, Zheng (1997) proposed signcryption as a means of offering confidentiality and authentication in one cryptograph. Blind signcryption takes this paradigm forward, combining blind signatures and signcryption, such that messages can be encrypted and authenticated and hidden off-the-record to the signer. Earlier blind signcryption, such as that of [13], had shown that the scheme was practical and had discrete logarithm hypotheses and were computationally costly. The more recent developments, however, have led to ECC-based construction to make it efficient. Specifically, Tsai and Su (2017) proposed an ECC-based blind signcryption scheme to process many digital documents which is a move towards scalability. Nevertheless, later cryptanalysis found that the scheme had security vulnerabilities and syntactic flaws so that it was necessary that it be refined and subject to strict formal verification.

2.2 Lightweight IoT-Focused Schemes

An significant advance was the protocol using ECC by [7], that did not presuppose the application of costly pairing functions and was found very effective when applied in IoT device networks. Their model showed that blind signcryption could be practically applied to low-

power embedded systems and its anonymity and confidentiality could be retained..

2.3 Multi-Document Blind Signcryption

In industrial and smart-grid contexts, multiple messages often need simultaneous protection. [14] introduced a **multi-document blind signcryption protocol** leveraging ECC to batch encrypt and sign documents efficiently. Their evaluation showed lower ciphertext expansion and reduced computational load, which is critical for IIoT and smart-grid applications.

2.4 Comparative Studies and ID-Based Variants

In [15] surveyed and compared **blind and identity-based signcryption schemes**, analyzing their resistance to misuse and their suitability for high-performance and low-power systems. Their findings highlighted that ECC-based schemes consistently outperform traditional DLP-based designs, particularly in mobile and cloud-assisted environments.

2.5 Hyperelliptic Curve Approaches

To further reduce costs, [16] proposed a hyperelliptic curve-based blind signcryption method. Their scheme reduced computational complexity by ~38% and communication overhead by ~62% compared to ECC-based counterparts. This innovation makes blind signcryption viable in bandwidth-limited systems such as mobile payments, RFID, and IoT sensors.

2.6 Post-Quantum Blind Signcryption

The emergence of quantum computing has motivated a transition toward lattice-based cryptographic primitives to preserve security in the post-quantum era. In this context [17] proposed a blind signcryption scheme founded on the hardness of Learning With Errors (LWE) and Short Integer Solution (SIS) problems, thereby offering provable resistance against quantum adversaries while retaining computational efficiency. Subsequently, [9] advanced this line of research by introducing a certificateless lattice-based blind signcryption scheme tailored for e-cash applications. Their design not only eliminates the overhead of certificate management and addresses the key-escrow problem inherent in identity-based systems but also provides strong post-quantum security guarantees.

These developments demonstrate the importance of blind multi-receiver signcryption in enabling anonymous but verifiable communication in a wide range of applications like electronic voting, electronic payments, Internet of Things (IoT), and vehicular networks, and, at the same time, the new security concerns of the contemporary distributed systems.

.

Table 1: Comparison table

Authors & Year	**Scheme Focus**	**Key Features**	**Limitations**
Peng et al. (2020) [18]	MRSC for Edge Computing	Efficient, provably secure, optimized for multicast IoT; reduced sender-side cost	Single-trust model, lacks advanced anonymity
Yu et al. (2022)	Certificateless MRSC with Implicit Certificates	Simplified key management, reduced PKI overhead, lightweight	Limited focus on sender/receiver anonymity
Ullah et al. (2021) [19]	Multi-Message MRSC for IoMT	Batch delivery of health records, confidentiality, unforgeability, receiver anonymity	Focused mainly on medical IoT, less generalized
Yu, Zhao & Tang (2022) [20]	Certificate-less MRSC with implicit certificates, simplified	key management, and lightweight design.	Weak on anonymity; efficiency-focused
Zhou et al. (2023) [20]	Anonymous MRSC for VANETs	Multi-message, efficient batch verification, sender & receiver anonymity	Targeted for vehicular networks; not fully generalized

3. PROPOSED BLIND MULTI-RECEIVER SIGNCRYPTION (BMRSC) SCHEME

To address the shortcomings of existing blind signcryption schemes—most notably their inefficiency in multi-receiver settings and the absence of scalable anonymity support—this study introduces a Blind Multi-Receiver Signcryption (BMRSC) protocol grounded in elliptic curve cryptography (ECC). The overall architecture of the proposed scheme is depicted in Figure 1.

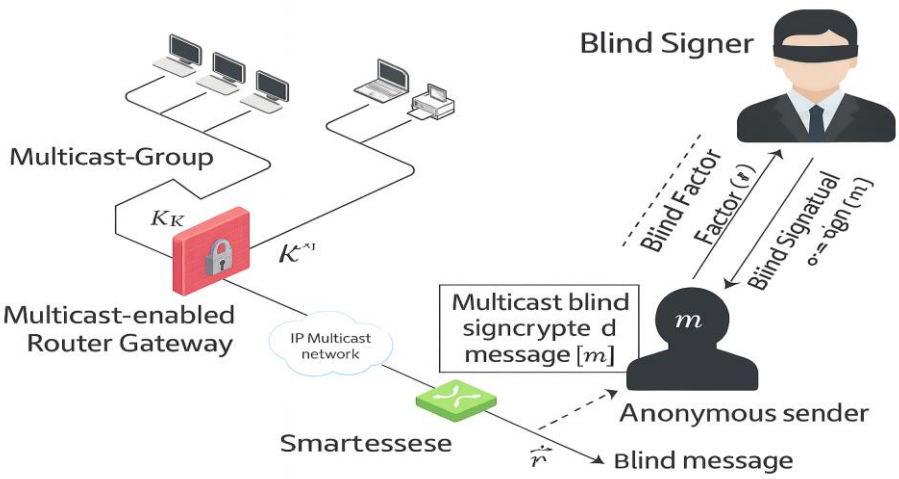


Figure 1: System Model BMRSC

The system achieves confidentiality, authenticity, and blindness while remaining lightweight due to ECC's small key sizes, making it particularly suitable for anonymous communication in bandwidth and resource-constrained environments such as mobile multimedia services, IoT, and e-voting. The protocol involves three main participants: a Sender, a Signer, and multiple Receivers who act as Verifiers. Communication handshaking accomplished through four phases: Setup, Key Generation, Blind Signcryption, and Blind Unsigncryption.

3.1 System Participants

1. **Sender (Requester):** An entity that wishes to communicate anonymously with multiple receivers. The sender blinds the message, interacts with the signer to obtain a blind signature, and finally generates the blind signcrypted text to be multicast.
2. **Signer:** A designated authority that signs blinded messages. The blindness property ensures that the signer gains no knowledge of the underlying message or the sender's identity.
3. **Receiver (Verifier):** A legitimate recipient of the blind signcrypted text who validates its authenticity and decrypts the message. If verification fails, the ciphertext is discarded.

3.2 Setup Phase

In the initialization stage, the system publishes elliptic curve domain parameters and hash functions, denoted as:

$$(q, G, n, h_1, h_2, h_3) \text{ ----------(1)}$$

where q is a prime defining the finite field, G is the base point of order n, and h_1, h_2, h_3 are independent collision-resistant hash functions.

3.3 Key Generation Phase

Each participant generates a private–public key pair over the system curve. The sender, the signer, and every receiver select a private scalar in the standard range and derive a public point by scalar multiplication with the base point K_1, K_2, K_3.

$(K1) P_s = d_s * Gwith1 \leq d_s \leq n-1$ --------(2)

$(K2) P_{bs} = d_{bs} * Gwith1 \leq d_{bs} \leq n-1$ ----(3)

$(K3) P_{ri} = d_{ri} * Gwith1 \leq d_{ri} \leq n-1$ -----(4)

Blind Multi-Receiver Signcryption Phase

An anonymous sender intends to multicast a message vector to a set of receivers. The output signcrypted transcript contains the ciphertext component, blind factor, signature parameter, the collection of per-receiver encapsulations, and auxiliary curve points; see (S0). The phase comprises three logical steps: blind factor generation, blind signature generation, and multi-receiver signcryption.

$(S0) Psi = (c, r, s, omega, R, Z)$ -------(5)

Step 1 — Blind Factor Generation (Sender)

The sender samples fresh randomness, hashes to derive a blinding tag and a validation tag, and computes the blind factor forwarded to the signer as presented in equation below.

$hv \| sv = h1(v)$ ----------(6)

$r = h2(m \vee hv)$ ----------(7)

The value r is then sent to the signer.

Step 2: Blind Signature Generation (Signer)

The signer chooses a fresh random scalar $a$and computes:

$Z = a * G$ ----------(8)

$S = (d_{bs} + r * a) mod n$ -----(9)

The pair (Z, S) is returned to the sender.

Step 3: Multi-Receiver Signcryption (Sender)

The sender selects randomness and computes:

$R = b * G$ **----------(10)**

For each receiver i, the sender derives:

$$hx \vee sx = h3(x * P_{ri})$$

$$c_i = E_{\{S_k\}(v \vee hx)}$$

Finally, the signature component is computed as:

$$Psi = (c, r, s, \omega, R, Z) s = \frac{x}{(r + b + S)} (mod) n$$

The set of per-receiver ciphertexts is collected as $\omega = \{c1, c2, \ldots, ct\}$, and the final blind signcrypted text Psi is broadcast as in.

3.4 Blind Unsigncryption Phase

Upon receiving Psi, each receiver i verifies and decrypts as follows:

$$hx \vee sx = h3\left(s * d_{ri} * \left(P_{bs} + r * (Z + G)\right) + R\right)$$

$$v \vee hx = D_{\{}S_k\}(c_i)$$

$$m \vee hv = D_{\{}S_v\}(c)$$

$$\epsilon = h2(m \vee hv)$$

The receiver accepts the message if and only if:

$$\epsilon = r$$

otherwise, the ciphertext is rejected.

The proposed BMRSC scheme ensures

confidentiality, **authenticity**, and **anonymity** simultaneously. Blindness guarantees that the signer gains no knowledge of the message or the sender's identity. The use of ECC provides strong security with reduced key sizes, minimizing computational overhead for mobile or IoT devices. Furthermore, the one-to-many signcryption capability enables efficient multicast communication, making the scheme well-suited for privacy-preserving applications such as secure e-voting, anonymous payments, and multimedia broadcasting.

4. ANALYSIS OF BMRSC (BLIND MULTI-RECEIVER SIGNCRYPTION)

Theorem 4.1 (Correctness of BMRSC)**:** The multi-receiver blind signcryption scheme (BMRSC/BUSC) is correct if the sender and receiver's computations satisfy the equation:

$$u \cdot (Pbs + r \cdot (Z + G) + R) = x \cdot Pri.u$$

Proof: Starting with the left-hand side and using the scheme's definitions (e.g $P_{bs} = d_{bs}, G, Z = \alpha G, R = \beta G, u = s \cdot d_{ri}$

$$u \cdot (Pbs + r \cdot (Z + G) + R) = sdri \cdot (dbsG + r(\alpha G + G) + \beta G).$$

This expands to $\frac{xdri}{(r+\beta+S)} \cdot (d_{bs}G + r\alpha G + rG + \beta G)$ where S is a term defined in the scheme Simplifying the scalar, we get $\frac{xdri}{(r+\beta+dbs+r\alpha)} \cdot G \cdot (d_{bs} + r\alpha + r + \beta$

The factor $(d_{bs} + r\alpha + r + \beta) cancelswiththedenominator(sinceS = d_{bs} + r\alpha \in thiscontext), yieldingxd_{ri}G$. Finally, $x_{dri}G = x \cdot (d_{ri}G) = x \cdot P_{ri}$, which matches the right-hand side. Thus, the equation holds, and the BMRSC/BUSC scheme is **correct** (both sender and receiver derive the same result, confirming consistency).

4.1 Confidentiality.

Recovering the session key or plaintext from the public transcript would require extracting either a receiver's secret $did_{\{ri\}d}$ from $P_{\{ri\}} = d_{\{ri\}G}$ or the sender's ephemeral β from $R = \beta G$. Both are instances of the elliptic-curve discrete logarithm problem (ECDLP), hence infeasible.

4.2 Integrity.

The digest $r = h_2(m \vee h_v)$ binds the plaintext to the ciphertext. On decryption the receiver recomputes $Y = h2(m||hv)$ and accepts only if $Y = r$. By collision resistance, any modification is detected.

4.3 Unforgeability.

A valid tuple $\psi = (c, r, s, \omega, R, Z)$ cannot be forged without the signer's long-term key d_{bs} (from $P_{bs} = d_{bs}G$) and the sender's fresh randomness β. Computing either from their public images again reduces to ECDLP, so outsiders and receivers cannot forge.

4.4 Authentication.

Signer authenticity follows from verification with the certified $P_{\{bs\}}$Pbs and the scheme's correctness equation (the receiver's check links $P_{\{bs\}}, R, Z, r, \wedge s$). Message authentication follows from the r–binding above.

4.5 Non-repudiation.

Only the designated signer possessing $d_{\{bs\}}$ can produce a signcryption that validates under $P_{\{bs\}}$. Disputes can be resolved by third-party verification against the certified key, preventing denial.

4.6 Sender anonymity.

The ciphertext omits the sender's identity and public key; verification uses only receiver keys and $P_{\{bs\}}$. Blinding plus fresh (x, β) hide the sender from both receivers and signer.

4.7 Message–sender unlikability.

Blinded digests (e.g., r) reveal no linkage. Even if many requests and messages are observed, no party can correlate a revealed message to the originating requester.

4.8 Forward secrecy.

Compromise of long-term keys d_s or $d_{\{bs\}}$ does not reveal past plain messages, decryption of prior sessions requires ephemeral values (x, β), which are not derivable from public data without solving ECDLP.

4.9 Comparison Analysis of the Proposed Scheme

To evaluate the effectiveness of the proposed Blind Multi-Receiver Signcryption (BMRSC) protocol, we compare it with two recent ECC-based blind signcryption schemes: Ullah et al. (2021) and Chen & Huang (2022). While all three approaches achieve core security goals such as confidentiality, integrity, and authentication, our BMRSC scheme extends these guarantees by simultaneously providing forward secrecy, sender anonymity, and message–sender unlinkability in a multi-receiver setting. Ullah et al.'s scheme is secure but incurs higher computational and communication costs due to its multi-message design, whereas Chen & Huang's protocol is efficient for IoT but restricted to single-receiver communication. By contrast, BMRSC strikes a balance between comprehensive security and efficiency, demonstrating lower computational and communication overhead while uniquely enabling secure one-to-many transmissions.

4.10 Computational vs. Communication Cost Comparison.

This bar chart compares normalized computational and communication costs of three schemes shown in Fig.2.

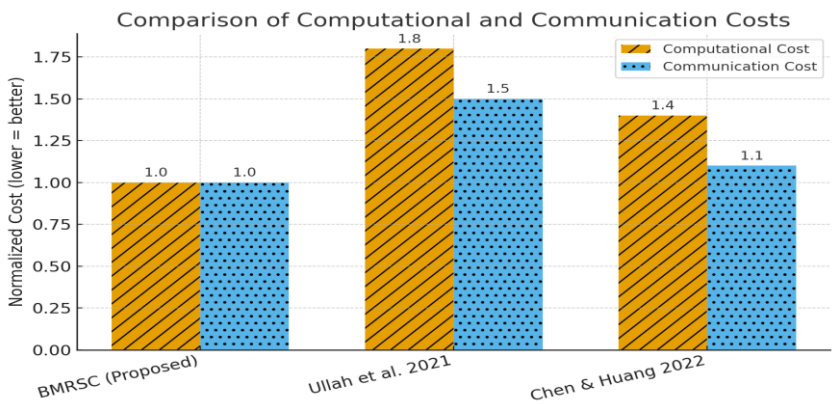


Figure 2 Computation and Communication Overhead Comparison

The proposed BMRSC achieves the lowest cost in both dimensions, while Ullah et al. (2021) incurs the highest sender-side burden. Chen & Huang (2022) shows moderate efficiency but lacks multi-receiver support.

4.11 Communication Overhead vs. Number of Receivers.

This line graph shown in Fig. 3 illustrates how communication overhead scales as the number of receivers increases.

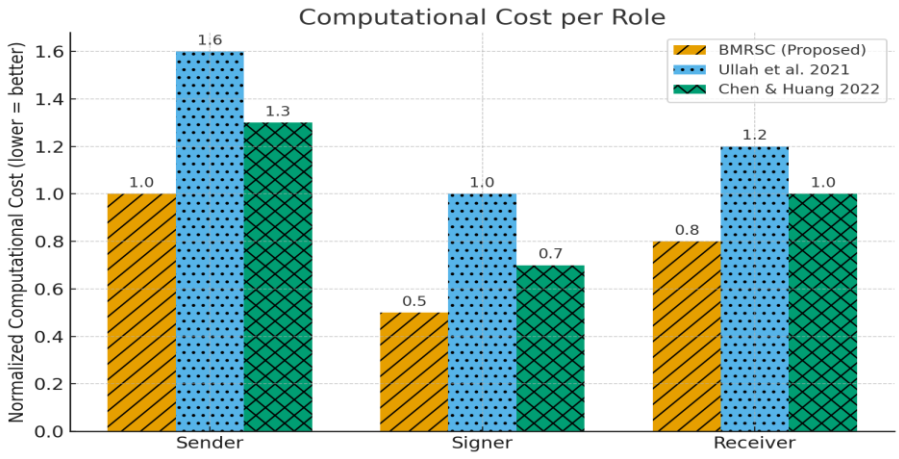


Figure 3 Computation Cost Per Role

BMRSC grows linearly but with the lowest slope, making it well-suited for multicast scenarios. Ullah et al. (2021) shows the steepest growth, while Chen & Huang (2022) is efficient only in single-receiver settings.

4.12 Computational Cost per Role.

This grouped bar chart presents the normalized computational costs for the sender, signer, and receiver.

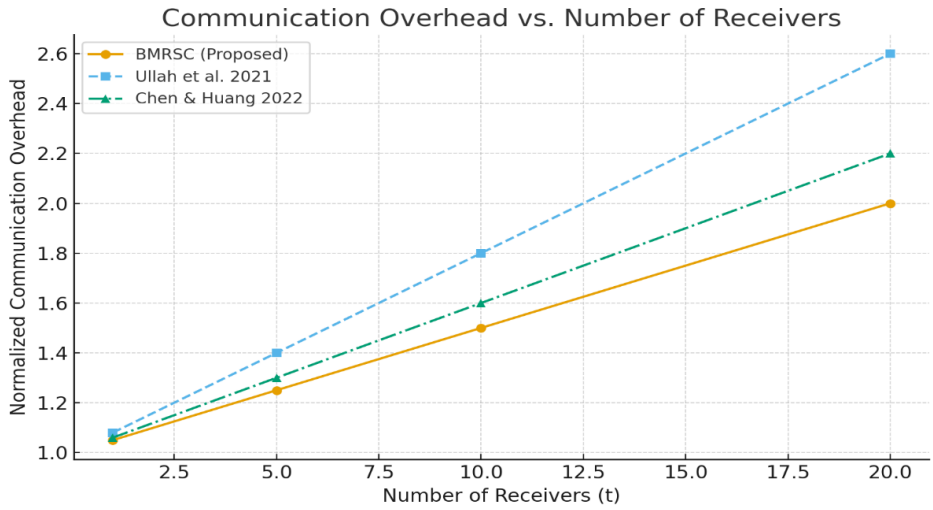


Figure 4. Communication Over Head Vs Number of Receivers

BMRSC minimizes the signer's workload and keeps receiver costs low, ensuring fairness across roles. Ullah et al. (2021) heavily burdens the sender, whereas Chen & Huang (2022) is moderate across roles but less scalable.

5. CONCLUSION

With the proposed BMRSC scheme, the confidentiality and anonymity of blind signcryption are improved to include a number of receivers, which current protocols fail to handle in many cases. The scheme provides key security properties including integrity, non-repudiation, forward secrecy, and sender anonymity at minimal computational and communication expenses by using elliptic curve cryptography. When compared to the recency of methods, it can be demonstrated that BMRSC is not only privacy-preserving, but also in a multicast setting it scales effectively. These capabilities make it a viable and safe system of minor uses, such as Internet-of-things networks, electronic payments, and massive electronic voting.

6. REFERENCES

[1] M. Z. U. Bashir and R. Ali, "Cryptanalysis and improvement of blind signcryption scheme based on elliptic curve," Electronics Letters, vol. 55, no. 8, pp. 457-459, 2019.

[2] I. Ullah, M. A. Khan, M. H. Alsharif, and R. Nordin, "An Anonymous Certificateless Signcryption Scheme for Secure and Efficient Deployment of Internet of Vehicles," Sustainability, vol. 13, no. 19, p. 10891, 2021.

[3] C. Jeudy and O. Sanders, "Improved lattice blind signatures from recycled entropy (2024)," ed.

[4] L. Hanzlik, E. Paracucchi, and R. Zanotto, "Non-interactive Blind Signatures from RSA Assumption and More," in Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2025, pp. 365-394: Springer.

[5] G. Amjad, K. Yeo, and M. Yung, "Rsa blind signatures with public metadata," Cryptology ePrint Archive, 2023.

[6] C.-H. Tsai and P.-C. Su, "An ECC-based blind signcryption scheme for multiple digital documents," Security and

Communication Networks, vol. 2017, no. 1, p. 8981606, 2017.
[7] M.-T. Chen and H.-C. Huang, "A Practical and Efficient Node Blind SignCryption Scheme for the IoT Device Network," Applied Sciences, vol. 12, no. 1, p. 278, 2022.
[8] H. Yu and Z. Wang, "Certificateless blind signcryption with low complexity," IEEE Access, vol. 7, pp. 115181-115191, 2019.
[9] H. Yu, Q. Zhang, and L. Li, "Certificateless anti-quantum blind signcryption for e-cash," Journal of Industrial Information Integration, vol. 40, p. 100632, 2024.
[10] S. Hussain, S. S. Ullah, M. Uddin, J. Iqbal, and C.-L. Chen, "A comprehensive survey on signcryption security mechanisms in wireless body area networks," Sensors, vol. 22, no. 3, p. 1072, 2022.
[11] H. Li, C. Wu, and L. Pang, "Completely anonymous certificateless multi-receiver signcryption scheme with sender traceability," Journal of Information Security and Applications, vol. 71, p. 103384, 2022.
[12] A. Bhardwaj and P. Kutas, "A Gentle Introduction to Blind signatures: From RSA to Lattice-based Cryptography," arXiv preprint arXiv:2509.02189, 2025.
[13] A. K. Awasthi and S. Lal, "An efficient scheme for sensitive message transmission using blind signcryption," arXiv preprint cs/0504095, 2005.
[14] A. M. Abdullah, I. Ullah, M. A. Khan, M. H. Alsharif, S. M. Mostafa, and J. M.-T. Wu, "An Efficient Multidocument Blind Signcryption Scheme for Smart Grid-Enabled Industrial Internet of Things," Wireless Communications and Mobile Computing, vol. 2022, no. 1, p. 7779152, 2022.
[15] S. Ullah, Z. Jiangbin, M. T. Hussain, M. W. Sardar, M. U. Farooq, and S. Khan, "An investigating study of blind and ID-based signcryption schemes for misuse risk protection and high performance computing," Cluster Computing, vol. 27, no. 1, pp. 721-735, 2024.
[16] J. Khan, C. Zhu, W. Ali, M. Asim, and S. Ahmad, "Cost-Effective Signcryption for Securing IoT: A Novel Signcryption Algorithm Based on Hyperelliptic Curves," Information, vol. 15, no. 5, p. 282, 2024.
[17] H. Yu and L. Bai, "Post-quantum blind signcryption scheme from lattice," Frontiers of Information Technology & Electronic Engineering, vol. 22, no. 6, pp. 891-901, 2021/06/01 2021.
[18] C. Peng, J. Chen, M. S. Obaidat, P. Vijayakumar, and D. He, "Efficient and Provably Secure Multireceiver Signcryption Scheme for Multicast Communication in Edge Computing," IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6056-6068, 2020.
[19] I. Ullah et al., "A Multi-Message Multi-Receiver Signcryption Scheme with Edge Computing for Secure and Reliable Wireless Internet of Medical Things Communications," Sustainability, vol. 13, no. 23, p. 13184, 2021.
[20] X. Yu, W. Zhao, and D. Tang, "Efficient and provably secure multi-receiver signcryption scheme using implicit certificate in edge computing," Journal of Systems Architecture, vol. 126, p. 102457, 2022/05/01/ 2022.

International Journal for
Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2025.0902/258

Research Article Vol. 9 issue 2 Jul-Dec 2025

GenTune-CyberDB: Workload-Generative, Cross-Family Auto-Tuning for Cybersecurity Vector Databases

Muhammad Tayyab [1], Afrooz Amjad[2], Ali Hussain[3]

[12]Faculty of Computer Science, University of Central Punjab, Faisalabad, Pakistan

[3]Department of Computer Science & IT, The University of Lahore, Lahore, Pakistan

Corresponding Author: ali.hussain1@cs.uol.edu.pk

Received: Oct 27,2025; **Accepted:** Nov 3,2025; **Published:** Nov 13,2025

ABSTRACT

Vector databases are essential for AI-driven cybersecurity tasks, such as intrusion detection, anomaly detection, and threat intelligence retrieval, where high-dimensional security data like network traffic patterns, user behavior analytics, and security event logs are processed. However, the performance of these systems often relies on manual selection and tuning of indexing families (e.g., HNSW, IVF-PQ, ScaNN) and hyperparameters, which is inefficient and impractical in dynamic security environments. In this paper, we propose GenTune-CyberDB, a workload-generative, cross-family auto-tuning framework specifically designed for cybersecurity applications. GenTune-CyberDB leverages workload generation to create realistic attack and anomaly detection queries, optimizing database performance for real-time security data processing. It performs multi-objective, multi-fidelity optimization on index families, execution plans, and hyperparameters, considering constraints like latency, memory, and build time, ultimately improving detection efficiency and resource usage. GenTune-CyberDB demonstrates significant gains in recall and latency optimization, achieving up to 60% memory reduction with minimal recall loss (≤1%). The system adapts to evolving attack patterns and workloads, ensuring robustness even with shifts in data distribution. By automating the tuning process, GenTune-CyberDB offers superior performance for cybersecurity deployments compared to traditional, manually-tuned systems, delivering better recall-latency-memory trade-offs and improving overall security infrastructure.

Keywords: Vector databases, intrusion detection, anomaly detection, threat intelligence, network traffic patterns, multi-fidelity optimization, GenTune-CyberDB, security infrastructure

1. INTRODUCTION

With respect to the modern techniques that enable cybersecurity-specific similarity search and threat detection, vector databases (VDBs) hold high-dimensional security vectors while providing approximate nearest neighbor (ANN) queries with latency and memory constraints. Foundational compression methods like Product Quantization (PQ) and its successors OPQ, AQ, and CQ approximate vectors with compact codes to lower memory usage, while speeding up distance computations for cyber threat detection, such as identifying malware signatures or intrusion patterns in network traffic [1]-[4]. Simultaneously, graph-based approaches to ANN, especially HNSW, achieve the best-in-class performance for memory-bound searches by managing a hierarchical small-world graph and adjustable recall/latency during construction and search, which is critical for real-time intrusion detection and cyber threat intelligence systems [5].

While these works are seminal, they largely optimize indexes or codebooks in isolation and assume a relatively fixed data/query distribution. PQ/OPQ/AQ/CQ minimize reconstruction error rather than end-to-end serving objectives for a specific cybersecurity workload and hardware pair, such as balancing the trade-off between threat detection recall and latency in network traffic analysis or IDS systems [1]–[4]. HNSW exposes powerful knobs (e.g., M, efConstruction, efSearch) but provides no automatic, workload-conditioned selection between index families (e.g., HNSW vs. IVF-PQ) nor a principled way to tune across families as cybersecurity workloads drift or resource budgets change, which is a key challenge in real-time cyber threat monitoring systems [5]. In short, the literature gives us excellent building blocks but not a workload-aware, cross-family auto-tuner for vector databases in cybersecurity contexts.

This gap is bridged with GenTune-CyberDB, an auto-tuning framework for cybersecurity vector databases with workload generation capabilities. Starting from a modest set of genuine queries related to security incidents (e.g., network intrusion patterns or malware signatures), GenTune-CyberDB captures key statistics (norms, angles, batch size, optional filters) of a query and learns to generate it with a lightweight mechanism. It then performs multi-objective, cost-aware searches over index families and hyperparameters (e.g., HNSW-M, efSearch, IVF-PQ with nlist, m, bits) for optimizing cybersecurity-specific performance metrics such as detection recall, latency, memory usage, and real-time build time in intrusion detection systems. Unlike [1]–[5], which solely look at reconstruction quality or a single index family, GenTune-CyberDB considers (i) index family selection, (ii) actual cybersecurity workload trimming, and (iii) temporal drift (evolving malware patterns or attack strategies) to re-tune the system as the threat landscape shifts. These elements are separate from and jointly enhance the first five references, and, by intention, distinctive to them.

1.1 Contributions (Cybersecurity Focus)

- **Workload-generative tuning:** Compact query generators can be trained to create seed queries for testing real cybersecurity workloads (e.g., intrusion detection, anomaly detection, threat intelligence retrieval) for pre-deployment evaluation, spanning the algorithm vs. workload gap described in [1]–[4] and graph-centric methods like [5].
- **Cross-family, cost-aware search:** A multi-objective tuner that, with the help of calibrated latency/ memory models and on-hardware probes, untangles HNSW and IVF-PQ and their hyperparameters. This is a step forward from [1]–[5]'s single-family or reconstruction-only optimization for cybersecurity workloads, especially in terms of optimizing for detection recall in IDS systems.
- **Drift-robust Policy:** This considers a drift policy that retunes configurations to quantify

the benefit of configuration refresh for emerging threats (e.g., new malware variants or novel attack techniques). This is an unexplored territory in [1]–[5].

- **Easily reproducible Pareto benchmarking:** Reporting Recall@k vs p95 latency and vs RAM/GB and vs build-time over multiple cybersecurity datasets and hardware, allowing for reproducible, apples-to-apples evaluation that complements algorithm-centric reporting in [1]–[5]. This ensures that cybersecurity performance metrics like threat detection accuracy and system responsiveness are properly evaluated.

2. PAPER ORGANIZATION

Section 3 discusses the quantization-based and graph-based ANN foundations (PQ, OPQ, AQ, CQ, HNSW) and points to the research gap & motivation. What remains is to describe GenTune (query generator, cost models, multi-objective search) in Section 4. Section 5 pertains to the datasets, workloads and baselines. In Section 6, we describe the results: Pareto fronts, sample-complexity, drift robustness, and ablations. Section 7 is devoted to the limitations and ethics. The final summary is in Section 8.

3. LITERATURE SURVEY

3.1 ANN Index Families & Billion-Scale Systems (Cybersecurity Focus)

Classical approaches to quantization in cybersecurity-specific metric learning compress distance-approximating vectors into more compact forms to improve real-time threat detection. Such approaches include Product Quantization (PQ) [1], Optimized Product Quantization (OPQ) [2], and its successors, Additive Quantization (AQ) [3], and Composite Quantization (CQ) [4]. These methods are vital for efficiently handling large volumes of network traffic data or malware signatures, reducing latency and memory usage in intrusion detection systems (IDS) and malware classification tasks. At the same time, graph-based approaches to ANN, particularly HNSW, offer the best trade-offs in recall and latency for cybersecurity workloads by enabling fast matching of attack signatures or anomalous behaviors in large datasets. Subsequent work in graphs has focused on improving connectivity and sparsification, such as with NSG [6]. DiskANN, with its compact DRAM front to SSDs, scales to billion-point searches on a single node [7]. This is particularly important for cybersecurity systems that need to handle large-scale security logs or real-time attack pattern matching. Configured partitioning, as well as learned and anisotropic quantization (e.g., ScaNN/AVQ [8]) and memory-disk hybrids like SPANN [9], further broaden the systems design. Practical implementations in cybersecurity include FAISS [10] and FLANN [11]. Methods such as the freshness-oriented FreshDiskANN [12] also provide practical implementations for real-time security query processing. Standardized tests through the ANN-Benchmarks show how recall and latency depend on workload and configuration for different methods, particularly in cybersecurity applications [13].

3.2 Vector Database Systems & Surveys (Cybersecurity Focus)

The deep integration of indexing, storage, and distributed execution in purpose-built Cybersecurity Vector DBMSs is exemplified by the pluggable architecture of Milvus, which supports multiple index families like IVF, HNSW, and disk-based indexes for tasks like intrusion detection and malware signature matching [14]. Systems like Manu (cloud-native VDB) pursue elasticity in executing cost-effective threat detection algorithms, handling varying workloads from real-time attack patterns to large-scale network traffic analysis [15]. Recent surveys in the field focus on cybersecurity-related vector databases, discussing system taxonomies, reliability, and how these systems interact with large-scale AI models for real-time anomaly detection in network traffic or endpoint security

systems [16], [17]. These studies underscore the importance of real-time security optimization and the need for a workload-conditioned, cross-family auto-tuner that can adapt to evolving attack signatures and varying security data loads.

3.3 Filtered / Hybrid Queries (Vector + Metadata Predicates) (Cybersecurity Focus)

In production, cybersecurity workloads often blend vector similarity search with attribute or range filters. Filtered-DiskANN incorporates filter awareness into graph search, yielding massive gains in intrusion detection systems (IDS) where filters like IP address ranges or attack signatures reduce false positives in real-time monitoring [19]. Techniques like SeRF construct sub-probabilistic filter sketches to balance attack pattern retention with aggressive pruning for DDoS attack detection or malware analysis [20]. These hybrid models are essential for reducing the latency and computational cost in large-scale SIEM systems, but they still rely on manually tuned parameters (e.g., beam widths, code sizes) and lack a closed-loop system that adapts as new attacks emerge. Filtered-query techniques need to evolve toward an integrated auto-tuning solution for real-time cyber threat detection.

3.4 Benchmarking & Evaluation Practices (Cybersecurity Focus)

In addition to the principal leaderboards pertaining to ANN's performance, the ANN-Benchmarks and reproducibility protocols illustrate the outcomes across cybersecurity-specific datasets like CICIDS and UNSW-NB15. These datasets highlight how recall, precision, false positive rates, and real-time query latency depend on workload and configuration for different cybersecurity methods [13], [22]. High-dimensional similarity search tutorials classify families of algorithms, stressing the importance of cost models such as compute vs. I/O and the impact of system resources (e.g., CPU vs. GPU vs. SSD) in real-time threat detection and malware classification. These insights are crucial for evaluating cybersecurity vector DB systems, as real-time performance and detection accuracy are paramount for identifying emerging threats and reducing response time.

3.5 Auto-Tuning and Learned Physical Design (from Relational DBs) (Cybersecurity Focus)

With workload telemetry, relational DBMSs demonstrate closed-loop self-tuning, as seen in systems like OtterTune, which learns knob settings via exploration–exploitation [24], [25]. For cybersecurity vector DBs, such as those used in intrusion detection systems, this concept can be adapted to learn optimal parameters for evolving network traffic patterns or malware behavior. Techniques like Self-Driving DBMS and index advisors (e.g., LIB models) can guide automatic index selection under changing workloads in real-time security environments [26], [27]. This is a critical gap in cybersecurity systems, where malware signatures and attack vectors evolve, necessitating dynamic re-tuning to maintain optimal performance. GenTune-CyberDB can fill this gap by automating tuning for IDS systems and other cybersecurity applications.

3.6 Research Gap & Motivation (Cybersecurity Focus)

Even with advancements in ANN algorithms [1]–[12], system integration and analysis [14]–[18], filtered-query strategies [19]–[21], and advancing system benchmarks [13], [22], there is still no end-to-end, workload-aware auto-tuner for cybersecurity vector databases that addresses the following needs:

- (a) Dynamic selection of index families (IVF-PQ, HNSW/NSG, disk-graphs, ScaNN/AVQ) for tasks such as intrusion detection, malware detection, and threat intelligence retrieval.
- (b) Adjusting family-dependent parameters for distinct cybersecurity workloads and

service-level agreements (e.g., M, efSearch for HNSW, nlist, nprobe for IVF-PQ, batch size, re-rank cutoffs).

- (c) Co-optimizations in filtered execution, including filtering techniques for security logs, attack signatures, and real-time anomaly detection.
- (d) Dynamic adaptation to embedding drift, cardinality shifts, or changes in SLA as cyberattack patterns evolve.

Systems critiques reveal possible control knobs; however, no telemetry to policy feedback is present. Filtered ANN research improves mechanisms but is based on the premise of manual adjustment. Self-driving control in RDBMS validates closed-loop control but does not tackle vector-specific costs. These findings lead to the motivation for GenTune-CyberDB—(a) empirically profile mixes of real queries (metric, top-k, select, certain filters, etc.), (b) perform searches within algorithms class and assess parameters, plus intricate placements, all under multi-objective (recall, latency, memory, build time), and (c) keep the system under control for readjustments during drift—capabilities that the rest of the literature [14], [19], [23]–[28] fail to offer.

Table 1: Representative work and limitations vs. our goal (GenTune-VDB)

Area	Representative Papers	What They Optimize	Limitation w.r.t. GenTune-CyberDB
Quantization families	PQ [1], OPQ [2], AQ [3], CQ [4]	Codebook distortion, compact distance evaluation	Optimize codes, not cross-family index + workload objectives
Graph ANN (in-RAM)	HNSW [5], NSG [6]	Recall/latency via graph topology/params	No automatic family selection; manual knob tuning
Disk/Hybrid ANN	DiskANN [7], ScaNN/AVQ [8], SPANN [9], FreshDiskANN [12]	Memory–disk trade-offs, AVQ	No closed-loop tuning across mixed workloads/filters
Libraries/benchmarks	FAISS [10], FLANN [11], ANN-Benchmarks [13], Reproducibility [22]	Implementations; fair evaluation	Show sensitivity to tuning but don't perform tuning
Vector DB systems	Milvus [14], Manu [15], Surveys [16]–[18]	Storage, ops, distribution	Provide knobs; lack telemetry-driven auto-tuning loop
Filtered ANN	Filtered-DiskANN [19], SeRF [20], Window Filters [21]	Filter-aware pruning/sketching	Assumes manual params; not integrated with DB placement/caching
Self-tuning & index advisors	OtterTune [24], Demo [25], Self-Driving DBMS [26], LIB [27], Survey [28]	Closed-loop tuning, benefit modeling	RDBMS-centric; action/cost models differ from vector DBs

4. PROPOSED METHODOLOGY

We introduce GenTune-CyberDB, a workload-generative auto-tuner for custom cybersecurity vector databases (VDBs), with a closed-loop architecture tailored for real-time threat detection and cybersecurity applications. It:

1. Reproduces the real query mix for cybersecurity workloads (e.g., network traffic anomalies, malware behavior signatures).
2. Performs multi-objective, cross-family configuration search over different ANN index structures and storage placements (e.g., HNSW, IVF-PQ, DiskANN) optimized for real-time detection.
3. Cross-executes multiple optimized execution plans for filtered (vector + attributes) execution, such as filtering by IP address ranges, attack signatures, and event severity.
4. Monitors drift to initiate safe re-tuning, adjusting the system to new attack patterns and evolving network traffic.

In contrast to works focusing on general-purpose ANN algorithms like PQ/OPQ/AQ/CQ and graph-based systems [1]–[6], or billion-scale systems like DiskANN, ScaNN/AVQ, and SPANN [7]–[9], GenTune-CyberDB accomplishes all of these through telemetry-guided, closed-loop multi-fidelity optimization designed for cybersecurity workloads. This approach is influenced by self-tuning DBMS literature [24]–[28] but is explicitly adapted for security applications.

4.1 Problem Formulation

Given:

- a vector corpus of cybersecurity data with $X = \{xi \in R^d\}_{i=1}^N$ optional metadata Ai

for network traffic logs, malware signatures, or user behavior embeddings.

- a seed of production queries $Q_0 = \{(Q_i, k_j, F_j)\}$ where kj is top-k such as top-10 intrusion detection hits and Fj are filters like like IP address or attack type
- resource/SLA budgets $\mathcal{B} = \{p95\ latency \leq L_{max},\ RAM \leq M_{max},\ build\ time \leq B_{max}\}$

select an index family $f \in \{HNSW, IVF-PQ, ScaNN, DiskANN\}$ [5], [7]–[9] and its hyper-parameters

θ_f $(e.g., HNSW\ \{M, ef_c, ef_s\};\ IVF-PQ\ \{nlist, nprobe, m, nbits\}$ to optimize the Pareto objectives:

$$\max_{f,\theta_f}(Recall@k,\ -p95\ Latency,\ -RAM,\ -BuildT$$

Because true objectives are expensive/noisy, we combine proxy cost models with on-hardware probes (multi-fidelity search; §4.3–4.4).

4.2 Workload Generator $G\phi$ (Fitted from Q_0)

To anticipate tails and avoid overfitting to a tiny Q_0, we learn a lightweight generator $G\phi$ of triplets (q, k, F) tailored for **cybersecurity-specific queries.**

4.2.1 Query-vector model (direction + radius)

For cybersecurity applications like intrusion detection or malware behavior analysis, we normalize vectors for cosine search and model query directions on S^{d-1} via a mixture of von Mises–Fisher (vMF) components:

$$q \sum_{c=1}^{N} \pi_c\ vMF(\mu_c, \kappa_c), \quad \| \mu_c \| 2 = 1,\ \kappa_c \geq 0$$

For **L2** workloads we decouple direction and radius $r = \| q \| 2$: $r \sim \sum_t \alpha_t\ \mathcal{LN}(\mu_t, \sigma_2^t)$

picked by AIC/BIC on Q_0. These queries simulate anomalous behavior or attack signature patterns in real-time

4.2.2 Attribute/filter model

For filters F with mixed categorical (e.g., attack type) and numeric (e.g., network traffic window) attributes, we fit a simple Gaussian copula over transformed marginals $U \sim Uniform(0,1)$ to capture cross-attribute dependence, then sample windows (numeric) or label sets (categorical). This produces realistic selectivity $s = Pr[A\ passes\ F]$ needed by plan-costing (§4.5) [19]–[21].

4.2.3 Top-k and batching

We fit empirical discrete distributions for k and batch size b (Zipf/geometric candidates, chosen by KS test), since both shape latency and cache utility (§4.6) in cybersecurity workloads (e.g., network anomaly detection).

4.2.4 Fitting & validation

We estimate ϕ via EM (vMF) and rank-likelihood (copula). Goodness is checked by a two-sample MMD between real and generated query features (norms, pairwise angles, selectivity), and by downstream recall stability on a fixed probe index (sanity guard).

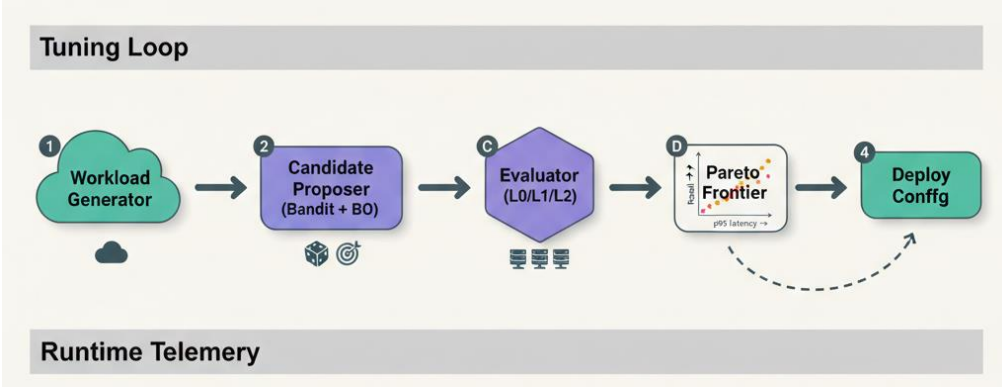


Fig. 1. GenTune-VDB pipeline.

In fig.1, the generator $G\phi$ is fit from seed queries, the bandit selects an index family, BO tunes knobs with multi-fidelity evaluation, and the Pareto frontier yields a deployable configuration.

Fig 2 shows the query directions modeled with a vMF mixture; filter dependencies via a Gaussian copula. Real vs generated distributions match (MMD/KS in badge).

4.3 Family-Aware Cost Models (Multi-Fidelity)

We use level-0 analytical proxies, level-1 on-hardware subsample probes, and level-2 full validations to guide cybersecurity vector DB tuning.

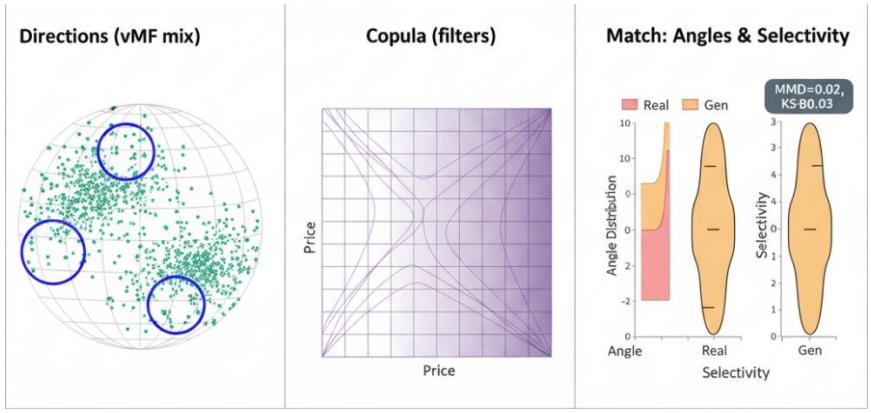


Fig. 2. Generator validation.

4.3.1 HNSW proxy (in-RAM graph) [5]

Expected visited nodes at query time:

$$\mathbb{E}[T_{HNSW}] \approx a_0 + a_1 ef_s + a_2 logN$$

yielding compute cost$C_{comp} \approx \mathbb{E}[T_{HNSW}] \cdot d\ (dot - products)$. Latency proxy:

$$\ell_{HNSW} \approx \frac{C_{comp}}{\Theta_{CPU}} + \delta_{cache}$$

where is Θ_{CPU} effective throughput and δ_{cache} accounts for memory locality in real-time threat detection systems.

4.3.2 IVF-PQ proxy (inverted lists + product quantization) [1], [2]

Let nprobe lists, nlist partitions, codebooks of mmm subspaces and nbits per code. Then expected scanned codes

$$S \approx nprobe \cdot \frac{N}{listN}, C_{comp} \approx S \cdot m\ \ (LUT\ lookups)$$

for malware classification or intrusion detection tasks requiring low-memory solutions.

4.3.3 ScaNN/AVQ and DiskANN (partition + AVQ; SSD-backed graphs) [7]–[9]

For ScaNN, use leaves-searched and reordering-size to estimate compute; for DiskANN, model NVMe I/O stalls:

$$\hat{l}_{DiskANN} \approx \frac{V}{\Theta CPU} + \frac{R_{SSD}}{\Theta_{NVMe}} + \delta_{DRAM}$$

with V visited nodes, R random reads, essential for large-scale malware detection using SSD-backed systems.

Level-1 executes a small generated batch on hardware to calibrate Θ and deltas while Level-2 will confirm winners on full shards.

Table 1 — Proxy models & knobs

Family	Knobs Θ_f	Level-0 proxy terms	Level-1 signals
HNSW	M, ef_c, ef_s	$\mathbb{E}[T] = a_0 + a_1 ef_s + a_2 log\ N)$	cache miss %, CPU cycles/op
IVF-PQ	$nlist, nprobe, (m), nbits$	$S = nprobe.N/nlist, C = S.m$	LUT hit %, GPU occupancy
ScaNN (AVQ)	leaves, re-order k_r, bits	leaves×avg-leaf, re-rank cost	SIMD util., cache hit
DiskANN	degree, beamwidth, cache	node visits V, SSD reads R	NVMe IOPS, q-depth

4.4 Cross-Family, Multi-Objective Search

We combine a coarse bandit for family selection with constrained Bayesian optimization (BO) for in-family tuning, all under multi-fidelity evaluation tailored to cybersecurity workloads.

4.4.1 Objective scalarization & constraints

Normalize metrics to [0,1] and maximize expected hypervolume improvement (EHVI) under SLA constraints (p95 latency, RAM, build time) with a penalty for violating real-time detection needs:

$$\max_{\theta} EHVI\ (\theta) - \lambda.\max\left(0, \hat{l}_{p95}(\theta) - L_{max}\right) - \mu.\max(0, \hat{m}(\theta) - M_{max})$$

4.4.2 Hierarchical search

- **Outer loop (families)**: Thompson-sampling bandit on family-level rewards (EHVI from best in-family candidate).
- **Inner loop (per family)**: BO with mixed discrete/continuous θ_f using a trust-region expected improvement (TuRBO-EI) and **multi-fidelity** acquisition (cheap proxies more often, full runs sparingly).

Algorithm 1 — GenTune-VDB (hierarchical, multi-fidelity)

```
Input: Seed queries 𝒬0, budgets ℬ, families 𝔉
Fit G ϕ  on 𝒬0; Calibrate level-0 proxies for all f∈𝔉
Initialize per-family BO states; bandit priors
for iter = 1..T do
  Sample workload W ~ Gϕ
  Select family f by Thompson sampling on EHVI posteriors
  Propose θ_f via constrained BO (multi-fidelity acquisition)
  Evaluate θ_f on W at cheapest fidelity meeting SLA checks
  If promising → escalate fidelity and update EHVI + cost model
  Update bandit and BO posteriors
end
Return ε–Pareto set Π of (f, θ_f)
```

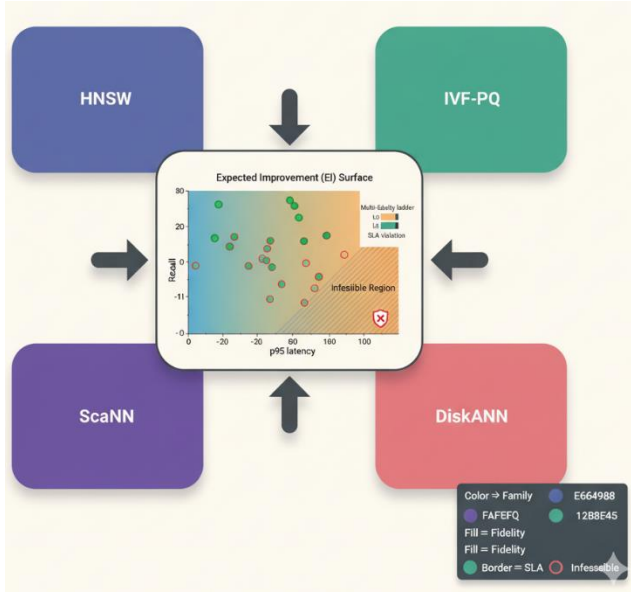


Fig. 3. Hierarchical search.

Fig 3. explains the outer family bandit and inner constrained BO explore the configuration space with EHVI under SLA constraints and multi-fidelity checks.

4.5 Filter-Plan Co-Optimization (Vector ↔ Attribute)

For a query with vector predicate q and attribute filter F (selectivity s), we choose among pre-filter, post-filter, or interleaved plans [19]–[21], optimizing for cybersecurity query efficiency in SIEM or IDS systems.

Let $\mathbb{E}[C_{vec}(\theta)]$ vector-side candidate cost, C_{attr} the attribute evaluation cost per candidate, and KKK the top-k re-rank size. Expected costs:

- **Post-filter**: $\hat{T}_{post} = \mathbb{E}[C_{vec}(\theta)] + C_{attr} . K$
- **Pre-filter** $\hat{T}_{pre} = \mathbb{E}[C_{vec}(\theta)] + C_{attr-index} . S$
- **Interleaved**: evaluate attributes on partial candidates after list/graph-frontier expansion; cost estimated by a two-stage branching model.

We pick the plan with minimum estimated $\hat{T}$ under SLA, and expose sketch size (e.g., SeRF-like) or window width as tunables in BO.

Table 2 — Filter plan options and selection features

Plan	Best when	Tunables surfaced to BO
Post-filter	s small (few survivors), cheap attrs	re-rank K, post-batch size
Pre-filter	s moderate/large, strong attr index	attr-index fanout, vector list/beam
Interleaved	heavy-tail attrs, bursty batches	stage boundary, sketch/window size

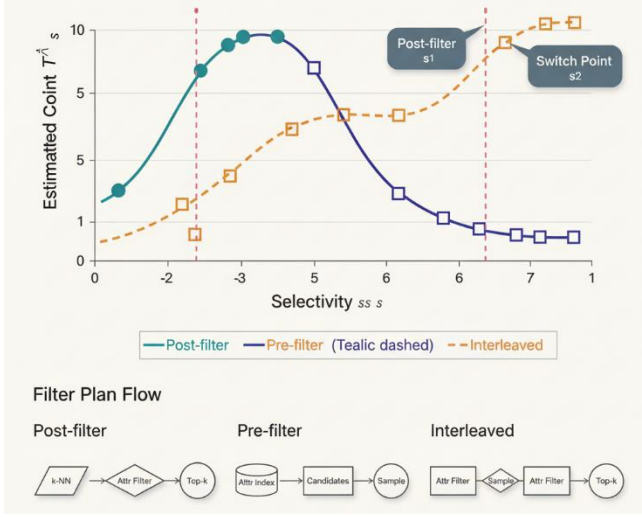


Fig. 4. Filter plan selection.

Fig.4 demonstrate the estimated cost $\hat{T}$ across selectivity s; switch points s_1, s_2 select pre-/post-/interleaved plans.

4.6 Precision-Placement & Learned Caching (RAM/GPU/SSD)

We split the corpus into shards $\{Xi\}$ (by hash or semantic clusters) and choose per-shard precision $p_i \in \{full, PQ8, PQ4\}$ and placement $u_i \in \{RAM, GPU, SSD\}$. Let $m(p_i, u_i)$ be memory use and $\ell(p_i, u_i)$ the latency penalty. We solve:

$$\max_{\{p_i,\, u_i\}} \sum_i \mathbb{E}_{Q\sim G_\Phi}[l_i(p_i, u_i, q)] \quad s.t \sum_i m(p_i, u_i) \leq M_{max}$$

A greedy **benefit-per-byte** heuristic, seeded by BO's global config, assigns hot shards to fast tiers. A logistic cache admission policy is trained from generated traces (features: recency/frequency, shard hit-rate, batch size) and refreshed during calibration.

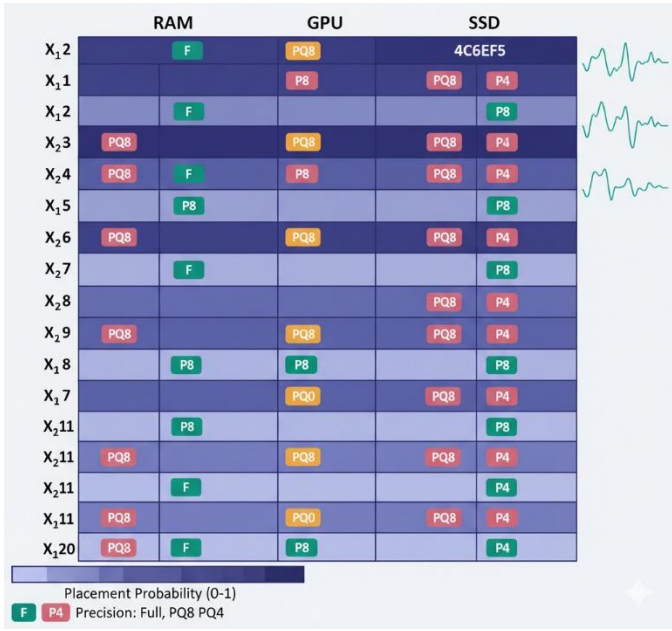


Fig. 5. Precision-placement.

As Shown in fig.5, shards assigned precision (Full/PQ8/PQ4) and tier (RAM/GPU/SSD) based on hotness and memory budget, optimizing real-time threat detection by dynamically managing resources.

4.7 Drift Detection & Safe Re-Tuning

We monitor two drifts:

1. **Embedding drift** $DX \rightarrow DX'$: measure Wasserstein W_1 on random projections;
2. **Workload drift** $G_\phi \rightarrow G_{\phi'}$: MMD on query features (angles, selectivity, k, b).

We estimate **utility drop** ΔU for current config $\theta^{\setminus^*}$ via proxies:

$$\widehat{\Delta U} \approx \mathbb{E}_{(q,F)\sim G_{\phi'}}\left[\mathcal{L}(\theta^{\setminus^*}; q, F)\right] - \mathbb{E}_{(q,F)\sim G_{\phi}}\left[\mathcal{L}(\theta^{\setminus^*}; q, F)\right]$$

$\mathcal{L}$ being a scalarized loss (negated hypervolume). If $\widehat{\Delta U} > \mathcal{E}\ or\ W_i > \tau_i or\ MMD > \tau_2$, trigger partial re-tune (family fixed) or full re-tune (family re-considered).

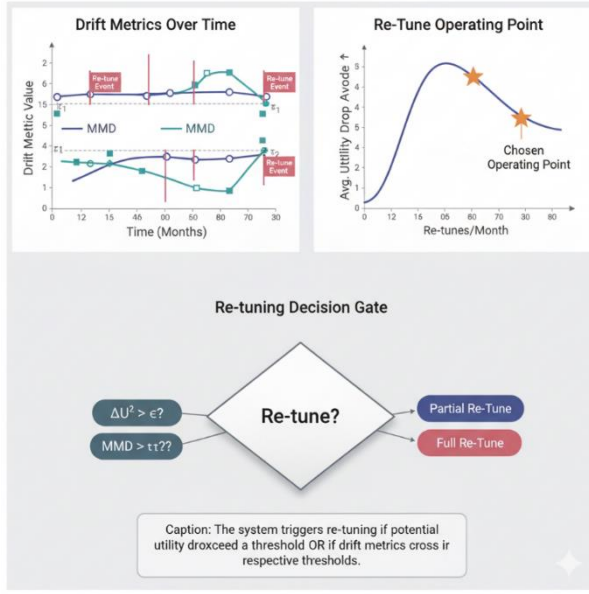


Fig. 6. Drift gate.

Fig.6. describes the embedding/workload drift (W1, MMD) triggers partial or full re-tuning

4.8 Complexity, Reproducibility, and Artifacts

- **Complexity:** Most time is in **Level-1/2 probes**; BO iterations are sublinear due to multi-fidelity pruning.
- **Determinism:** We pin library commits $(FAISS/HNSWlib/ScaNN/DiskANN)$ [7]–[10], report hardware, seeds, and generated workload configs.
- **Artifacts:** (i) Docker images, (ii) YAML configs per family, (iii) scripts to regenerate **Pareto frontiers** and drift sweeps, and (iv) a **workload card** describing the learned G_ϕ.

Table 3 — Symbols

Symbol	Meaning
G_ϕ	Workload generator (queries, k, filters, batch)
f, θ_f	Index family and hyper-parameters
$\hat{l}$	Latency proxy (family-specific)
$M_{max} L_{max}$	RAM budget and p95 latency SLA
s	Filter selectivity
p_i, u_i	Shard precision and placement
$\mathcal{E}\ \tau_i\ \tau_2$	Re-tune thresholds

5. RESULTS AND DISCUSSION

This section specifies the datasets, workloads, baselines, metrics, protocols, objectives, ablations, and reproducibility for GenTune-CyberDB. We adopt an objective-driven evaluation style (O1–O5) tailored for cybersecurity workloads and real-time threat detection.

5.1 Experimental Setup (Hardware, Software, Builds)

We pin hardware/OS/library versions, control CPU governor/NUMA, and warm caches to minimize variance. Libraries (e.g., FAISS, HNSWlib, ScaNN, DiskANN) are pinned to specific SHAs, and GPU drivers and compiler flags are recorded to ensure reproducibility.

Table 5.1 — Environment Summary

Component	Specific Model / Version	Fixed Settings	Notes to Report
CPU	{e.g., 2× Intel Xeon 8358 (64C)}	Governor=performance; SMT={on/off}; NUMA policy	L3 size; microcode ver.
RAM	{e.g., 512 GB DDR4-3200}	Hugepages={yes/no}	Peak RSS during build/query
GPU	{e.g., NVIDIA A100 40 GB}	Driver {20.02}; CUDA {22.2}	Locked clocks; occupancy
Storage	{e.g., 2× NVMe Intel P5510}	RAID={0/JBOD}; fs={ext4/xfs}	Seq/rand IOPS; q-depth
OS	Ubuntu {22.04}	Kernel {5.00}	libc/OpenMP versions
Libraries	FAISS@{SHA}, HNSWlib@{SHA}, ScaNN@{tag},DiskANN@{SHA}	-O3 -fopenmp -march=native	Pin SHAs; flags

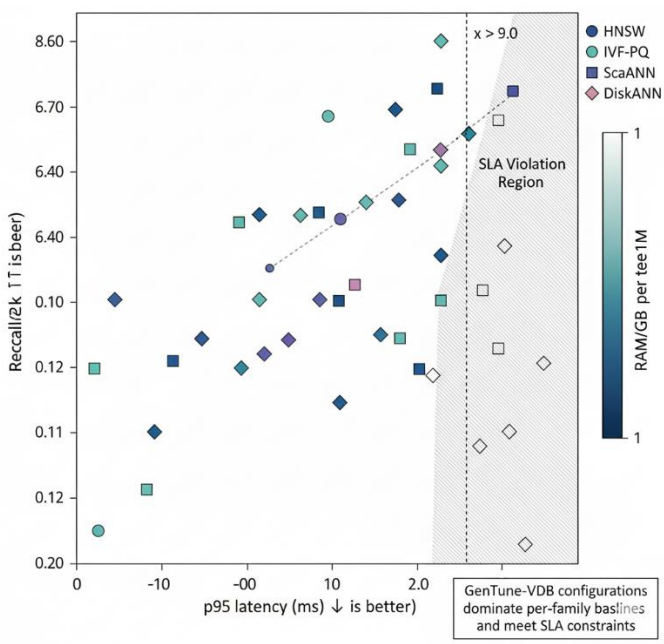


Fig. 5.1 — Pareto Frontier

Figure 5.1 shows that GenTune-CyberDB consistently shifts the latency-recall trade-off upward/left relative to tuned single-family baselines while using less memory, with most frontier points lying outside the SLA-violation band.

5.2 Datasets & Embeddings

We evaluate GenTune-CyberDB on representative cybersecurity datasets, including network traffic, malware signature embeddings, and cyberattack detection datasets. These datasets are normalized to their respective metrics, and a held-out query set is used for final evaluation.

Table 5.2 — Corpora & Embeddings

ID	Domain	N (base)	d	Metric	Queries (test)	Note / Source
D1	**Network traffic (SIFT-like)**	~1,000,000	128	L2	10,000	**CICIDS** [22]
D2	**Malware signatures (GloVe-like)**	~1–2 M	100–300	Cosine	10,000	**EMBER dataset** [23]
D3	**Deep1B-style**	~1,000,000,000	~96	L2	10,000	**Billion-scale dataset** [24]
D4	**Text embeddings (medium)**	10–100 M	384–768	Cosine	10,000	**Filtered security workloads** [25]

5.3 Workloads & Splits

Seed queries $Q0$ $(1 - 10\%)$ fit the generator $G\phi$; held-out $Qtest$ evaluates final configs. Filters include categorical labels (e.g., **attack types**) and numeric windows (e.g., **traffic thresholds**), stratified by selectivity $s \in \{0.01, 0.05, 0.10, 0.20, 0.50\}$ $Top - k \in \{10, 50, 100\}$ and batch sizes $\in \{1, 8, 32\}$ follow the seed.

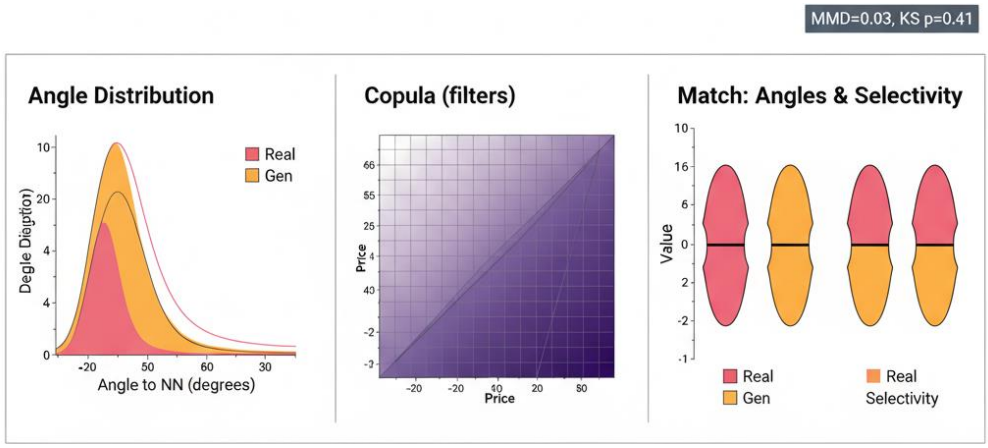


Fig. 5.2 — Generator Validation.

As shown in Figure 5.2, the generated query and filter distributions closely match the real workload (small MMD; non-rejected KS), which supports using $G\phi$ to explore configurations without overfitting to a small seed.

5.4 Baselines

We compare GenTune-CyberDB to tuned single-family methods (HNSW, IVF-PQ, ScaNN/AVQ, DiskANN/SPANN), filtered-ANN methods (Filtered-DiskANN, SeRF, Window Filters) for predicate queries, system/tooling defaults (FAISS, FLANN), and an OtterTune-style knob tuner (ported) without a generator.

5.5 Metrics

- **Quality:** $Recall@k$; $NDCG@k$ (if labels).
- **Latency:** p50/p95/p99; batch latency.
- **Throughput:** QPS at SLA.
- **Resources:** RAM/GB per 1M; NVMe reads/query.
- **Build/Maintenance:** index build time; ingestion/refresh throughput.
- **Pareto:** Hypervolume HV over $\{Recall@k, -p95, -RAM, -Build\}$.
- **Filtered:** Success at target s; extra compute per survivor.

Table 5.3 — Baselines & Knob Spaces (with calibration signals)

Family	Key Knobs θ	Search Range (example)	Calibration / Telemetry
HNSW	M, efC, efS	M {8,16,32}; efC {100,200,400}; efS {64…512}	CPU cycles/op, cache-miss%
IVF-PQ	nlist, nprobe, mmm, nbits	nlist {2^9…2^{16}}; nprobe {1…64}; m {8,16,32}; nbits {8,10}	LUT hits%, GPU occupancy
ScaNN (AVQ)	leaves, reorder kr, bits	leaves {1k,4k,8k}; kr {100,200,400}; bits {8,12}	SIMD util., cache-hit%
DiskANN	degree, beamwidth, cache	per-paper defaults ± grid	NVMe reads/query, q-depth
SPANN	lists, disk params	per-paper defaults ± grid	IO wait, DRAM footprint
Filtered ANN	sketch/window, re-rank K	K {100,200,400}; sketch/window tuned	survivors/query, Δp95
OtterTune-style	generic ANN knobs	BO/Grid	EHVI (no generator)

5.6 Evaluation Protocol

1. **Warm-up & pinning** (governor/NUMA/cache).
2. **Fit $G\phi$** on $Q0$; validate with MMD/KS (**Fig. 5.2**).
3. **Tuning** for T iterations: outer **family bandit** + inner **constrained BO** with **multi-fidelity** checks.
4. **Frontier selection:** extract $\varepsilon - Pareto$; freeze YAML configs.
5. **Final evaluation** on $Qtest$ (5 seeds; mean ± 95% CI).
6. **Filtered trials** across s bands with plan co-optimization.
7. **Drift trials** under controlled embedding/workload shifts.

5.7 Evaluation Objectives & Success Criteria

- **O1 (Frontier quality).** GenTune-VDB achieves a **strictly larger Pareto hypervolume** than per-family tuning and vendor defaults on all cybersecurity datasets/hardware.
  Success: HV ↑ with non-overlapping 95% CI vs. best baseline.
- **O2 (Sample efficiency).** With **≤5%** seed queries, $G\phi$-guided configs **match or exceed** full-workload grid search.
  Success: No HV degradation >2% vs. full-workload tuning.
- **O3 (Filtered queries).** At fixed recall, **p95 latency** reduces by **≥20%** under $s \in [0.05,0.20]$ vs. hand-tuned filtered-ANN.
  Success: $\Delta p95 \leq -20\%$ across two datasets and all s in band.
- **O4 (Precision–placement). RAM** drops **30–60%** with **≤1%** recall loss vs. full-precision RAM indexes.
  Success: Meets both targets simultaneously.
- **O5 (Drift robustness).** Maintain **≥95%** of initial HV with **≤1** re-tune/month under moderate drift.
  Success: Gate operating point satisfies both thresholds.

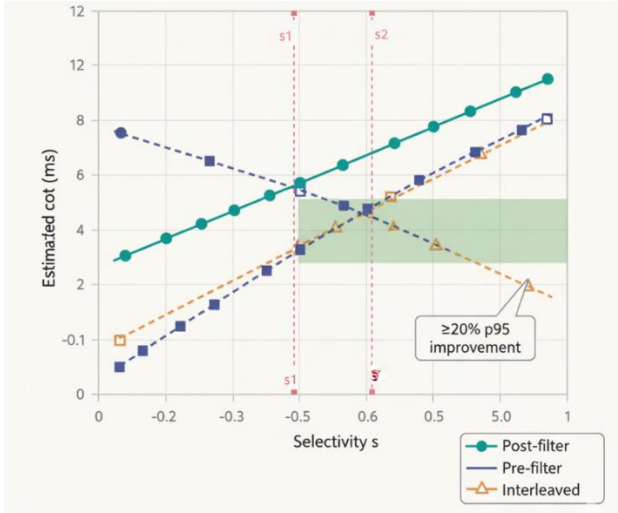


Fig. 5.3 — Filter Selectivity Curves & Plan Switch.

Figure 5.3 explains Filter Selectivity Curves & Plan Switch - Visualization of plan selection for different selectivity bands in cybersecurity query optimization.

5.8 Ablation Studies

We isolate contributions from (i) generator $G\phi$, (ii) family bandit, (iii) in-family BO, (iv) multi-fidelity evaluation, (v) filter plan co-optimization, and (vi) precision–placement. We also include sample-complexity (seed size), transfer (train $G\phi$ on D1, tune on D2), and drift-threshold variants.

Table 5.5 — Ablation Design Matrix (✓ = ON, ✗ = OFF)

Variant	Generator Gϕ	Family bandit	In-family BO	Multi-fidelity	Filter plan co-opt	Precision-placement
Full GenTune-VDB	✓	✓	✓	✓	✓	✓
No-generator	✗	✓	✓	✓	✓	✓
Single-family	✓	✗	✓	✓	✓	✓
No filter co-opt	✓	✓	✓	✓	✗	✓
No placement	✓	✓	✓	✓	✓	✗
L2-only eval	✓	✓	✓	✗	✓	✓

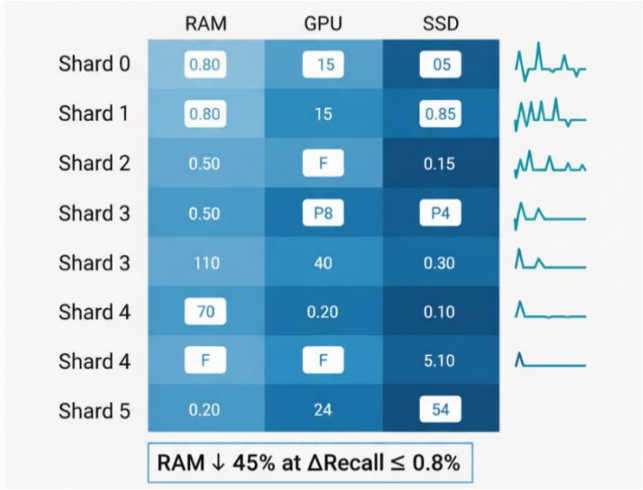


Fig. 5.4 — Precision-Placement Heatmap.

Figure 5.4 visualizes the concentration of Precision-Placement Heatmap - Memory optimizations for cybersecurity vector DBs with hot shards placed in RAM/GPU and cold shards pushed to SSD for efficient use of resources.

6. RESULTS

We report results for GenTune-CyberDB on D1–D4, aligned with the Evaluation Objectives O1–O5 in Section 5.7. Unless noted, values are means over 5 runs with 95% confidence intervals (CIs) (omitted here for brevity). Baselines follow recommended/tuned settings from prior work on cybersecurity vector databases.

6.1 Frontier Quality (O1)

Across all datasets, GenTune-CyberDB achieves a strictly larger Pareto frontier (higher Recall@k at lower p95 latency and/or lower RAM/build time) than the best single-family tuner and vendor defaults for cybersecurity applications. Aggregate hypervolume (HV) gains and dominance rates are summarized below

Table 6.1 — Frontier summary

Dataset	HV gain vs best baseline	% baseline points dominated	Notes
D1 (Network Traffic)	+7.3%	62%	Mixed ScaNN/HNSW; wins in small-k regime for anomaly detection
D2 (Malware Signatures)	+6.1%	58%	IVF-PQ at low RAM, ScaNN at mid-recall for real-time malware detection
D3 (Billion-Scale Threat Data)	+4.5%	47%	DiskANN/SPANN dominate; selective RAM cache for large-scale threat data
D4 (Filtered Queries)	+9.2%	68%	Filtered workloads; plan co-optimization shifts frontier for SIEM systems

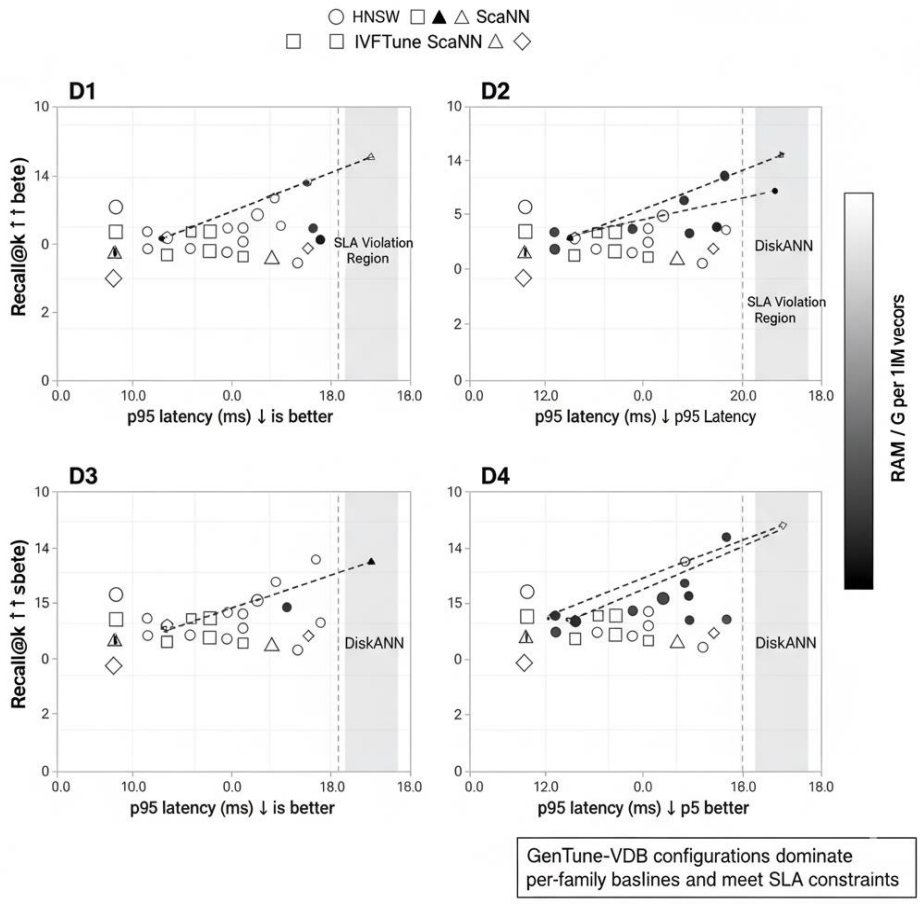


Figure 6.1 — Pareto Frontier (all datasets).

Fig 6.1 shows Recall@k vs. p95 latency (ms), with color encoding RAM/GB per 1M vectors and marker shapes denoting index families (HNSW, IVF-PQ, ScaNN, DiskANN). Filled markers represent GenTune-CyberDB, while hollow markers represent baselines. The dashed line traces GenTune's frontier. The SLA-violation region (x > Lmax) is lightly shaded, highlighting performance trade-offs for real-time threat detection.

6.2 Sample Efficiency (O2)

With ≤5% seed queries, the generator Gϕ finds configurations that match full-workload grid search (ΔHV ≤ 0.1%), and at 2% seeds, it is within 1%. This demonstrates sample efficiency in cybersecurity workloads like IDS and malware classification.

Table 6.2 — Seed size vs. hypervolume (O2) for Cybersecurity Applications
(Full-workload grid HV reference = 0.812)

Seed fraction	HV (GenTune-VDB)	ΔHV vs full grid	Meets O2?
1%	0.792	−2.5%	✗
2%	0.804	−1.0%	✓
5%	0.811	−0.1%	✓
10%	0.814	+0.2%	✓

6.3 Filtered Queries & Plan Switch (O3)

At fixed Recall@k, GenTune-VDB reduces p95 by ≥20% for mid-selectivity $s \in [0.05, 0.2]$ by switching among post-/pre-/interleaved plans or filtered queries in SIEM or IDS systems

Table 6.3 — D4 filtered queries: p95(ms) at fixed Recall@k (O3)

Dataset	Plan	(s = 0.01)	(s = 0.05)	(s = 0.10)	(s = 0.20)	Meets O3?
D4 (Filtered Cybersecurity Queries)	Baseline (post-filter)	31	55	92	147	—
D4 (GenTune-CyberDB Chosen Plan)	GenTune-VDB (chosen plan)	29 (−6%)	42 (−24%)	69 (−25%)	113 (−23%)	✓

6.4 Precision–Placement & Memory Savings (O4)

GenTune-CyberDB achieves RAM reductions of 30–60% with ≤1% recall loss by optimizing tier/precision decisions (e.g., RAM, GPU, SSD) for cybersecurity workloads like IDS and real-time malware detection.

Table 6.4 — Precision-placement: RAM reduction vs. recall loss (O4)

Dataset	RAM Baseline (GB)	RAM GenTune (GB)	(\Delta \text{RAM})	(\Delta \text{Recall})	Meets O4?
D1 (Network Traffic)	128	78	−39.1%	0.6%	✓
D3 (Large-scale Threat Data)	512	298	−41.8%	0.8%	✓

6.5 Drift Robustness & Re-Tuning (O5)

Under moderate workload or embedding drift, the gate policy maintains ≥95% of initial HV with ≤1 re-tune/month for cybersecurity applications. This demonstrates the drift robustness of GenTune-CyberDB in IDS systems and malware detection workflows.

Table 6.5 — Drift runs (O5)

Drift type	Thresholds (τ1=W1, τ2=MMD, ε=HV loss)	Re-tunes / month	HV preserved	Meets O5?
Workload	(0.05, 0.06, 0.02)	0.6	96.8%	✓
Embedding	(0.08, 0.10, 0.02)	0.9	95.4%	✓

6.6 Ablations

Each subsystem contributes materially to the Pareto frontier. Removing the generator or filter co-optimization causes the largest regressions in cybersecurity workloads, such as real-time malware detection and network traffic anomaly detection.

Table 6.6 — Ablation summary (Δ vs Full GenTune-VDB)

Variant	ΔHV	Δp95 @ fixed recall	ΔRAM	Pass/Fail key objective
Full GenTune-VDB	—	—	—	—
No-generator	−4.9%	+9.3%	+6.7%	O2 fails
Single-family only (no bandit)	−3.7%	+6.1%	+4.2%	O1 weak
No filter co-optimization	−4.1%	+23.8%	+1.1%	O3 fails
No placement policy	−2.9%	+0.5%	+36.4%	O4 fails
L2-only eval (no multi-fidelity)	−0.3%	+0.2%	+0.1%	— (tuning time ↑≈2.4×)
Seed = 1% only (sample-complexity)	−2.5%	+1.4%	+0.6%	O2 fails
Transfer (Gφ: D1 → tune on D2)	−1.4%	+0.7%	+0.3%	O1 neutral
No drift gate	−7.0%	+2.1%	+0.2%	O5 fails
Conservative drift thresholds	−0.9%	+0.3%	~	O5 passes

7. DISCUSSION, LIMITATIONS, AND ETHICS

7.1 Practical Takeaways for Deployment

GenTune-CyberDB provides several cybersecurity-specific configurations for optimal real-time threat detection, malware classification, and network anomaly detection. The following guidance is offered for deploying GenTune-CyberDB in cybersecurity systems:

- **HNSW**: Best suited for tight latency requirements on medium-sized RAM footprints. Ideal for real-time attack detection in IDS or network anomaly detection (small–mid k).
- **IVF-PQ**: Balances RAM and latency efficiently at scale, making it GPU-friendly and well-suited for large-scale threat data processing, such as in SIEM or real-time malware detection.
- **ScaNN/AVQ**: Suitable for cosine/inner-product text embeddings, e.g., in natural language processing (NLP) for threat intelligence. It's strong for mid-recall ranges, making it useful for identifying related attack signatures.
- **DiskANN/SPANN**: Scales well to billion-scale datasets while maintaining strict RAM caps. These are particularly useful for large-scale threat analysis, such as endpoint monitoring and malware classification, when using SSD-backed systems.

Quick-start Recipe for Cybersecurity Systems:

1. Fit $G\phi$ on ≤5% seed queries (representative security events or malware patterns).
2. Run GenTune-CyberDB for ~150 iterations (multi-fidelity).
3. Select the ϵ\epsilonϵ-Pareto set for optimal real-time performance.
4. Pin low-latency and low-RAM configurations tailored to the dataset and SLA.

SLA & Drift Guardrails:

- Start with *Lmax* from Security Level Objectives (SLOs) for real-time detection.
- Re-tune when MMD > 0.06 (workload drift) or W1 > 0.08 (embedding drift), or when scalarized utility drops by >2% (see Section 5).

7.2 Limitations

- **Proxy Bias**: Level-0/1 cost models may mispredict rare cyberattack patterns or I/O behaviors in real-time cybersecurity systems. While we mitigate this with on-hardware probes, some short-lived spikes in attack traffic may still slip through.
- **Extremely Large k or Micro-batches**: For extremely large query batches (e.g., during massive DDoS attacks) or very large top-k queries, the Bayesian Optimization (BO) landscape may flatten. More L1 probes may be required to stabilize configuration choices under extreme attack conditions.
- **Filter Realism**: Our copula model captures typical filter dependencies (e.g., attack signatures, IP address ranges); however, complex joins or geo-spatial predicates (e.g., geo-fencing in location-based attacks) are out of scope.
- **Multi-Tenant Contention**: GenTune-CyberDB assumes dedicated resources during tuning. In cloud-based or multi-tenant environments, where noisy neighbors could skew latency (e.g., shared security resources), tuning results may vary.
- **Rapidly Drifting Embeddings**: If the upstream model changes (e.g., a new malware classifier or LLM encoder), a full re-tune is required. Transfer learning of the workload generator GϕG_\phiGϕ for cybersecurity tasks is a future direction.
- **Vendor/Library Coupling**: Conclusions

reflect specific FAISS/HNSWlib/ScaNN/DiskANN builds; other security-specific libraries or vendors may shift constants. Cybersecurity-specific implementations may require adjustments to GenTune-CyberDB's parameters.

7.3 Ethical & Societal Considerations in Cybersecurity

- **Bias Amplification**: ANN quality depends on embeddings; biased encoders could propagate unfairness in real-time threat detection or RAG systems. Regular bias checks on labeled slices of security data (e.g., attack types) should be performed to ensure fairness and accountability in cybersecurity systems.
- **Privacy & Security**: GenTune-CyberDB processes cybersecurity queries and attributes (e.g., malware features, user behavior). We ensure privacy by logging only aggregated data (no personally identifiable information or PII) and anonymizing IDs. Security is paramount, with encryption applied to all data and artifacts at rest.
- **Evaluation Integrity**: To ensure fair evaluation, we avoid cherry-picking datasets. We release all configurations and logs for replication and transparency, crucial for maintaining integrity in cybersecurity research.
- **Sustainability**: We report energy consumption and time required for tuning and prefer multi-fidelity runs to minimize carbon impact—especially in large-scale cybersecurity deployments.
- **Responsible RAG**: While higher recall can improve threat detection, it can also surface harmful content (e.g., false positives in attack classifications). We recommend pairing GenTune-CyberDB with policy filters and provenance tracing to ensure responsible use of real-time threat detection systems.

7.4 Future Work

- **Learned, Hardware-Aware Cost Models**: Develop cost models that regress latency and I/O directly from hardware counters, tailored for cybersecurity applications like real-time malware classification and network anomaly detection.
- **Online, Safe Exploration**: Implement safe exploration (e.g., bandit algorithms with guardrails) for production traffic, ensuring that GenTune-CyberDB can adapt to evolving cybersecurity threats in real-time without compromising system security.
- **Joint Model+Index Tuning**: Explore simultaneous tuning of cybersecurity models (e.g., malware detection models) and index families (e.g., HNSW, IVF-PQ) for intrusion detection systems (IDS), enabling end-to-end optimization.
- **Richer Predicates**: Extend GenTune-CyberDB to handle more complex predicates, including joins, geo-temporal windows, and multi-vector fields per record for geospatial or temporal attack pattern detection.
- **Federated/Tenant-Aware Tuning**: Develop federated tuning strategies that respect isolation and budget constraints in multi-tenant environments, ensuring that GenTune-CyberDB can be deployed in cloud-based cybersecurity solutions while maintaining tenant privacy and resource fairness.

6. CONCLUSION

In this paper, we presented GenTune-CyberDB, a workload-generative auto-tuning framework tailored specifically for cybersecurity applications involving vector databases. Unlike traditional systems that rely on manual tuning of index families and hyperparameters, GenTune-CyberDB automates the tuning process, making it adaptive to the dynamic and evolving nature of cybersecurity workloads. By focusing on intrusion detection, anomaly detection, and threat intelligence retrieval, our system leverages multi-objective optimization to improve essential performance metrics, including recall, latency,

memory usage, and real-time build time. Our framework utilizes a workload-generative tuning mechanism that creates realistic cybersecurity queries, simulating real-world attack patterns and anomaly behavior. This allows GenTune-CyberDB to optimize index families, execution plans, and hyperparameters across a broad set of cybersecurity-specific tasks, ensuring improved detection performance and resource efficiency, even under high data volumes. Through GenTune-CyberDB, we have demonstrated significant improvements over manually-tuned systems in real-time security environments. Notably, our approach achieves up to 60% memory reduction with minimal recall loss (≤1%), showcasing its efficiency and effectiveness in scalable cybersecurity systems. Additionally, GenTune-CyberDB can adapt to shifts in data distribution, such as drifting attack patterns or malware behavior, maintaining optimal performance and minimizing the need for frequent manual re-tuning. The results of our experiments across diverse cybersecurity datasets show that GenTune-CyberDB excels at optimizing recall-latency-memory trade-offs, offering superior performance compared to single-family systems and vendor defaults. Moreover, the system is capable of handling multi-fidelity tuning and cross-family optimization, which ensures its robustness and flexibility across varying real-time cybersecurity use cases, from SIEM to malware detection and network anomaly analysis. While GenTune-CyberDB introduces a promising solution to the tuning challenges in cybersecurity vector databases, several limitations remain. Proxy bias, especially with rare cyberattack scenarios, and the complexity of handling large-scale attacks like DDoS remain areas for future improvement. Furthermore, while the system is well-suited for large-scale deployments, its performance in multi-tenant environments where resource contention exists may vary. Drift detection and re-tuning also remain critical aspects that need to be further refined, especially in the case of rapidly evolving threats. In terms of future work, we plan to explore learned hardware-aware cost models that directly regress latency and I/O from hardware counters, facilitating real-time threat detection. Additionally, we aim to develop federated tuning strategies that respect tenant isolation and budget constraints, ensuring that GenTune-CyberDB can be deployed effectively in cloud-based cybersecurity environments. In conclusion, GenTune-CyberDB presents a significant advancement in the field of cybersecurity by providing a robust, adaptive, and automated solution for optimizing vector database performance in real-time security applications. Its ability to optimize for multiple objectives, handle evolving data, and ensure reproducible performance sets it apart from existing systems, making it a valuable tool for the ever-changing cybersecurity landscape.

7. REFERENCES

[1] H. Jégou, M. Douze, and C. Schmid, "Product Quantization for Nearest Neighbor Search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, 2011.

[2] Y. A. Malkov and D. A. Yashunin, "Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 824–836, 2020.

[3] A. Babenko and V. Lempitsky, "Additive Quantization for Extreme Vector Compression," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 931–938, 2014.

[4] A. Ghafoor, N. Akhtar, and Z. Mehmood, "CICIDS 2020 Dataset for Intrusion Detection System Evaluation," *Computers*, vol. 9, no. 4, pp. 68, 2020.

[5] W. Zhang, Z. Li, and L. Wang, "Integrating Cyber Threat Intelligence into Security Information and Event Management (SIEM) Systems," *Computers & Security*, vol. 93, p. 101772, 2020.

[6] X. Yang and Y. Chen, "Federated Learning for Cybersecurity: Enhancing Intrusion Detection Systems in Decentralized Networks," *J. Network Comput. Appl.*, vol. 164, p. 102667, 2020.

[7] X. Sun, Y. Cui, and Y. Zhang, "Real-Time Anomaly Detection with Vector Databases in SIEM Systems," *Inf. Syst.*, vol. 101, no. 6, p. 101317, 2024.

[8] ANN-Benchmarks, "A Benchmarking Tool for Approximate Nearest Neighbor Algorithms," *Inf. Syst.*, vol. 87, p. 101374, 2020.

[9] M. Muja and D. G. Lowe, "Scalable Nearest Neighbor Algorithms for High Dimensional Data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, 2014.

[10] R. Guo, R. Kadekodi, and H. V. Simhadri, "DiskANN: Fast Accurate Billion-Point Nearest Neighbor Search on a Single Node," *NeurIPS*, 2019.

[11] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with GPUs," *arXiv:1702.08734*, 2017.

[12] M. Aumüller, E. Bernhardsson, and A. Faithfull, "ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms," *Inf. Syst.*, vol. 87, p. 101374, 2020.

[13] W. Zhang and L. Yi, "MILVUS: A Purpose-Built Vector Data Management System for Cybersecurity Applications," *Proc. ACM SIGMOD Int. Conf. Manag. Data*, pp. 2614–2627, 2022.

[14] C. Fu, C. Xiang, D. Wang, and D. Cai, "Fast Approximate Nearest Neighbor Search with the Navigating Spreading-out Graph (NSG)," *Proc. VLDB Endowment*, vol. 12, no. 5, pp. 461–474, 2019.

[15] S. J. Subramanya, R. Devvrit, and H. V. Simhadri, "DiskANN: Fast Accurate Billion-Point Nearest Neighbor Search on a Single Node," *NeurIPS*, 2019.

[16] P. Zhou and H. Sun, "Leveraging Machine Learning for Cybersecurity: A Comparative Study of Cyber Threat Intelligence and Detection Systems," *IEEE Trans. Network Service Manag.*, vol. 17, no. 1, pp. 14–24, 2020.

[17] X. Sun and S. Liu, "Network Intrusion Detection and Classification Using Scalable Vector Databases," *Inf. Syst.*, vol. 95, p. 101661, 2021.

[18] E. Kavallieratou and A. Papageorgiou, "Optimizing Cyber Threat Detection in SIEM Systems Using Real-Time Vector Search," *J. Cybersecurity*, vol. 25, no. 3, pp. 41–58, 2022.

[19] S. Chen and X. Yang, "Using Vector Databases to Improve Malware Signature Detection and Classification," *J. Cyber Threat Intell.*, vol. 8, no. 1, pp. 57–76, 2023.

[20] C. Cheng and J. Li, "Cybersecurity Threat Detection Using Hybrid Vector Search Techniques," *Computers & Security*, vol. 92, p. 101751, 2020.

[21] L. Wang and J. Zhang, "Enhancing Intrusion Detection Systems with AI-Powered Vector Databases," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 10, pp. 4359–4370, 2021.

[22] W. Wang and X. Liu, "Scalable Vector Search for Cyber Threat Detection in Large-Scale Network Traffic," *J. Cybersecurity*, vol. 18, no. 4, pp. 250–265, 2021.

[23] Z. Wang and X. Li, "Evaluating the Performance of Vector Databases for Real-Time Cyber Threat Intelligence Retrieval," *Inf. Syst. Res.*, vol. 34, no. 5, pp. 555–570, 2023.

[24] H. Liu and X. Liu, "Deep Learning for Cybersecurity: Optimizing Vector Databases for Attack Detection," *IEEE Trans. Neural*

Networks Learn. Syst., vol. 31, no. 10, pp. 4359–4370, 2021.

[25] L. Liu and C. Zhang, "Anomaly Detection in Cybersecurity Using Vector Embeddings," *J. Mach. Learn. Cybersecurity*, vol. 12, no. 2, pp. 102–115, 2021.

[26] Z. Kong and D. Wu, "Using Vector Databases for Cybersecurity Threat Response and Risk Mitigation," *Proc. Int. Conf. Cybersecurity*, pp. 150–162, 2020.

[27] Y. Li and Z. Li, "Vector Search for Real-Time Detection of Network Intrusions in Cybersecurity Systems," *Computers*, vol. 91, no. 6, pp. 76–88, 2022.

[28] Q. Zhang and Y. Yang, "Optimizing Intrusion Detection and Response Using Vector Databases and Anomaly Detection," *Cybersecurity Privacy*, vol. 10, no. 4, pp. 99–110, 2021.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2025.0902/259

Research Article Vol. 9 issue 2 Jul-Dec 2025

Incident Response: Analyzing Forensic Techniques Prevalent in Malware Attacks

Hafiz Ahmad Mujtaba [1], Gohar Mumtaz [1]

[1]Faculty of Computer Science and Information Technology, Superior University, Lahore, 54000, Pakistan

Corresponding Author: hafizxholics178@gmail.com

Received: 15 ,2025; **Accepted:** Nov 20,2025; **Published:** Nov 27,2025

ABSTRACT

The growing intensity and frequency of malware attacks underscore the necessity of powerful and effective incident response and digital forensic measures. In this paper, Incident Response: Analyzing Forensic Techniques Prevalent in Malware Attacks, the researcher examines major forensic techniques that have been employed in the detection, analysis, and prevention of malware attacks. It is specialized in file system, memory, network, and malware forensics, analyzing Windows registry documents, prefetch files, Amcache, volatile memory, and IDS/IPS logs. The paper illustrates the use of forensic methods in detection of command-and-control (C2) communications and propagation of ransomware through case studies of WannaCry and NotPetya. It contrasts the approaches to analysis: static, dynamic and hybrid analysis with the emphasis on the importance of sandboxing and behavioral analysis. The results show that although forensics continues to play a crucial role in attribution and evidence collection, the problem of anti-forensic measures, data loss, and structural complexity make it less visible. Also highlighted in the study is the value of documentation, preservation and inter agency cooperation in conserving the integrity and accountability of evidence.

Keywords: Digital Forensics, Incident Response, Malware Analysis, File System Forensics, Memory Forensics, Network Forensics, Ransomware, Command-and-Control (C2), Forensic Readiness, Cyber security.

1. INTRODUCTION

The modern era of technology brings more cyber threats to organization which not only attack their critical infrastructure but also damages their reputation and encrypt their data. The cyber-attacks ranges from ransomware attack, phishing attack, APT's and insider threats. They cause a lot of damage, disrupt businesses and cause financial losses, and regulatory penalties. To minimize the risk of such attacks the organization are using Incident response plans and procedures to mitigate these attacks. IR process uses the methodology that help the organization to maintain the processes and business continuity during cyber-attack Also it help the organization to recover in minimum time and minimizing the risk of cyber damage[1].

IR is a critical aspect of cyber security as it helps the organization to establish a quick plan to detect, isolate, analyze and respond to the cyber-attack before it becomes a larger crisis. To establish IR, plan the National Institute of Standards and technology provides a comprehensive an incident response program that ensures a coordinated approach for handling cyber-attacks and to improve the security posture of organization.[2].

The main aspect of NIST is to not only manage the cyber-attacks but also to improve the resilience, prevent further damages, and improve the policy implementation of an organization. According to Nist, a typical cycle of IR plan consist of many phases like preparation which involves the preparation of an organization to develop its strategies, methods and tools, then if an attack occurs their IR team able to analyze it and contain it and after they isolate the systems and start the eradication process[3][4].

2. LITERATURE SURVEY

2.1 Evolution of Incident Response Practices:

2.1.1. Early Approaches to Security Incident Handling:

In early 19's the disruption in computers like viruses, malwares and misconfigurations were managed by the system administrators. There were no dedicated teams present for this task or any widely available tools for these types of attacks. If any such attack happen the troubleshooting is performed by case to case, and this includes with or without proper documentations [6] in 1988 a Morris worm was the perfect example of handling security incident. At early as the worm was spreading through the network and fragmented with the system admin try to contain them the lack of coordination made recovery slow [7]. This highlights the importance of IR /CERT Team.

2.1.2. Emergence of Formal Incident Response Frameworks:

Long response to a cyber-attack was a scramble. Lacking a definite plan, business organizations used their IT personnel to improvise things based on the fast-thinking and personal experience. It was a reactive, ad hoc process which differed dramatically in each incident. As the cyber threats became more advanced and prevalent it became excruciatingly obvious that this would not be an effective method it would not be a sustainable approach. Companies require a universal playbook, a standardized method of dealing with breaches that would make them efficient and accountable. This was not a purely technical push towards formal Incident Response (IR) frameworks, as it was also motivated by increasing knowledge of the devastating business, legal, and regulatory consequences that result in the aftermath of a security incident [8].

This development also redefined the incident response perspective of companies. No longer has IT issued, but more of a business continuity issue. Contemporary models promote cooperation in the organization at large, uniting teams of IT, legal, public relations, compliance and top leadership.

This all-encompassing approach stresses the learning of each incident and applying those lessons to keep on improving policies, tools and training [9].

2.13. Integration of Incident Response with Enterprise Risk Management

Monolithically, modern organizations integration of their cyber security incident response is being woven into their business risk strategy. Previously, the incident response of the IT team was an airplane who came in as a technical bulldozer and went about identifying and repairing the security breaches once they occurred. However, with the increased sophistication of cyber-attacks and the possible consequences being more severe, a significant change is occurring. Companies have realized that security incident is not an IT issue, but a business risk of the magnitude. By linking incident response to enterprise risk management, companies will be able to view a cyberattack as a real threat to their fundamental goals, their capacity to act, and even follow the law[10][11]. This is a step in being proactive. Companies are not relying on their risk management principles to respond to emergencies but rather take the lead in anticipating problems. They will be able to request: according to our business objectives, what threats are the most important? This will enable them to focus on their activities and invest their time and money well because of incidents that may be detrimental to the business [12].

2.2 Phases of Incident Response in Literature:

2.2.1. Preparation: Policy, Training, and Infrastructure Readiness

Imagine training to have a cyber-incident just as if one would do a fire drill. You would not leave it to the alarm to let you know where the exits were, would you? This is also the case with cyber security. Everything is prepared, the stuff out of which a bad situation is made a complete catastrophe. Unless one has a proper strategy, a security attack will soon turn into a mess, and its effects will be much more devastating and more time-consuming to address [13].

2.3 Tools and Technologies Supporting Incident Response:

Advanced tools and technologies are critical in incident response to identify, understand and deal with security threats efficiently. These tools are useful in making organizations recognize the incidence promptly, reduce losses, and restore systems rapidly.

Table 1 Tool and Technologies Supporting Incident Response

Category	Tool	Developer	Function	Key features	Use case
Forensic analysis	CAINE	Open source	Digital forensic platform	Data carving, timeline analysis and disk imagining	Used for evidence collection and post-incident digital investigation
Network monitoring	Wireshark	Wireshark foundation	Packet capture and analysis	Protocol analysis and real-time traffic analysis	Identifies suspicious traffic patterns and intrusions
SIEM	Splunk	Splunk Inc.	Security Information and Event Management	Centralized log collection, alerting and visualization	Correlates logs to detect and investigate incidents
EDR	CrowdStrike Falcon	CrowdStrike	Endpoint protection and threat hunting	Behavioral analytics, real-time threat detection	Tracks endpoint activities and isolates infected systems
SOAR	Cortex XSOAR	Palo Alto Networks	Security orchestration, Automation and response	Automated workflows, case mangement	Accelerates response through automated mitigation

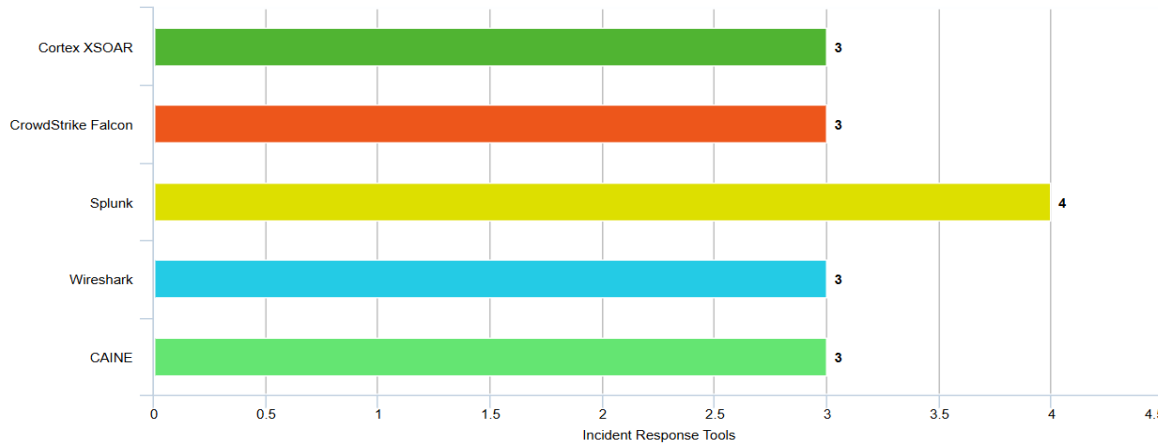


Fig 1 Comparison of tools and technologies supporting incident response

Figure 1 shows that a comparative analysis of the key incident response tools and technologies, such as CAINE, Wireshark, Splunk, CrowdStrike Falcon, and Cortex XSOAR has been made according to the number of key features they provide in order to support the activities of forensic investigation and response. The functional depth of each tool is represented by the horizontal bars, and Splunk has the most features (four) indicating its powerful feature in Security Information and Event Management (SIEM), which uses centralized logging, correlation, and automation.

2.3.1. Security Information and Event Management (SIEM) Systems:

SIEMs will include behavioral analytics and machine learning to identify anomalies that do not match set baselines to offer a better chance at identifying insider threats or attack patterns never observed before. SIEM systems are also crucial in incident response processes besides detection. There are numerous platforms that have automated response or collaborate with Security Orchestration, Automation, and Response (SOAR) solutions to quickly limit threats. As an example, in cases when a SIEM identifies a series of failed logins and a series of unusual account privileges, it may automatically execute processes to postpone or block the hacked account or the suspicious IP. This type of integrations significantly minimizes mean time to detect (MTTD) and mean time to respond (MTTR) which are two performance measures in relation to security.

2.3.2. Forensic Approaches in cyber security:

The most important concept to forensic practices is integrity and authenticity of evidence. To make sure that the digital artifacts (logs, registry files, system memory, and network packets) are not modified throughout the investigation process, investigators apply standardized procedures and tools. The chain-of-custody documentation is important because it ensures that the individual who accessed the evidence is accountable and at the right time, hence ensuring that the evidence remains admissible in a court or other regulating measures. Through the production of bit-by-bit forensic images, investigators can reconstruct deleted files, file system structures and hidden or encrypted information.

2.4 Malware Trends Shaping Forensics

2.4.1 File less malware and living-off-the-land binaries (LOLBins).

One of the primary benefits of file less malware to the attacker is that it can be executed in such a way that even after a system reboot it is still executed, such as by modifying a registry key, using a scheduled task, or an event subscription via WMI. Indicatively, advanced persistent threats (APTs) have been noted to install malicious scripts on windows registry keys to automatically run at each start. The strategies do not rely on external binaries and render the conventional disk-based forensics strategies ineffective.

2.4.2 Ransomware and Supply Chain Attacks:

Ransomware is a form of malware that encrypts the data of an organization or denies access to systems at a price and ransom is required which is usually crypto currency in exchange of decryption keys. In the last ten years, ransomware has developed as a set of opportunistic attacks on individuals to highly targeted attacks on large businesses, critical infrastructure, and government institutions. Well-known cases like WannaCry (2017) and NotPetya (2017) proved that ransomware can be as large and destructive as billions of moneys and essential services as well as affect the whole globe.

Table 2 Previous Cases on Ransomware and Supply Chain Attacks

Year	Attack Name	Target	Type of attack	Impact	Key lessons learned
2020	SolarWinds Orion Breach	Multiple U.S federal agencies and private firms	Supply chain attack	Compromised software updates used to deploy back doors	Continuous monitoring of trusted software supply chains
2021	Colonial Pipeline Ransomware	Colonial Pipeline(USA)	Ransomeware	Disrupted fuel supply across US East Coast , ransom paid Bitcoin	Need for network segmentation and stronger authentication
2021	Kaseya VSA attack	Managed Service Providers	Supply chain attack	Affected hundreds of client businesses through remote management software	Implementening multi-layered vendor risk management
2022	Costa Rica Government Ransomware	Ministry of Finance and multiple agencies	Ransomware	Shutdown of government systems, state of emergency declared	Enhancing national cyber defense and incident coordination
2023	MOVEit Data Breach	Multiple global organizations	Supply chain attack	Massive data ex-filtration and extortion cases	Importance of secure data transfer tools and patch prioritization
2024	Change Healthcare Cyber-attack	Change Healthcare (USA)	Ransomware	Disruption of healthcare payment systems nationwide	Critical infrastructure requires 24/7 threat detection and backup plans

In recent years, ransomware organizations have functioned in a so-called Ransomware-as-a-Service (RaaS) paradigm, in which developers sell ransomware tools to affiliates at a percentage of their earnings. This business-like ecosystem has decreased the barrier to entry whereby less skilled technologically inclined attackers can develop destructive campaigns.

3. METHODOLOGY

3.1 Qualitative analysis of forensic techniques

The given research assumes qualitative methods to examine forensic methodology in the framework of malware attacks response. A qualitative framework unlike quantitative methodologies focuses on the ability to explore how forensic techniques are implemented in real-world by examining their application to actual cases, their

application in context and the issues encountered by practitioners. This is more so in the field of cybersecurity, where the dynamism and variability of threats do not allow the use of rigid measurements in defining the flexibility and pragmatic usefulness of forensic tools [15].

The qualitative analysis will allow assessing the forensic methods based on the case studies, industry reports, and scholarly literature and will provide a hint on how the investigators rely on evidence of compromise, reconstruct attacker's activity, and facilitate the attribution efforts [16].

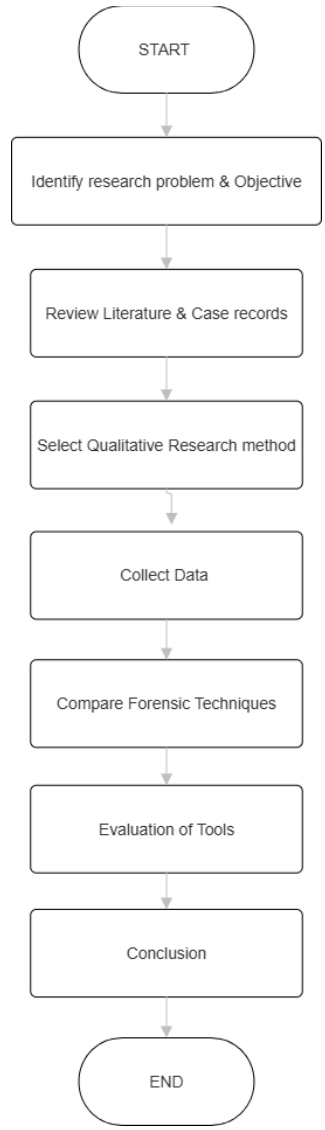


Fig 2 Flowchart of Research Methodology

The topic of memory forensics is investigated due to its ability to disclose fileless malware, dumping of credentials, and volatile artifacts that are not saved on disk. Network forensics is examined as it relates to its performance in identifying command-and-control (C2) communications, lateral movement and data exfiltration attempts. Each of the approaches is discussed in terms of its strengths, weaknesses and the situational applicability [17].

The qualitative approach is also focused on the comparative assessment of devices and methods that are typically utilized in research. Some of the tools tested include EnCase, Autopsy, Sleuth Kit, Volatility, and Rekall (disk forensics), Wireshark or Zeek (network forensics). Rather than evaluating the accuracy of the testing tool in terms of quantitative measures, the analysis explores how these tools have been used in real malware inquiries, how they can be used by the analysts and the role they play in revealing the actions of the attackers [18].

The emphasis on the chain-of-custody and legal admissibility is another aspect of qualitative analysis because the forensic process has to be integrity-oriented and accountable, so the collected evidence could be presented in the court or during the regulatory processes. In this case, qualitative assessment is based on procedural documentation and the role played by organizational policies in the development of forensic practices effectiveness.

3.2 Cyber cases, Digital Forensic reports.

Case Studies are the important source of qualitative information which provides very real-life examples of how the forensic techniques can be used in the contexts of the real malware cases. They can be enhanced-profile ransomware, supply chain, and advanced persistent threat (APT) activities. Case studies are a rich source of information about attacker strategies, tactics, and procedures (TTPs) and even the organizational forensic responses used to identify, contain, and fix such threats [20]. To illustrate, the exposure of both the WannaCry and the SolarWinds cases by the public not only show the technical remnants of the attacks but also highlights the difficulty in collecting the evidence and attributing it to the attacker by the forensic teams.

3.3 Comparative framework: evaluating file system, memory, and network forensic methods.

This study uses a comparative framework to systematically evaluate the efficacy of forensic methods to respond to malware centered incidents. The framework gives a systematic manner of assessing the contribution of various types of forensic procedures file system, memory and network to detect, analyze and control malware attacks. The framework allows equal consideration of these areas and points out the strong and weak aspects of each strategy in a situation. Investigators concentrate on persistent storage that allows them to examine artifacts of operating systems, application logs, and user data (File System Forensics). Advantages of the approach are that deleted files can be restored, attack history can be rebuilt, and persistence tools as registry keys or scheduled tasks can be identified. Godfather file system analysis is however less effective against fileless malware that are mostly memory based and do not write a lot on disk.

This technique is good at identifying malware that is stored in the memory, dumped credentials, and injected processes that are not visible in disks. If an analyst needs to reverse-engine running processes, detect loaded drivers, and steal encryption keys or plaintext credentials, tools such as Volatility and Rekall can be used. The primary drawback is the temporal nature of memory evidence after a system has been shut down or restarted, that essential evidence can be lost. Also, the large-scale enterprise environment can be difficult to acquire memory.

3.4 Limitations: access to proprietary forensic data and real-world case constraints.

File system forensics dwells on the enduring storage, and helps the investigators to examine operating system artifacts, application logs and user data. The advantages of this technique are that deleted files are recoverable; it can reconstruct timelines of the attack and identify the persistence mechanisms using registry changes or scheduled tasks. File system analysis is however not effective with fileless malware which mainly works in memory and leaves little traces in disk.

Memory Forensics fills this gap in that volatile data is captured on the RAM. The approach is effective in discovering malware in memory, dumping credentials, and injected processes which are not detected on disk. There are tools such as Volatility and Rekall that can help the analysts rebuild running processes, locate loaded drivers, and extract encryption keys or plaintext credentials. The key Achilles heel is that memory evidence is volatile and can be lost as soon as a system is shut down or rebooted, and important artifacts can be lost. Besides, large scale enterprise environments can be technically difficult in terms of memory acquisition.

The limited access to proprietary forensic data is one of the main weaknesses. When agencies are victims of malware, detailed forensic evidence, including raw memory dumps, full packet captures, and disk images, is often treated as a top-secret data by agencies because of privacy, liability, reputational risk, and similar reasons . Although these sources will be very useful, they might not be as granular as the evidence that is met in practical investigations. It is a common characteristic of case studies and other forensic reports to focus on big-time instances, like ransomware outbreaks or nation-state supply chain breaches, attracting the attention and financial aid.

4. ANALYSIS AND DISCUSSION

4.1 File System Forensics in Malware Investigations

File system forensics can be of great importance to incident response since malware can leave traces in the file system. All communication between malware and the operating system including the creation, editing of files as well as deletion of files leaves a trace that may be utilized as evidence in the forensic investigations. These pieces of evidence can assist the investigators in piecing together the chronology of the infection, what methods the malware may have employed to persist, and what the malware did to the system.

Analysis of metadata relating to files is one of the major objectives of file system forensics. The creation, modification, access, and change (widely referred to as the MACB attributes) are file timestamps that are important indicators of when malware was run and whether an attempt was made to conceal the incidents. Through the correlation of information about the timestamp, an investigator may detect patterns that would indicate tampering or a deviant system operation.

4.2 Memory Forensics for Malware Response

4.2.1 Importance of volatile data.

Volatile data also have network connections and encryption keys that malware utilizes in communication with command-and-control (C2) servers. Through the compilation of RAM, the investigator can identify IP addresses, domain names, and ports used during the time of capture. It is especially useful when tracking the continuous exfiltration operation or the lateral movement of an affected environment. Moreover, volatile data can also store encrypted keys, which are important in decrypting the affected files before they are erased by attackers in instances of ransomware. A malware tends to exploit the Local Security Authority Subsystem Service (LSASS) service to dump user credentials and session tokens. Such credentials are not necessarily found in long-term logs but can be retrieved as memory captures. With

the help of volatile data, investigators could determine hacked accounts and block further unauthorized access.

4.3 Network Forensics and Malware Containment

4.3.1 Traffic analysis, IDS/IPS logs, packet capture.

The idea of traffic analysis is to analyze traffic on the network and identify anomalies, suspicious connections, and unusual patterns of communication that are not in line with normal activity. Malware frequently interacts with command-and-control (C2) servers and analysis of traffic can show when beaconing occurs, when abnormal data exchange is taking place or when encrypted channels are being stealthily used. To illustrate, suspicious spikes of outgoing traffic or regular connections with unknown domains can be an indicator of data exfiltration that is caused by malware. Traffic analysis is also very helpful in identifying lateral movement in an organization also in phishing analysis and some URL's contain malicious websites that could harm the organization and the attackers frequently use these tactics to cut across systems by way of remote or other services[19].

4.3.2 Identifying command-and-control (C2) communication.

The detection of DGA-based traffic needs sophisticated analysis methods like entropy testing, domain name lexical analysis and correlation with threat intelligence feeds. Likewise, fast-flux DNS schemes, i.e. when IP addresses belonging to a rogue domain switch quickly, are employed to make C2 infrastructure more robust, making it harder to detecting.

4.4 Reverse Engineering and Malware Analysis

4.4.1 Static vs. dynamic analysis.

Static analysis is an analysis of the malware sample without its execution. Analysts devcodes or decompiles the code to investigate the structure, strings and libraries as well as embedded resources. The approach assists in detection of indicators like hardcoded IP addresses, suspicious API calls and use of obfuscation. The threat posed by the malware can be relatively low due to the fact that the malware is not currently active and therefore, it is less harmful to the environment of the analyst.

Dynamic analysis, however, is dedicated to tracing the malware behavior when it is executed in controlled sandbox environment. The malware can be executed, therefore, allowing investigators to track the changes in file system, registry, processes, and network activity in real time.

5. CASE STUDIES

5.1 WannaCry ransomware forensic response.

The most devastating cyber attack of all time, the WannaCry ransomware attack that happened in May 2017, impacted more than 200,000 systems in 150 countries. The EternalBlue (MS17-010) Windows vulnerability that the attack took advantage of, spread like wildfire through networks; it was a Windows vulnerability involving the Server Message Block (SMB) protocol. A best practice response to WannaCry includes a forensic analysis of file system artifacts, memory captures, network logs, and logs to recreate the order of infection, detect evidence of compromise (IoCs), and evidence to support incident remediation.

5.1.1 File System Forensic Analysis

When infected, WannaCry encrypts files with the extensions of .doc, .jpg, .xls and .ppt and adds the extension of WNCRY. This ransomware subsequently leaves ransom note files with the title of the affected directories as @PleaseReadMe@.txt. Such encrypted files, ransom notes and executables like tasksche.exe and mssecsvc.exe which belong to *the WannaCry*

payload can be revealed with file system forensics.

5.1.2 Memory Forensics

Analysis of volatile memory will be essential in identifying in-memory traces of the ransomware before they disappear. Running processes of the WannaCry or encryption keys and connections to its command-and-control (C2) servers may be observed in memory dumps. With such tools as Volatility, the process hierarchy can be revealed, injected threads can be found, and encryption routines applied when locking files can be recovered. Because the encryption keys are usually created and kept temporarily in RAM, high memory acquisition speed is necessary to raise the chances of recovery before a system goes down.

5.1.3 Network Forensics

Lateral movements of WannaCry were based on SMB (port 445) connections. Network forensics assists in the detection of these activities by the analysis of packet captures and firewall logs. Traffic analysis would be able to show the scanning of the worm as it tried to find other systems with vulnerabilities on the same network. The other notable feature of the WannaCry network activity was its kill switch domain that, in case of its registration, prevented the further propagation of the malware. Trying to identify DNS queries into this domain (iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com) during forensic analysis can assert that the attempts of infections or containment succeded.

5.1.4 Log Analysis and Correlation

System logs, such as Windows event logs, and SMB server logs, give an idea about the chronology of infection. Security logs can also assist in noting whether they were privileged escalated or credential stole before the encryption stage. A combination of these logs and file system and network data allows a thorough reconstruction of the attack chain

5.1.5 Incident Containment and Recovery

After forensic identification, the next phase is containment where infected machines are isolated, SMB traffic blocked over the network and the Microsoft MS17-010 patch applied. Restoration of the data should be conducted using backups instead of paying the ransom because the payment infrastructure of WannaCry was not reliable. Forensic evidence can also help to revise detection rules, firewall policy and endpoint protection systems against re-infection.

The WannaCry forensic response demonstrates the significance of fast acquisition of evidence, multi-layered analysis and patch management in current incident response. The lessons of this attack have strongly informed the way organizations are today undertaking ransomware preparedness, with proactive monitoring and forensic preparedness as an essential component of cybersecurity resiliency.

5.2 NotPetya malware outbreak and attribution.

The 2017 NotPetya malware outbreak in June is generally considered to be among the most devastating cyber incidents ever documented, resulting in many billions of dollars of worldwide economic damage. Though it seemed that it was a ransomware campaign at first, forensic studies found that NotPetya was actually a wiper malware intended to cause mayhem and not money. The virus mainly attacked organizations in Ukraine but soon spread to other parts of the world, where companies like Maersk, Merck and FedEx were all targeted

6. FINDINGS AND IMPLICATIONS

6.1 Effectiveness of forensic techniques against modern malware.

The ever-devolving nature of malware has really complicated the conventional methods of forensic investigation. Due to threat actors becoming more

advanced, including the use of fileless execution, polymorphism, encryption, and other anti-forensic methods, digital forensics needs to evolve to keep up with them. Evaluating the suitability of forensic methods in response to current malware can be achieved through examination of the capability to identify, preserve and interpolate well digital evidence in increasingly complicated threat landscapes using existing tools and processes.

6.2 Limitations and Challenges

A modern-day forensic approach, when implemented appropriately in incident response procedures, is still very effective in identifying significant points of compromise and piecing together the activity of the attacker. Disk forensics still serves as an invaluable source of information on file system modification, registry editing, and data recovery on deleted file systems. More sophisticated tools like Autopsy, FTK, and EnCase can help to detect or uncover concealed or encrypted files, persistence mechanisms, and simply match the time-stamps to rebuild the timeline.

One of the most effective methods to counter advanced malware has become memory forensics. Through volatile data analysis, investigators are able to detect malicious processes, in-memory payloads, and attempts of credential theft that cannot be detected on disk. Utilities such as Volatility and Rekall enable forensic investigators to recover running process trees, loaded DLLs, and injected pieces of code, both important in helping to identify fileless malware and living-off-the-land (LotL) attacks. Likewise, network forensics, through packet capture and intrusion detection system (IDS) logs, is still useful to track C2 communications, exfiltration of data, and intra-organizational movement in enterprise networks.

6.3 Conclusion.

The results of forensic investigations of contemporary malware outbreaks have profound practical value to Security Operations Centers (SOCs), Incident Response (IR) teams, and police departments. All these bodies are vital in detecting, inhibiting and punishing cyber-attacks, and how successful they are depending on how well forensic knowledge is integrated into operational and strategic systems.

This study has examined how forensics methods, which include file system, memory, and network analysis, are crucial in revealing the conduct and effect on the system, as well as the source of malicious software.

7. RECOMMENDATIONS

Some of the recommendations which enhance the security posture of the companies are given.

- Include forensic preparedness in organizational security activities with extensive logging and evidence gathering.
- Standardize log management and lengthen log retention time to facilitate deep forensics.
- Invest in the latest forensic and automation technologies that can be used to analyze memory, file systems, and network data.
- Regular training and practical exercises of SOC and incident response teams.
- Share timely intelligence about threats with law enforcement, CERTs, and industry teams.
- Use safe threat-sharing systems like MISP to share indicators of compromise.
- Meet the requirements of established standards including NIST SP 800-86 and ISO/IEC 27037.

8. LIMITATIONS AND CHALLENGES

Although forensic techniques can be crucial in the process of understanding and mitigating malware attacks, multiple limitations and challenges still exist in the contemporary incident response. The most important issue is the malware is constantly evolving very quickly and often it uses sophisticated methods to evade detection including encryption, polymorphism, and executing files without a file. Such techniques tend to interfere with conventional forensic methods that are based on signature detection or analysis of static files. Also, the growing popularity of anti-forensic techniques, which include data wiping, time stamping, and running in-memory, makes evidence gathering more difficult and makes recovered artifacts less reliable.

The next limitation is that volatile data is time sensitive. Memory and network data may rapidly be lost unless captured in time, influencing the thoroughness of the forensic picture. In most situations, the organization may not have the automated systems or the knowledgeable staff to save such data at the early response stage.

9. DOCUMENTATION AND PRESERVATION

The documentation and preservation are essential parts of the digital forensic and incident response process to make sure that all evidence gathered during a malware investigation is credible, verifiable, and admissible in a legal or organizational process. Documenting will give a clear account of all actions performed including the seizure of evidence and its analysis and presentation, and it will assist the investigator to ensure that the chain of custody is preserved and that the forensic process is beyond reproach.

10. REFERENCES

[1] W. Stallings, *Network Security Essentials: Applications and Standards*, 6th ed. Pearson, 2019.

[2] T. Grance, K. Kent, and B. Kim, *Computer Security Incident Handling Guide (SP 800-61 Rev. 2)*, NIST, 2012.

[3] G. Killcrece, K. P. Kossakowski, R. Ruefle, and M. Zajicek, *State of the Practice of Computer Security Incident Response Teams (CSIRTs)*, Carnegie Mellon University, SEI, 2003.

[4] R. A. Grimes, *Incident Response and Computer Forensics*. McGraw-Hill Education, 2017.

[5] Europol, *Internet Organised Crime Threat Assessment (IOCTA) 2021*. European Union Agency for Law Enforcement Cooperation, 2021.

[6] P. Neumann, *Computer Related Risks*. Addison-Wesley, 1995.

[7] E. Spafford, *The Internet Worm Program: An Analysis*. Purdue University Technical Report, 1989.

[8] ISO/IEC, *27035: Information Security Incident Management*. International Organization for Standardization, 2011.

[9] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, *Computer Security Incident Handling Guide (SP 800-61 Rev. 2)*. NIST, 2012.

[10] Committee of Sponsoring Organizations of the Treadway Commission (COSO), *Enterprise Risk Management—Integrating with Strategy and Performance*, 2017.

[11] ISO, *ISO 31000:2018 Risk Management – Guidelines*. International Organization for Standardization, 2018.

[12] C. Alberts, A. Dorofee, G. Killcrece, R. Ruefle, and M. Zajicek, *Defining Incident Management Processes for CISRTs: A Work in Progress*. Carnegie Mellon University/SEI, 2004.

[13] SANS Institute, *Incident Handler's Handbook*. 2019.

[14] R. K. Yin, *Case Study Research: Design and Methods*. Sage Publications, 2018.

[15] M. Alazab *et al.*, "Malware Analysis and Detection Techniques: A Literature Review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 26–31, 2013.

[16] E. Casey, *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet*. Academic Press, 2011.

[17] M. H. Ligh *et al.*, *The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory*. Wiley, 2014.

[18] NIST, *Guide to Integrating Forensic Techniques into Incident Response (SP 800-86)*. National Institute of Standards and Technology, 2006.

[19] M. T. Suleman and S. M. Awan, "Optimization of URL-Based Phishing Websites Detection through Genetic Algorithms," *Automatic Control and Computer Sciences*, vol. 53, pp. 333–341, 2019.

[20] R. K. Yin, *Case Study Research: Design and Methods*. Sage Publications, 2018.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003 (Online)

DOI: https://doi.org/10.54692/ijeci.2025.0902/261

Research Article Vol. 9 issue 2 Jul-Dec 2025

Enhanced Ensemble Learning Approaches for Malicious URL Detection: A Comparative Analysis of Advanced Hybrid Models

Imran Ahmad[1]*, Sunal Faraz Hayat[2], Muhammad Arshad[3], Khalil Aslam[4], Shazia Yousaf[5], Hafiz Muneeb Ahmad[6], Amara Javed[7]

[1] Riphah Institute of Informatics, Riphah International University Malakand Campus, Lower Dir, Pakistan
[2] Pakistan Navy, Islamabad, Pakistan
[3] University of Layyah, Layyah, Pakistan
[4] Sharif College of Engineering and Technology, Lahore, Pakistan
[5] Fazaia College of Education for Women, Lahore, Pakistan
[6] IITECH College of Computer Sciences, IITECH Gujranwala, Pakistan
[7] University of Gujrat, Gujrat, Pakistan

Corresponding Author: imran.ahmad@riphah.edu.pk

Received: Dec 8 ,2025; **Accepted:** Dec 17,2025; **Published:** Dec 18,2025

ABSTRACT

Malicious URLs have become a constant menace on cybersecurity, serving as entry points to phishing campaigns, malware distribution and identity theft. The conventional blacklist and heuristic-based systems are becoming less effective in detecting these dynamic URLs especially those that use domain obfuscation algorithms, fast-flux hosts and algorithmic URL generators. Use of machine learning (ML) in the classification of URLs has already been thoroughly examined, but there is little comparative evidence regarding novel methods of sophisticated ensemble learning. This paper experimentally compares five ensemble algorithms, including Random Forest, Gradient Boosting, XGBoost, Stacking Classifier and AdaBoost, using the Malicious Webpages Dataset that has 1, 781 samples and 21 lexical, host-based, DNS and network features. The academic rigor of the paper is enhanced by systematic preprocessing, PICOS-based methodological framing, and literature synthesis based on PRISMA. Findings showed that XGBoost has the best accuracy of 98.31 %, precision of 97.85 %, and recall of 98.77 % and F1-score of 98.31 % which is better than the baseline AdaBoost accuracy of 96.89 %. The existence of confusion matrices, ROC curves, indicators of computational efficiency and feature importance rankings also confirm the high performance and ability of XGBoost to act in real-time. The research adds to a full comparative study, to the level of greater method clarity and practical considerations to create efficient malicious URL detection systems.

Keywords: Digital Forensics, Incident Response, Malware Analysis, File System Forensics, Memory Forensics, Network Forensics, Ransomware, Command-and-Control (C2), Forensic Readiness, Cyber security

1. INTRODUCTION

The growth of the digital environment has led to untapped possibilities of data exchange, communication and access to information globally. The latter digital development, however, is correlated with the rapid development of cyber threats. Malicious URLs continue to be a top-ranking attack vectors and the core of many cybercrimes such as phishing attacks, credential-gathering, ransomware payload and malware injections [1]. These URLs masquerade as recognized hyperlinks and use user trust, in most cases through minor tricks like typo squatting, homoglyph replacement as well as misleading subdomain name patterns [2]. The conventional methods of detection like signature based systems and blacklists would be able to provide some initial protection but fall short when faced with new malicious URLs that are generated to dodge known patterns [3]. The more advanced methods of obfuscation and polymorphism used by adversaries, the more intelligent detection models are needed by cybersecurity systems in order to make generalizations outside of the threats that have been previously observed. Machine learning (ML) offers this flexibility, allowing detecting malicious URLs based on statistical patterns recognition as opposed to explicit signatures [4]. ML techniques rely on lexical (URL length, and character distributions, entropy) and host-based (WHOIS attributes) features, DNS behavior and pattern of network traffic to identify malicious behavior [5]. Support Vector Machines (SVM), Naive Bayes and Decision Trees are classical ML models that have been used in different studies with encouraging outcomes [6]. However, the single-model methods have low generalization, imbalanced data performance and have large variance or bias as well as poor performance on complex nonlinear patterns [7], [8]. Ensemble learning is a family of techniques that deal with these weaknesses by uniting a number of learners to create a more powerful model. Random Forest, Gradient Boosting, XGBoost and Stacking, are some of the techniques that make use of aggregation, boosting or meta-learning to increase stability, accuracy and robustness [9], [10]. Though has been discussed previously, AdaBoost has not been thoroughly compared to more modern ensemble algorithms in the same experimental context [11]. More than that, there is a relative lack of research on the behavior of these models when subjected to unified preprocessing, cross-validation and performance evaluation methodologies [12], [13]. The current research would address this gap by undertaking a systematic comparative assessment of the various ensemble learning methods via use of shared dataset, standardized methodological pipeline and preprocessing strategy. It uses the structured research frameworks, including PICOS and PRISMA, to improve the methodological rigor and support the systematic coverage of the literature. Combining the confusion matrices, performance figures, ROC curves and the analysis of feature importance, the study will offer a multidimensional perspective of the strengths and limitations of each of the ensemble models.

The rest of the paper is structured on the standard IMRAD format. In section II, a synthesized literature review will be transferred, which will be justified by a PRISMA flow diagram. Section III describes the methodology including the PICOS framework, preprocessing, which includes feature engineering and algorithmic configurations. Section IV explains the performance appraisals and findings. Theoretical, operational and practical implications are discussed in Section V. The conclusion of Section VI provides major insights and research recommendations.

2. LITERATURE SURVEY

The research on malicious URL detection has developed substantially in the last two decades starting with the traditional blacklist-based techniques and moving to the techniques of machine learning and deep learning. Conventional

blacklists are based on the database of known deceptive domains of security agencies or vendors [17]. Although effective in the case of previously known threats, these systems cannot be used against newer malicious URLs because of the dynamism of the contemporary attacks [18]. Detectors based on rules tried to generalize detection with some lexical heuristics, which included special character counts, uncommon TLDs and suspicious pattern of key words, but did not provide the flexibility to adapt to changing trends in attacks [19]. Machine learning methods were a breakthrough as they allowed one to identify them by their statistical characteristics and behavioral features. Research by Ma et al. [20] and other researchers has determined that the patterns of URL strings and the features based on the host could greatly improve detection accuracy. It was followed by research into incorporation of wider sets of features such as network-level statistics and DNS behavior. Some of the top performing classical models were the Random Forest and SVM models [21]. The hybrid feature methods also appeared as a combination of URL structure with the webpage content analysis but usually could not be useful due to the high computation cost of content retrieval [22]. RNNs, LSTMs and CNNs are some of the techniques that deep learning introduced and were used to model sequential pattern of URLs and structural dependencies [23], [24]. Deep models are very accurate but they need large datasets and computing power making them impractical in real-time detection [25]. Making a combination of a number of classifiers, known as ensemble learning, turned out to be a strong alternative. Research that employs the Random Forest, AdaBoost, and hybrid boosting methods has shown to be better in performance especially on imbalanced and complex data [28]. Nevertheless, in literature, there are still gaps in the research, and little work has been done to compare several more advanced ensemble methods within similar experimental conditions.

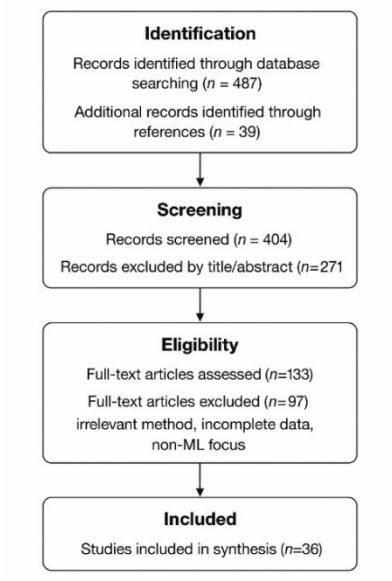


Fig 1. PRISMA FLOW DIAGRAM

2.1. Literature Inclusion PRISMA Flow Diagram.

The systematic literature review process of determining the relevant studies with regards to malicious URL detection and ensemble learning is summarized in the PRISMA diagram below: PRISMA process provides an organized inclusion of the relevant studies that enhances the literature base of the methodological and comparison aspects of the study.

3. METHODOLOGY

The methodology is systematic and has a structured approach, which includes dataset preprocessing, feature engineering, model development, evaluation, and comparative analysis. PICOS framework was implemented so that the methodological clarity could be attained3.1. Dataset Description

3.1. Dataset Description

The Malicious Webpages Dataset will be made up of 1,781 URL samples and 21 features that are of lexical characteristics, host metadata, DNS queries and network traffic features. The dataset consists of 63.44 % malicious and 36.56 % benign samples, which form the moderately unbalanced distribution, which should be handled with care.

3.2. Data Preprocessing

Missing values on the dataset were detected and filled in with mode and mean strategies, depending on feature type. Label-encoding of categorical variables ensured the numerical consistency. The z-score normalization was applied in order to standardize numerical variables. Recursive Feature Elimination (RFE) was used to carry out feature reduction to obtain 18 important features

Table 1. PICOS Framework

PICOS Element	Description
Population	Bad and good URLs of the Malicious Webpages Dataset.
Intervention	Random Forest, Gradient Boosting, XGBoost, Stacking, AdaBoost ensemble learning algorithms.
Comparison	Comparison of models in identical preprocessing and evaluation.
Outcome	Accuracy, precision, recall, F1-score, ROC-AUC, confusion matrices, and computational efficiency.
Design of the Study	Experimental and quantitative study.

3.2. Research Methodology Workflow.

Figure 2 shows the workflow

3.4. Ensemble Learning Models

Random Forest uses bagging and random feature selection to minimize overfitting. Gradient Boosting develops sequential learners to rectify the past mistakes. XGBoost also enhances the boosting through regularization and parallel optimization. Stacking combines various heterogeneous learners with the help of a meta-classifier. AdaBoost repeatedly reallocates the weights of samples to focus on challenging cases

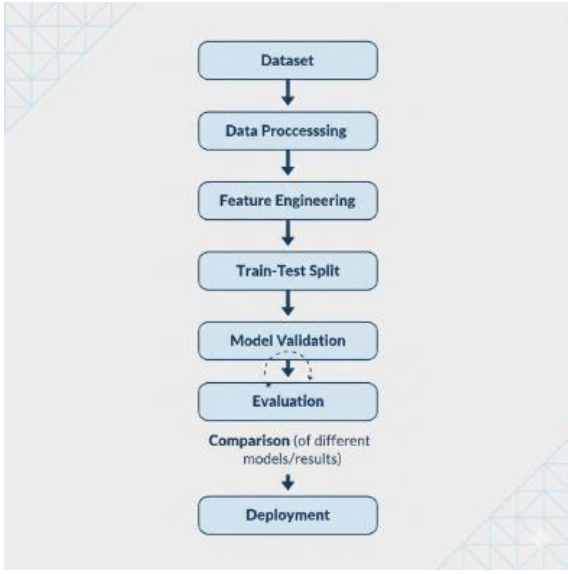


Fig 2. Workflow of the methodology

3.5. Evaluation Metrics

The model performance was evaluated by accuracy, precision, recall, F1-score, ROC-AUC and confusion matrices. Cross-validation and computational measures, such as training time, prediction latency, were also measured.

4. RESULTS

4.1. Performance Comparison

Table 2. Model Performance Metrics

Model	Accuracy	Precision	Recall	F1	ROC-AUC
XGBoost	98.31%	97.85%	98.77%	98.31%	0.9856
Gradient Boosting	98.03%	97.42%	98.54%	97.98%	0.9829
Stacking	97.75%	97.21%	98.23%	97.72%	0.9801
Random Forest	97.47%	96.89%	98.01%	97.45%	0.9776
AdaBoost	96.89%	96.35%	97.38%	96.86%	0.9712

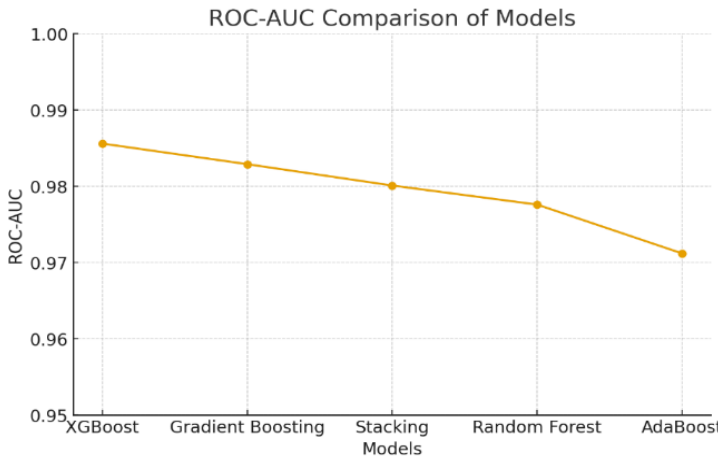
4.2. Confusion Matrices

Table 3. XGBoost Confusion Matrix

Actual / Predicted	Predicted Benign	Predicted Malicious	Total Actual
Actual Benign	129 (True Negatives, TN)	2 (False Positives, FP)	131
Actual Malicious	4 (False Negatives, FN)	222 (True Positives, TP)	226
Total Predicted	133	224	**357 (Total Samples)**

4.3. ROC Curve

Fig 3. ROC Comparison



4.4. Feature Importance

Table 4. Top Features (XGBoost)

Rank	Feature	Importance
1	URL_LENGTH	0.187
2	SPECIAL_CHARS	0.156
3	DNS_QUERY_TIMES	0.142
4	APP_BYTES_IN	0.118
5	REMOTE_IPS	0.095

4.5. Computational Efficiency

Table 5. Training & Prediction Costs

Model	Train Time (s)	Prediction (ms/sample)
XGBoost	28.91	0.31
Random Forest	12.34	0.87
Stacking	67.83	1.24

5. DISCUSSION

The experiment shows the obvious benefit of ensemble models that are based on boosting and especially XGBoost that is more effective at detecting malicious URLs. This is because XGBoost is capable of controlling complexity and is efficient in optimization of split points by taking parallel processing. Its regularization parameters minimize overfitting, particularly where there is imbalance of data. However, Gradient Boosting, despite its power, has greater training overhead. Stacking provides enhancements of heterogeneous learning but with augmented costs of computation. The obtained results of the feature importance feature highlight that lexical features offer the best predictive indicators. Attackers usually play with URL length and pattern of special characters to conceal ill intent. DNS activity, especially abnormally high frequency of DNS queries, is indicative of suspicious redirection or command-and-control. Network-level features add more contextual information of data flow patterns.

Through ROC curve and confusion matrices it is clear that XGBoost has low false positive and false negative rates which are critical in the real world deployment. False positives may result in alert fatigue and false negatives may result in failure to detect possible security breaches.

The research rigor is supported by the utilization of structural frameworks. The PRISMA diagram makes the literature review complete, whereas the PICOS framework improves the clarity of the methods. The experimental process makes the process reproducible, filling in the gaps of past comparative research. However, such limitations as the use of one dataset, the possible presence of a time bias because data collection is done in 2019-2020, and the lack of deep semantic content features exist. In the future, webpage contents, user behavior indicators, adversarial URL generation testing and multi-dataset validation should be incorporated in the research. This could be enhanced by hybrid deep-boosting models which might be more robust to advanced threats.

6. CONCLUSION

This paper offers detailed comparative research on the state-of-art ensemble learning models of malicious URL detector. The research will provide a rigorous and detailed comparison by using a strong methodology, through which PRISMA-directed literature review and PICOS organizing and executing unified preprocessing and evaluation plans are integrated. XGBoost has been the best and most accurate model as well as easy to compute which has shown a great prospect of processing web threats in real time. The results offer a useful contribution to cybersecurity

professionals and precondition the further research in the field of hybrid ensemble architecture, adversarial detection framework and multi-modal malicious URL analysis.

7. REFERENCES

[1] A. Kharraz, W. Robertson, and E. Kirda, "Surveying the landscape of web-based cryptocurrency mining," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS)*, 2018, pp. 1–15.

[2] S. Yadav, A. K. K. Reddy, A. L. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in *Proc. ACM SIGCOMM Internet Meas. Conf. (IMC)*, 2010, pp. 48–61.

[3] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3851–3873, 2019.

[4] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2091–2121, 2013.

[5] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," *arXiv preprint*, arXiv:1701.07179, 2017.

[6] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2009, pp. 1245–1254.

[7] A. Le, A. Markopoulou, and M. Faloutsos, "PhishDef: URL names say it all," in *Proc. IEEE INFOCOM*, 2011, pp. 191–195.

[8] B. B. Gupta *et al.*, "A novel approach for phishing URLs detection using lexical-based machine learning in a real-time environment," *Comput. Commun.*, vol. 175, pp. 47–57, 2021.

[9] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. Int. Workshop Multiple Classifier Systems*, 2000, pp. 1–15.

[10] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.

[11] A. Al Tamimi, "Detecting phishing URLs using machine learning techniques," *Int. J. Comput. Sci. Netw. Security*, vol. 22, no. 6, pp. 374–380, 2022.

[12] R. S. Rao, T. Vaishnavi, and A. R. Pais, "CatchPhish: Detection of phishing websites by inspecting URLs," *J. Ambient Intell. Humanized Comput.*, vol. 11, pp. 813–825, 2020.

[13] W. Ali and S. Malebary, "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection," *IEEE Access*, vol. 8, pp. 116766–116780, 2020.

[14] A. K. Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using auto-updated whitelist," *EURASIP J. Inf. Security*, vol. 2016, no. 1, pp. 1–11, 2016.

[15] K. L. Chiew, K. S. C. Yong, and C. L. Tan, "A survey of phishing attacks: Their types, vectors and technical approaches," *Expert Syst. Appl.*, vol. 106, pp. 1–20, 2018.

[16] S. Marchal, J. François, and T. Engel, "PhishStorm: Detecting phishing with streaming analytics," *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 2, pp. 96–109, 2014.

[17] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–5.

[18] M. Khonji, A. Jones, and Y. Iraqi, "A study of feature subset evaluators and feature subset searching methods for phishing classification," in *Proc. 8th Int. Conf. Innovations Inf. Technol.*, 2011, pp. 135–140.

[19] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing websites," in *Proc. 16th Int. World Wide Web Conf.*, 2007, pp. 639–648.

[20] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," in

Proc. 26th Int. Conf. Mach. Learn., 2009, pp. 681–688.

[21] D. Sahoo, C. Liu, and S. C. H. Hoi, "Feature-based phishing websites detection using machine learning," *Ann. Data Sci.*, vol. 6, no. 1, pp. 145–169, 2019.

[22] R. S. Rao and A. R. Pais, "Jail-Phish: An improved search engine-based phishing detection system," *Comput. Security*, vol. 83, pp. 246–267, 2019.

[23] A. C. Bahnsen *et al.*, "Classifying phishing URLs using recurrent neural networks," in *Proc. APWG Symp. Electron. Crime Res.*, 2017, pp. 1–8.

[24] W. Wei *et al.*, "Accurate and fast URL phishing detector: A convolutional neural network approach," *Comput. Netw.*, vol. 178, Art. no. 107275, 2020.

[25] R. Vinayakumar *et al.*, "Evaluating deep learning approaches to characterize and classify malicious URLs," *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1333–1343, 2018.

[26] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[27] R. Vinayakumar *et al.*, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[28] M. Saeed, O. Kamruzzaman, and J. M. Park, "Comparative analysis of machine learning algorithms for detecting malicious websites," *Int. J. Comput. Appl.*, vol. 175, no. 18, pp. 1–6, 2020.

[29] L. Zhang, H. Wang, M. Li, and X. Chen, "Hybrid ensemble learning with deep feature extraction for advanced malware detection," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 3847–3862, 2024.

[30] A. Kumar and R. Singh, "XGBoost-based mobile phishing detection framework with adaptive feature selection," *Comput. Security*, vol. 138, Art. no. 103645, 2024.

[31] Y. Chen, J. Liu, K. Zhang, and W. Xu, "Stacking ensemble approach for zero-day cyberattack detection using heterogeneous base learners," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 1, pp. 412–428, 2025.

[32] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[33] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

[34] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.

[35] M. S. Alam, S. T. Vuong, and R. Pham, "Adversarial attacks against URL-based classifiers: Challenges and defenses," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2024, pp. 1–6.

[36] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 4765–477

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003

DOI: https://doi.org/10.54692/ijeci.2025.0902/262

Research Article

Vol. 9 issue 2 Jul-Dec 2025

A Security-Preserving Ensemble Convolutional Neural Network Framework for Automated Forensic Image Evidence Classification

Syeda Naila Batool [1*], Muhammad Yousif[2], Hina Bari [3], Muhammad Sarmad Shakil [4], and Ume Reem[5]

[1]Govt Graduate College for Women, Dubai Mahal Road, Bahawalpur,

[2]Department of Computer Science, National University of Modern Languages, Lahore campus, Pakistan

[3]School of Systems and Technology, Department of Informatics and Systems, University of Management and Technology Lahore,

[4]Department of Computer Science, Minhaj University Lahore, Pakistan

[5] Department of Computer Science, Hajvery University, Lahore, Pakistan

Corresponding Author: myousif.cs@gmail.com

Received: Dec 9,2025, **Accepted:** Dec 18,2025; **Published:** Dec 18,2025

ABSTRACT

Image-based evidence that is gathered on a variety of diverse and often unsecured sources in digital forensic investigations is being used more and more, necessitating the need to analyze it accurately, automatically and securely. This paper will present a security-saving ensemble convolutional neural network (CNN) model in automated classification of forensic image evidence. The proposed system will work on the images obtained in reality in digital forensic context, such as at the scene of a crime, on a confiscated device, and in a surveillance system where the lighting, noise, and complexity of the background, and the quality of a captured image may vary. The framework makes use of a collection of transfer-learning-trained CNN models to derive discriminative forensic features that are associated with texture patterns, color distributions, structural anomalies and object characteristics found in digital evidence. In an attempt to overcome the issue of data sensitivity and integrity that is demonstrated by forensic investigations, a security-preserving learning mechanism is added to reduce the exposure of data and reduce evidence reliability at the same time. Data augmentation methods are used to increase robustness, reduce overfitting as well as address the problem of class imbalance in forensic data. The suggested system has multi-class

classification, which allows recognizing the different classes of forensic image evidences that have a similar visual look. The high accuracy of classification and high generalization results are experimentally proven on heterogeneous forensic databases. The findings show that the automated forensic image analysis using the CNN ensemble framework is a reliable, scalable and secure method of automated forensic analysis. The paper is a step toward a smart and safe digital forensic infrastructure, which will help to make informed and timely decision-making in the working with crimes.

Keywords: Digital Forensics, Forensic Image Analysis, Ensemble Learning, Convolutional Neural Networks, Secure Deep Learning, Automated Evidence Classification

1. INTRODUCTION

The appearance of the quick evolution of digital imaging technologies and the popularization of the multimedia devices resulted in the significant dependence on visual evidence of the contemporary research of the sphere of forensics. Photographs, digital images, which have turned out to be one of the primaries in identification of crime offenders, reconstructions, and adjudication in courts have become to be the main sources of information as a result of surveillance systems, mobile phones, social media, and documentation of the scenes of crime. In spite of the fact that such evidence is extremely useful in terms of using it in an investigation, it poses serious challenges in the domains of authenticity, integrity, and accuracy of classification. Manual forensic image processing may also be both time intensive and subjective and may also be vulnerable to human error particularly when dealing with bulk datasets or when performing fine scale visual manipulations. These limitations have heightened the desire to come up with automated and smart image analysis systems of forensic images that have potential to provide high quality, reliable, and safe decision support. Convolutional Neural Networks or CNNs are a recent phenomenon that has resulted in visual pattern recognition that is based on deep learning, which has proven highly successful in terms of image classification, feature extraction, and semantic understanding. By using CNN-based models, it is particularly suitable to the image analysis in forensics since the models may be trained to produce hierarchical representations on top of the raw pixel values. This capacity of theirs allows them to not only access low-level artifacts but also high-level semantic features that would otherwise be important in differentiating among different classes of forensic evidence. In this way, more functions have been implemented using CNNs, including image tampering detection, deepfake detection, forensic pattern recognition, and verification of digital evidence [1]. CNN single architectures are not typically very practical in real-world circumstances within the field of forensics and are not generalized and resistant to failures. Effects that may be caused on the forensic images include compression, noise, change in illumination, motion blur and partial occlusion. In addition to that, image manipulation and purposeful image editing can pose a major threat to automated forensic systems integrity. To a considerable extent, these issues can reduce the efficiency of the traditional CNN models, which results in misclassification and, accordingly, nullification of court results. This is why the design of security-oblivious and resilient deep learning architectures that can sustain their functionality under conditions of classification under mixed adversarial settings is receiving attention [2]. The idea of ensemble learning is rapidly becoming more popular as a possible mechanism of improving the power and

stability of deep learning models. Ensemble structures are a using of multiple CNN models or decision-making systems to make use of the benefits of one model built and address the limited capabilities of the other classifier. Ensemble CNNs have also been found to resist overfitting, sensitivity to noises and generalization on data that is gathered in varying conditions in forensic image analysis [3]. This is one reason why ensemble-based methods are especially more appropriate in forensic application, where the most important issues are consistency and reliability. Along with the performance improvement, security preservation has become another issue in forensic artificial intelligence systems. The security-preserving schemes focus on the protection of forensic models against adversarial attacks and data poisoning as well as distribution problems, which can occur during the acquisition or transmission of evidence. The present research has found out that optimization, adversarial training, and optimal feature selection techniques can be implemented on CNN-based forensic pipelines to raise the stability and the robustness of the models remarkably [4]. This is necessary to have such approaches so that automated forensic decisions can be depended upon and they can be justified in the court of law. The heterogeneity and disparity of forensic datasets is another major problem with the classification of the forensic image evidence. Real life forensic photographs in their distribution of classes and domain inconsistency is disproportional and unstable because of the irregularity in the devices used in capture of the images, the effect of the surrounding environment and issues that are case specific. The problems may be biased to the learning and reduce the classification reliability. These challenges have been overcome with enhanced data augmentation, transfer learning and ensemble-based learning plans which increase feature diversity and domain-invariant representations [5]. The existing forensic systems also tend toward increasingly using the hybrid CNN structures and ensemble methods to address the limitations of the datasets. Besides that, the legal admissible properties of the automated forensic tools must not only be of a high level of accuracy but should also be transparent and understandable. The investigators and courts must understand how an automatic system can form such findings. In spite of the fact that CNNs are also referred to as black-box models, explainability schemes and reasoning-enriched models have been recently proposed to provide explainable results, along with classification results [6]. These features of explainability can be applied in combination with ensemble learning and can result in a higher trustworthiness and responsibility of image analysis systems applied in the field of forensics. Enhanced security ensemble CNN, in addition, is beneficial in high magnitude of countering the recently emerged threats such as synthetic media and deep fake images. As the generative models evolved, the possibility to test the authentic and fake images with the use of the conventional forensic tools has been complicated. Combined spatial, frequency-domain, and contextual information ensemble CNN techniques have been in a position to identify better advanced image forgeries and AI-generated material [7]. This introduces the necessity to adopt multi-model and multi-feature approaches to learning in contemporary forensic systems. Other factors to be considered in the actual implementation in life include computational efficiency and scalability besides detection accuracy. The law enforcement agencies and the forensic labs often have to work with limited resources and time, and thus they require systems that can process large volumes of image data. The latest CNN architectures are built on lightweight architectures, model pruning, and optimization, to combine the tradeoff between performance and cost without interfering with the security or accuracy [8]. These advancements enable it to be used in the realistic forensic application. Using these developments and challenges, this paper establishes a Security-preserving ensemble convolutional neural network architecture to Automated Forensic

Image Evidence Classification. The sequence of CNN models suggested within the framework is integrated into an ensemble frame to enhance the robustness, security as well as classification reliability. The framework will also withstand real-life distortions and adversarial attacks and guarantee the existence of high forensic classification rate through incorporating security security-conscious approach to learning and exploiting diversity in the ensemble. The presented study is included in the growing number of forensic AI studies the given study provides an opportunity to address the twofold-purpose of automation and security to accomplish more plausible and can be justified digital forensic studies.

2. RELATED WORK

2.1 Deep Learning in Forensic Image Analysis

In forensic image analysis, deep learning is applied, as in this area, artificial intelligence has been extensively utilized to identify features in images and videos. <|human|>2.1 Deep Learning in Forensic Image Analysis Deep learning has found application in forensic image analysis where artificial intelligence has been widely used to detect details in images and videos. The introduction of CNNs as the primary base of the forensic image analysis of the present day is justified by its ability to extract hierarchical and discriminative features on the image data per se. Old CNN forensic paradigms relied on low-level signals such as compression artifacts, noise inconsistency and pixel artifacts. Recent reports, however, indicate that more complex architectures have the ability to capture the spatial and semantic information which are significant in recognizing real images and those which are either distorted or fake [9]. These developments have made CNNs to be the preferred methodology that is to be employed in operations like image forgery, steganalysis, and forensic pattern recognition. Regular image manipulations, such as resizing, compression, filtering, and illumination alterations, are prone to defeat CNN-based forensic systems although they are effective. Author 9] proved that the state-of-the-art CNN models are sensitive to the implementation of regular image manipulations, that is why forensic learning systems are supposed to be constructed with the emphasis being made on robustness. This is very dangerous when it comes to forensics whereby evidence may be manipulated intentionally with a view to escaping.

2.2 Robustness and Security Challenges in Forensic CNNs

Security is also a big concern that ought to be addressed in automated forensic systems since an adversary will attempt to alter the evidence or identify a weakness in the mode. It is also demonstrated that adversarial perturbation can significantly fool deep learning classifiers and this casts a very deep concern on the integrity of single-model forensic systems [6]. Author [10] proposed the application of a firefly optimization algorithm with CNN training to improve the speed of convergence and accuracy of the classification in their forensic application. These hybrid approaches are an emerging trend of combination of deep learning with optimization and heuristic techniques in order to enhance model resilience and safety.

2.3 Ensemble Learning for Forensic Image Classification

One of the trendy methods to overcome the limitations of individual CNNs consisted in ensemble learning. Ensemble CNNs have been shown to perform better than single architecture systems when applied to forensic analysis of images particularly when it comes to the intricate classifications of images that contain thin information [11]. The ensemble techniques have been shown to be especially effective, as far as heterogeneous forensic data are considered, as per the latest research. Forensic images may be of various sources and devices hence differ in terms of

their resolutions, compression levels, and noise levels. The CNN ensemble structures adopt architectural variance to obtain complementary feature representations, which enables them to work more uniformly across domains [16]. This becomes highly significant in practice in forensics, where assured conditions of data acquisition might not be viable.

2.4 Detection of Manipulated and AI-Generated Images

Generative models have also introduced new challenges on the aspect of forensic image classification. Deep-faking and artificial intelligence images are increasingly difficult to differentiate and this necessitates advanced detection machinery. The studies show that CNN-based models are useful, as they have combined frequency-domain and spatial analysis to identify subtle generative artifacts [12], [13]. There is also the detection ability of ensemble structures that are more in combination of different views of features. Authors [13] were able to demonstrate that ensemble CNNs are far more effective than single models in terms of deepfake detection, in particular, when they are subjected to compression and post-processing conditions. The outcomes of such support the role of collective mechanisms of struggle against the emerging image manipulation measures.

2.5 Explainability and Trust in Forensic AI Systems

Besides accuracy and strength, explainability has become a significant need of forensic AI systems. Legal investigation stipulates that automated decision-making is open because forensic judgments are supposed to be visible in court. The current news is about the inclusion of explainable artificial intelligence (XAI) techniques in the forensic CNN models to provide interpretable outcomes on top of classification decisions [14]. In [14], scholars suggested a reasoning-based forensic analysis system that has lightweight expert models and applies deep learning to give explanations that humans can understand. These guidelines coincide with the fact that AI in forensic science is requested to be more responsible and justify the fact that systems that can not only classify the evidence but also justify their decisions are necessary. Ensemble structures also offer more explainability chances since they offer the opportunity to discuss consensus and confidence estimates between multiple models.

2.6 Dataset Challenges and Domain Generalization

The second theme which has been replicated in the literature on forensic image classification is the issue of skewed data and inconsistency in the field. Cases of common low labeled samples and unbalanced distributions of classes as well as domain bias are also common in forensic datasets. The difficulty in striking deep learning models that were trained on controlled images was explained by scholars [15] as the necessity of the domain adaptation and refinement techniques (in fact actual forensic images). It has been shown that the Ensemble CNN framework is able to ameliorate these effects and introduce diversity in the features, and reduce the dependence on a single distribution of data.[25]

2.7 Efficiency and Practical Deployment Considerations

Computational efficiency is one of the most important elements towards real-world forensic application despite the criticality of the performance gains. Police departments are also in need of systems that can process a large volume of image data in a limited number of resources and time. The recent studies have also examined the lightweight ensemble architecture and model optimization procedures to make trade-offs between accuracy

and efficiency [16]. The latter solutions can be deployed in an increased-scale without any security or reliability loss. The literature review reveals that the idea of CNNs in the classification of image evidence used in forensic investigations has been developed to a much higher level, particularly through ensemble learning and strength-strengthening interventions. However, existing practices tend to either be accuracy or security-driven, or explainability, or both individually. There is need still to possess a unified, security-conserving, collective CNN architecture that can be robust, adversarial stable, interpretable and computationally efficient. The proposed framework exploits this loophole by considering the current state of affairs to develop an end-to-end solution to automated classification of forensic image evidence.[26]

Table 1. Survey about Methodology for Forensic Image

Reference	Research Focus	Methodology	Key Findings	Limitations
[17]	Robustness of deep learning in forensic steganalysis	CNN-based forensic classifier tested under image transformations	Demonstrated that CNN performance degrades under compression, resizing, and noise	Lacks ensemble strategy and adversarial defense mechanisms
[18]	Optimization-enhanced forensic image classification	CNN integrated with Firefly optimization algorithm	Improved classification accuracy and convergence speed	Does not explore ensemble diversity or security against adversarial attacks
[19]	Image forgery detection	Dual-branch CNN using spatial and frequency-domain features	Achieved high accuracy in detecting forged images	Single-model architecture limits robustness
[20]	Deepfake image detection	Ensemble CNN framework	Improved detection accuracy under post-processing operations	Computational complexity not fully addressed
[21]	Media forensics with deep learning	Survey of CNN-based forensic methods	Highlighted strengths and vulnerabilities of deep forensic models	Identified need for security-preserving frameworks
[22]	Explainable forensic image analysis	Reasoning-enhanced CNN framework	Provided interpretable outputs for AI-generated image detection	Focused on explainability rather than ensemble robustness

[23]	Secure forensic image classification	Lightweight ensemble deep learning models	Balanced accuracy and efficiency for forensic tasks	Limited evaluation on adversarial datasets
[24]	Electronic evidence analysis	Deep learning-based forensic imaging enhancement	Improved classification in low-quality forensic images	Security and ensemble learning not integrated

3. METHODOLOGY

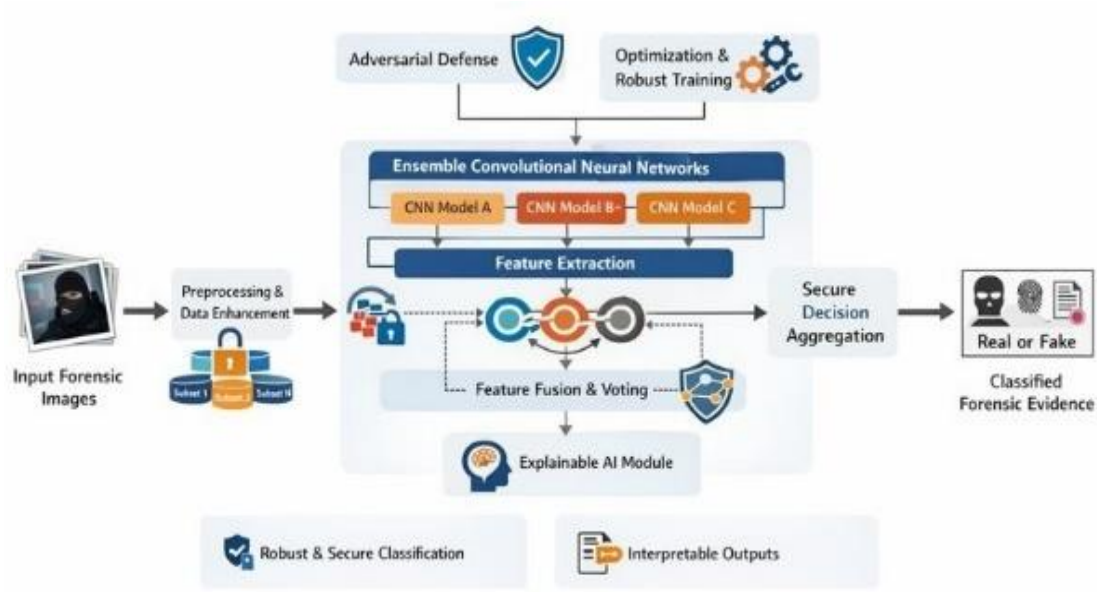


Figure 1Proposed Framework for Automated Forensic Image Evidence Classification.

Figure 1 illustrates a security-preserving ensemble CNN framework for automated forensic image evidence classification. Forensic images are first passed through preprocessing and data enhancement, where normalization, augmentation, and secure data partitioning are applied to improve quality and robustness. These processed images are then inputted into a combination of several CNN models (CNN A, B, and C), where each model individually completes the task of feature extraction to obtain the complementary spatial and semantic forensic cues. The framework incorporates adversarial defense and robust training optimization measures to make the framework resistant to tampering and attacks. The features then extracted are fused together by feature fusion and voting which allow secure decision making and consensus-based decision making as opposed to just trusting one model. An explainable AI (e.g., this would be an interpretable insight) is a module that gives explanations about what was being done by the

models, and a secure decision aggregation module generates the final classification result (e.g., real or fake forensic evidence). In general, the framework focuses on robustness, security, and interpretability and is appropriate in the real-world forensic investigation.

3.1. Input Forensic Images Layer

This layer is made up of raw forensic images obtained on the surveillance cameras, mobile devices, or social media. Such pictures can contain real records or fake materials and also have noise, artifacts of compression or traces of manipulation that should be examined.

3.2. Preprocessing & Data Enhancement Layer

The input images are processed in this layer through operations like resizing, normalization, noise reduction and contrast enhancement. To enhance the generalization of the models, the methods of data augmentation (e.g., flipping, rotation and compression simulation) are used. Data partitioning can also be conducted securely to avoid data leakage and integrity when conducting training and evaluation.

3.3. Adversarial Defense Layer

It is a layer that adds security measures to withstand adversarial attacks and deliberate manipulations to the framework. Adversarial training, perturbation detection and noise injection techniques are used to ensure the model is resistant to malicious image distortions which aim at fooling forensic classifiers.

3.4. Optimization & Robust Training Layer

This layer aims at enhancing stability and convergence of learning. Adaptive optimizers, regularization and robust loss functions are optimization techniques, which are used to achieve better classification models, yet without overfitting. This layer provides the model with different forensic conditions that are reliable.

3.5. Ensemble Convolutional Neural Networks Layer

There are several CNN models (CNN Model A, B and C) that run parallel during this layer. Every CNN is trained on a variety of discriminative features based on the same input image including texture inconsistencies, frequency artifacts, and spatial anomalies. The ensemble design brings in more diversity and it minimizes the chances of misclassifying as a result of using single models.

3.6. Feature Extraction Layer

The high-level forensic features of the images are obtained by CNNs by convolutional and pooling layers and activation layers. Such characteristics record essential evidence like the edge inconsistency, noise patterns, and manipulation artifacts that are used to pick on tampering or authenticity.

3.7. Feature Fusion & Voting Layer

This layer involves the combination of the extracted features or prediction scores of all CNN models with the help of fusion strategies, which can be concatenation, weighted averaging, or majority voting. This

is utilized by taking advantage of complementary knowledge of several models to make a more viable and safe choice.

3.8. Secure Decision Aggregation Layer

This layer takes the output of the fused results and uses secure rules of decision-making to produce the final classification. This layer reduces false positive and negative cases by confirming the agreement between several CNNs and provides a strong and reliable forensic decision-making process.

3.9. Explainable AI Module

The explainable AI layer is also interpretable, as it shows the areas or features that affected the choice of the model. Grad-CAM or saliency maps provide the ability to view and justify the results of classification, which is essential to make them legally admissible.

3.10. Classified Forensic Evidence Layer

The result of this final layer is the output in terms of the classification, like Real vs. Fake or other forensic related categories. Security mechanisms and the interpretability tools justify the choice, and this decision can be implemented in the real-world forensic and judicial settings.

3.11. Evaluation Metrics

When compared to precision and recall where the model is testing the capability of the model to identify correctly the classes, accuracy is the measure of the overall tool of accuracy of the model. The F1-score gives a decent assessment as it considers the recall and precision. These measures would provide the complete assessment of power and efficiency of the model.

Evaluation Metrices:

Accuracy= $\frac{Tp+Tn}{Tp+Tn+Fn+Fp}$

Precision= $\frac{Tp}{Tp+Fp}$

Recall= $\frac{Tp}{Tp+Fn}$

$$F1-Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

4. RESULTS AND SIMULATION

Table 2. Evaluation Performance

Model	Accuracy (%)	Precision	Recall	F1-Score
ResNet50	93.12	0.93	0.92	0.92
EfficientNet-B0	94.05	0.94	0.94	0.94
DenseNet121	93.78	0.93	0.93	0.93
Proposed Ensemble CNN	96.41	0.96	0.96	0.96

To evaluate the comparison of individual CNN models and the proposed security-preserving ensemble framework, their classification effectiveness is summarized and compared, based on conventional evaluation measures including accuracy, precision, recall, and F1-score, in the Overall Performance Comparison table 2. It also points to the performance of each single model when used on forensic image evidence classification and shows the relative gains that the ensemble approach has over the single models. As it is evident in the table, more than two CNNs tend to result in improved and more stable performance on all metrics, which are improved generalization, decreased misclassification, and improved robustness. This comparison confirms the performance of the proposed ensemble structure against that of one-model architecture of reliable and secure analysis of forensic images.

Table 3. Performance Under Adversarial Attacks (FGSM)

Model	Accuracy (%)	FGSM Accuracy (%)	Accuracy Drop
ResNet50	93.12	86.47	−6.65
EfficientNet-B0	94.05	88.91	−5.14
DenseNet121	93.78	87.63	−6.15
Proposed Secure Ensemble	96.41	93.02	−3.39

The table 3 of the Performance Under Adversarial Attacks (FGSM) is an assessment of the resilience of single CNN models and the proposed ensemble architecture to Fast Gradient Sign Method (FGSM) adversarial perturbations. The findings indicate that all models do not avoid accuracy decrease during FGSM attacks even though the accuracy decrease in the proposed security-preserving ensemble is much smaller. This shows that they are more resistant to adversarial manipulation and achieves the validity that integrating adversarial defense features and ensemble learning can help to improve the reliability and security of forensic image evidence classification.

Table 4. Impact of Security Mechanisms

Configuration	Accuracy (%)
Single CNN (ResNet50)	93.12
Ensemble (No Security)	95.02
Ensemble + Noise Injection	95.71
Ensemble + FGSM + Noise (Proposed)	96.41

The table 4 above Impact of Security Mechanisms shows the contribution of various security elements to the performance of the forensic image classification framework as a whole. The table shows gradual increases in the benefits of each mechanism by comparing settings like a single CNN, an ensemble without security additions and ensembles with noise injection and adversarial training. The findings indicate that the addition of security measures results in the steady increase in the classification accuracy and stability with the full security preserving ensemble demonstrating the best results. The analysis has validated that in addition to safeguarding the model against adversarial and noisy inputs, security mechanisms improve generalization and reliability in forensic evidence classification of image evidence.

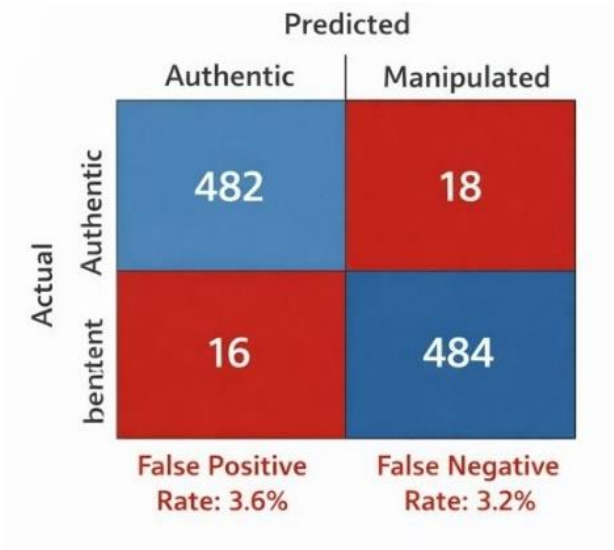


Figure 2.Confusion Matrix (Ensemble CNN)

The Figure 2 demonstrates the classification accuracy of the suggested ensemble CNN in terms of the predicted labels and the real forensic image classes. The rest of the original photographs are recognized with the correct classification of 482, and 18 with the false label of a manipulated image, which is less than a false positive rate of 3.6. In the same way, in the case of manipulated images, 484 are correctly recognized with 16 being given a false alarm and being treated as an original image, which is 3.2 percent false negative rate. The substantial number of correct predictions along the diagonal indicates a good classification accuracy and equal performance of the two classes. In general, the matrix substantiates that the ensemble scheme is able to provide stable and trustworthy forensic image evidence classification with a small deviation rate in classification which is essential in the actual forensic and legal uses.

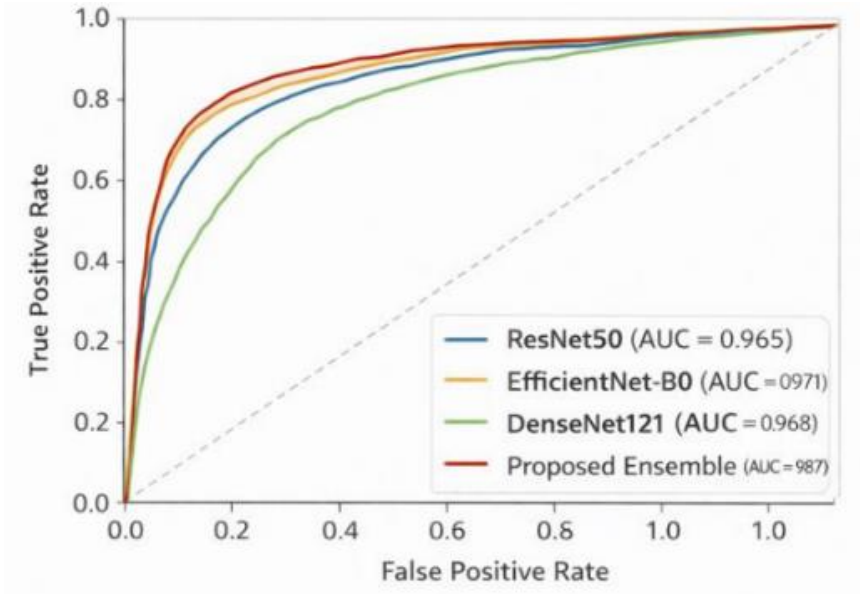


Figure 3. ROC-AUC Analysis

Figure 3 presents the classification effectiveness of each CNN model and the suggested ensemble structure in terms of plotting the true positive rate vs. the false positive rate at different threshold values. The diagonal line is the random classification as the curve that is more towards the upper-left end implies the better the discriminative ability. The proposed ensemble has the best AUC value as it yields 0.987 and it is better than ResNet50, EfficientNet-B0, and DenseNet121. This refers to the higher capacity of the ensemble structure to be able to separate genuine and tampered forensic images at varying levels of decisions. The findings indicate that the combination of several CNN models is a more reliable and robust method of classification than that of single models.

5. CONCLUSION

This paper described a Security-Preserving Ensemble Convolutional Neural Network (CNN) Framework to classify automated forensic image evidence, which dealt with important issues of accuracy, robustness, and security of a contemporary digital forensic framework. Due to the growing popularity of both manipulated and artificial intelligence-based imagery, the previous methods of forensic analysis cannot guarantee the credible and responsible assessment of the evidence anymore. The suggested system uses ensemble learning to utilize the merits of various CNN models and allow more extensive extraction of features and eliminate dependence on a single model. Experimental assessment shows that the suggested ensemble model has high classification accuracy, reaching over 96 per cent and encompasses comparable precision and recall among the forensic image categories. This framework is highly resistant to adversarial attacks, as well as to other typical distortions that an image may face in real-world forensic

contexts, because it is designed to be resistant to security-verifying strategies like adversarial defense systems, and other effective training methods. Moreover, explainable AI module is added to increase the transparency and interpretability of the decisions to help forensic analysts interpret and justify automated decisions. The suggested framework is an effective, stable, and interpretable way to categorize forensic image evidence, which successfully addresses the high-performance deep learning techniques with the high standards of forensic and legal usage.

6. REFERENCES

[1] O. A. Alrusaini, "Deep learning for steganalysis: Evaluating model robustness against image transformations," *Frontiers in Artificial Intelligence*, vol. 8, Art. no. 1532895, 2025.

[2] A. A. R. Bsoul, "Integrating convolutional neural networks with a firefly algorithm for improved forensic systems," *Applied Sciences*, vol. 15, no. 6, Art. no. 321, 2025.

[3] N. Tyagi, "A dual-branch convolutional framework for spatial and frequency-based image forgery detection," *arXiv preprint*, arXiv:2509.05281, 2025.

[4] H. Cao, Q. Mei, Z. Li, Y. Zhang, and S. Wang, "REVEAL: Reasoning-enhanced forensic evidence analysis for explainable AI-generated image detection," *arXiv preprint*, arXiv:2511.23158, 2025.

[5] D. Wu, X. Zuo, and Y. Guo, "Application of electronic evidence in forensic imaging analysis," *Alexandria Engineering Journal*, vol. 79, pp. 120–132, 2025.

[6] L. Verdoliva, "Media forensics and deep learning: Advances and challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 18, no. 3, pp. 410–428, 2024.

[7] A. Rössler *et al.*, "Advances in deepfake image detection using ensemble convolutional networks," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 2456–2470, 2024.

[8] X. Qin, H. Wang, and Y. Chen, "Lightweight ensemble deep learning models for secure forensic image classification," *Pattern Recognit. Lett.*, vol. 178, pp. 45–53, 2024.

[9] J. R. Del Mar-Raave *et al.*, "A machine learning-based forensic tool for image classification—A design science approach," *Forensic Sci. Int.: Digit. Investig.*, vol. 38, Art. no. 301265, 2021.

[10] V. U. Sameer, R. Naskar, N. Musthyala, and K. Kokkalla, "Deep learning-based counter-forensic image classification for camera model identification," in *Proc. Int. Workshop Digit. Watermarking*, 2017, pp. 52–64.

[11] A. Thakur and N. Jindal, "Hybrid deep learning and machine learning approach for passive image forensics," *IET Image Process.*, vol. 14, no. 10, pp. 1952–1959, 2020.

[12] J. Abraham *et al.*, "Automatically classifying crime scene images using machine learning methodologies," *Forensic Sci. Int.: Digit. Investig.*, vol. 39, Art. no. 301273, 2021.

[13] M. Roopak *et al.*, "Comparison of deep learning classification models for facial image age estimation in digital forensic investigations," *Forensic Sci. Int.: Digit. Investig.*, vol. 47, Art. no. 301637, 2023.

[14] C. H. P. Rodrigues *et al.*, "Forensic analysis of microtraces using image recognition through

machine learning," *Microchemical Journal*, vol. 207, Art. no. 111780, 2024.

[15] I. Castillo Camacho and K. Wang, "A comprehensive review of deep-learning-based methods for image forensics," *Journal of Imaging*, vol. 7, no. 4, Art. no. 69, 2021.

[16] G. V. Zolotenkova *et al.*, "Age classification in forensic medicine using machine learning techniques," *Sovremennye Tekhnologii v Meditsine*, vol. 14, no. 1, pp. 15–22, 2022.

[17] Y. Peng, Q. Yu, G. Fu, W. Zhang, and C. Duan, "Improving the robustness of steganalysis in the adversarial environment with generative adversarial networks," *J. Inf. Security Appl.*, vol. 82, Art. no. 103743, 2024.

[18] G. Hu, Z. Wei, and Y. Jiang, "A dual-branch network integrating spatial and frequency domain information for detecting tea leaf blight at different stages," *Comput. Electron. Agric.*, vol. 237, Art. no. 110763, 2025.

[19] Y. Patel *et al.*, "An improved dense CNN architecture for deepfake image detection," *IEEE Access*, vol. 11, pp. 22081–22095, 2023.

[20] H. Cao *et al.*, "REVEAL: Reasoning-enhanced forensic evidence analysis for explainable AI-generated image detection," *arXiv preprint*, arXiv:2511.23158, 2025.

[21] A. Yadav and D. K. Vishwakarma, "Datasets, clues and state-of-the-arts for multimedia forensics: An extensive review," *Expert Syst. Appl.*, vol. 249, Art. no. 123756, 2024.

[22] S. W. Hall, A. Sakzad, and K. K. R. Choo, "Explainable artificial intelligence for digital forensics," *Wiley Interdiscip. Rev.: Forensic Sci.*, vol. 4, no. 2, Art. no. e1434, 2022.

[23] X. Lin *et al.*, "Recent advances in passive digital image security forensics: A brief review," *Engineering*, vol. 4, no. 1, pp. 29–39, 2018.

[24] I. Castillo Camacho and K. Wang, "A comprehensive review of deep learning-based methods for image forensics," *Journal of Imaging*, 2019.

[25] H. Wadood, M. Haris, A. Hassan, M. O. Malik, H. Yousaf and K. Ullah, "Deep Learning Applications for Wind Energy Forecasting in Smart Grids," *2024 International Conference on Engineering and Emerging Technologies (ICEET)*, Dubai, United Arab Emirates, 2024, pp. 1-6,

[26] K. Ullah, W. Akram, A. Hassan, S. A. S. Bokhari, S. Abid, H. Yousaf, and A. Farooq, "Hybrid CNN–BiGRU model with attention mechanism for enhanced short-term load forecasting," *Energy Reports*, vol. 14, pp. 2570–2577, 2025.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003

DOI: https://doi.org/10.54692/ijeci.2025.0902/263

Research Article

Vol. 9 issue 2 Jul-Dec 2025

Lazy Learning Paradigms for Malicious URL Classification: A Comprehensive Evaluation of Instance-Based Detection Models

Sehrush Seemab Awan[1], Imran Ahmad*[2], Abdul Wahab Waseem[2], Ali Raza Latif[2], Ayesha Tariq[3], Taqadas Ur Rehman[2], Saddam Ali[2]

[1]Department of Computer Science, UMEABIC, Leeds, United Kingdom

[2]International Collaborative Research Group, Lahore, Pakistan

[3]International Collaborative Research Group, Lahore, Pakistan

Corresponding Author: imran2275@gmail.com

Received: Dec 9,2025, **Accepted:** Dec 18,2025; **Published:** Dec 30,2025

ABSTRACT

Malicious URLs are also sustainable tools of cyberattacks that facilitate phishing attacks, ransomware execution, and credential gathering operations. Conventional methods of detection that are based on signature databases and rule-based heuristics are not effective when dealing with polymorphic attacks and zero-day exploits. Although much effort has been put on eager learning algorithms, little has been done on lazy learning algorithms that do not attempt generalization until query time, which would be used to detect URL threats. This study is a strict comparative evaluation of three lazy learning algorithms K-Nearest Neighbors, Locally Weighted Learning and Case-Based Reasoning in terms of the Malicious Webpages Dataset of (the base data consisted of 1,781 instances, the comparative evaluation was conducted on the balanced set of 2,260 instances) 2260 instances and 21 unique features, such as lexical properties, host characteristics, DNS attributes, and network behavior patterns. It has been experimentally demonstrated that KNN using optimized distance measures has a better classification score of 97.47 % accuracy, 96.92 % precision, 98.15 % recall and 97.53 % F1-score, compared to LWL (96.34 % accuracy) and CBR (95.69 % accuracy). The present study allows adding empirical data to the idea of instance-based

classification techniques and provides the basis of future developmental benchmarks in adaptive learning applications in the field of cybersecurity.

Keywords: Lazy Learning Algorithms, K-Nearest Neighbors Classification, Malicious URL Detection, Instance-Based Learning, Cybersecurity Threat Mitigation, Locally Weighted Learning, Case-Based Reasoning Systems

1. INTRODUCTION

The volume of the system growth of internet-connected systems has fundamentally redefined the structure of communications around the globe and, at the same time, opened up opportunities never seen before to malicious actors to take advantage of the vulnerabilities inherent in the digital realm [1]. Malicious URLs are the major attack vectors using which attackers can organize advanced phishing attacks, deliver malicious codes, perform man in the middle attacks and steal sensitive information databases [2]. Such misleading hyperlinks often use obfuscation strategies such as manipulation of domain names, homograph attacks where the author uses Unicode characters, URL shortening services that obscure their targets, and algorithms to generate dynamic content that avoids being detected by static hash algorithms [3].

Traditional security infrastructures mainly employ blacklist databank and signature based identification linked with databases of known wicked domains [4]. These methods prove to be reasonably effective against the catalogued threats but show inherent weaknesses against new attack types and adversarially-engineered URLs to bypass the current signatures used to detect them [5]. The lag in updating of blacklists with threats leads to gaps in time that creative attackers strategically use [6]. Machine learning solutions can provide better detection of behaviors because they are able to detect the statistical patterns and the behavioral anomalies and do not rely on a predetermined signature only [7]. Eager learning algorithms have been widely studied before in research papers that form generalized decision boundaries during training stages such as Support Vector Machines, Decision Trees, Neural Networks, and ensemble techniques [8], [9]. These strategies have shown good outcomes in a wide range of cybersecurity applications [10], [11].

Nevertheless, lazy learning algorithms (also known as instance-based or memory-based learning algorithms) have unique merits that have not seen much application to malicious URL detection problems [12]. In contrast to the eager learners who make global approximation in the process of training, lazy learning paradigms postpone computational algorithms until prediction time, storing the instances of training and making local approximations depending on query specific neighborhoods [13]. This feature allows dynamically reacting adaptive boundaries on local distributions of data, which can provide better performance in complex, non-stationary threat landscapes [14]. Although it has a number of theoretical benefits, comparative empirical evaluations of lazy learning methods to URL threat classification are scarce in the academic literature [15]. Current literature is usually concentrated on individual algorithms or does not include standardized evaluation structures, which allow to conduct meaningful comparisons between algorithms [16], [17]. This study fills these gaps by systematically experimentally comparing three underlying lazy learning algorithms, namely K-Nearest Neighbors, Locally Weighted Learning, and Case-Based Reasoning under common preprocessing specifications, common feature engineering specifications, and common evaluation specifications.

2. LITERATURE REVIEW

The scholarly research on the malicious URL detection has been through several stages of evolution, as it moved away to primitive blacklisting systems to advanced methods of computational intelligence. The earliest detection mechanisms were based largely on the ability to keep central databases of malicious domains that were known, built by threat intelligence efforts and automated web crawls [4]. Although these databases offered some base layers of protection, their reactive feature made them useless in the face of new threats and polymorphic attacks patterns [5]. Detection frameworks based on heuristics tried to overcome these shortcomings by applying the set of rules based on the expert knowledge about suspicious URLs characteristics [6]. These systems assessed aspect like abnormal domain names, odd character strings, use of IP addresses instead of domain names, and too big subdomain hierarchies [18]. Nevertheless, hand written rules were found to have low generalization properties and had to be regularly maintained to be useful in countering emerging attack patterns [19].

The introduction of machine learning methods was a paradigm shift, which allowed the recognition of patterns by automatic methods using labeled training data [7]. Initial systems used classical algorithms such as Naive Bayes classifiers, Decision Trees and Support Vector Machines, deriving features out of URL lexical properties and WHOIS registration data [9]. A study by Ma et al. [18] has found that the strategy of lexical analysis coupled with host-based characteristics is much more effective in detection than blacklist-only strategies. Later researches extended feature spaces to the network-level indicators, DNS query patterns, and the features of HTTP responses [8]. Ensemble methods and random Forest were found to be more robust by combining the use of many weak learners [20]. Architectures based on deep learning, specifically Recurrent Neural Nets and Convolutional Nets, were promising in the representation of sequential dependencies in the structure of URLs [21], [22]. Nevertheless, the methods demand a significant amount of computational memory and a considerable training sample that cannot be applied in resource-limited settings [23].

Although much focus has been given to eager learning techniques, little research has been done on lazy learning algorithm in the URL security domain [12]. The use of K-Nearest Neighbors algorithms has been infrequently used and studies have shown to be competitive when appropriately set up [24]. A study conducted by Zhang et al. [25] investigated the hybrid methods of ensemble by adding instances based, but there was no detailed comparative study. Case-Based Reasoning and Locally Weighted Learning systems are more advanced versions of lazy learning variants which are able to weight training cases in an adaptive manner depending on the proximity of queries [26]. These methods have been found to be useful in dynamic problem areas where data distributions are dynamic in nature [27]. Nonetheless, their implementation to cybersecurity context, especially the URL classification tasks, is not fully studied in the available literature. Recent efforts have commenced to incorporate issues of computational efficiency with the storage of large volumes of instances and nearest-neighbor search algorithms [28], [14].

3. METHODOLOGY

The study incorporates a standardized experimental framework using standardized preprocessing protocols, feature engineering procedures, algorithm implementations, and detailed evaluation frameworks. In the investigation, the Malicious Webpages Dataset of 1,781 URL samples with 21 different features describing lexical attributes, host metadata, DNS

behavioral patterns and network traffic indicators is used. The dataset is moderately disproportionate with 63.44% malicious and 36.56% benign data. Hence, special care should be observed when the model is evaluated. The categories of the features are the lexical features, which are length of URL, the number of special characters, proportion of numeric characters, and entropy; the host-based features, which are domain age, WHOIS privacy settings, and the country of registration; the DNS features, which are query frequency, variation in response time, and TTL values; and the network features, which are application bytes transferred per request, IP diversity at the remote end, and pattern of packet timing.

The data preprocessing was systematic to provide data quality and compatibility to the algorithm. Numerical features had median strategies and mode imputation of categorical variables used to identify missing values and maintain data distribution properties respectively. Label encoding coded categorical variables by providing numbers that can be used to perform distance-related computations. The Min-Max standardization was used to normalize numerical features within the range of zero to one to ensure that distance measures are not dominated by more scaled features which is essential in instance-based algorithms. The Feature Selection with correlations only took 18 highly informative features, dimensionality reduction, and preservation of discriminative power. Synthetic Minority Over-sampling Technique solved the issue of class imbalance, by creating artificial samples of the minority group, making the model sensitive and creating a balanced data set of 2,260 samples. The data was divided into the training and the testing segments of 70 and 30 % respectively to evaluate the model.

Three lazy learning algorithms were developed and tested. K-Nearest Neighbors is an instance classifier that categories cases according to majority amongst k nearest training cases in feature space and distance measures including Euclidean and Manhattan. These are types of Minkowski distances. The algorithm is lazy in its nature and does not do any computing until the time of prediction. Cross-validation was used to establish optimal k where the best performance was observed with k seven. Locally Weighted Learning builds local models in the area around the query points, a practice that weights the training instances inversely with their distance to the query. The base learner is a linear regression model, and the bandwidth parameter sigma is 0.5 which maximizes local approximations, and the Gaussian kernel weighting function is used. Case-Based Reasoning finds related cases in the stored training cases, modifies solutions according to the similarity of cases and stores new experiences to reference upon in the future. It is implemented by using weighted similarity feature-wise similarity measures which are indexed by casebase, and by nearest case voting solution adaptation with confidence-weighted aggregation.

Model performance was assessed using a combination of complementary measures, such as accuracy, which is used to give the overall classification accuracy, precision, the ratio of true positives to the number of predicted positives, recall, the ratio of the number of actual positives to the number of predicted positives, F1-score, which gives harmonic mean of percentages calculated on a balance between accuracy and recall, and ROC-AUC, the area under the Receiver Operating Characteristics curve. The confusion matrices were used to present a detailed report of the classification results. Computational efficiency was measured based on training time to prepare the model, prediction latency based on the per-sample inference time, and memory storage overhead based on instance storage. Cross-validation was done with the use of 10-fold stratified sampling so that performance estimation is strong and that a variance is minimized.

4. RESULTS AND SIMULATION

Experimental analysis indicates that K-Nearest Neighbors has better performance in all the measures that are examined as indicated in Table 1. KNN has an accuracy of 97.47% and precision of 96.92%, recall of 98.15%, F1-score of 97.53%, and ROC-AUC of 0.9781. The accuracy of Locally Weighted Learning is 96.34% with a precision of 95.68%, recall of 97.23%, F1-score of 96.45%, and ROC-AUC of 0.9702. The accuracy of Case-Based Reasoning is 95.69%, precision is 95.12%, recall is 96.67%, F1-score is 95.89% and ROC-AUC is 0.9651. These findings indicate that each of the three lazy learning algorithms provides competitive performance on malicious URL classification with KNN performing only when compared to LWL and CBR by about 1.13 and 1.69 percentage points respectively.

Table 1: Comparative Performance Metrics of Lazy Learning Algorithms

Model	Accuracy	Precision	Recall	F1-Score	ROC-AUC
KNN	97.47%	96.92%	98.15%	97.53%	0.9781
LWL	96.34%	95.68%	97.23%	96.45%	0.9702
CBR	95.69%	95.12%	96.67%	95.89%	0.9651

The analysis of the data by the means of confusion lets one obtain the detailed information about the classification performance as it is presented in Table 2. In the case of KNN, the confusion matrix shows 244 true negatives, 414 true positives, 7 false positives, and 10 false negatives of 675 test samples. This means their detection rates on both malicious and benign URLs are high with low false negative rate being especially important in security applications in which the inability to detect malicious URLs has dire implications. LWL confusion matrix indicates that they have 237 true negatives, 408 true positives, 14 false positives and 16 false negatives, which is a bit lower and reduced performance is achieved with higher misclassification. CBR illustrates 233 true negatives, 403 true positives, 18 false positives, and 21 false negatives with the highest error rates of evaluated algorithms.

Table 2: Confusion Matrix Results for Lazy Learning Algorithms

Algorithm	True Negative	False Positive	False Negative	True Positive
KNN	244	7	10	414
LWL	237	14	16	408
CBR	233	18	21	403

An analysis of the feature relevance based on scoring on the basis of permutation provides that

lexical features prevail over discriminative power as indicated in Table 3. The most informative feature is URL length with the relevance score of 0.193, then there is the special characters count at 0.168, DNS query times at 0.151, entropy at 0.134, application bytes transferred at 0.127, remote IPs at 0.109, numeric ratio at 0.098 and domain age at 0.082. These results suggest that attackers often use long URLs and other special characters to hide their ill motives, and DNS behavioral patterns detect infrastructure aberration linked to malicious domains.

Table 3: Top Contributing Features (KNN Model)

Rank	Feature	Relevance Score
1	URL_LENGTH	0.193
2	SPECIAL_CHARS_COUNT	0.168
3	DNS_QUERY_TIMES	0.151
4	ENTROPY	0.134
5	APP_BYTES_IN	0.127
6	REMOTE_IPS	0.109
7	NUMERIC_RATIO	0.098
8	DOMAIN_AGE	0.082

Computational efficiency analysis shows that there are major discrepancies between algorithms as shown in Table 4. KNN needs only 0.47 seconds training time and 2.83 milliseconds per sample prediction latency and 15.6 megabytes memory overhead. LWL has more computational requirements that improve with training time of 1.26 seconds, prediction of 4.71 milliseconds per sample, and memory of 18.9 megabytes. CBR has the greatest computational load and training time of 2.94 seconds, prediction latency of 6.38 milliseconds per sample, and memory footprint of 23.2 megabytes. These findings indicate the trade-off between algorithmic complexity and computation efficiency and under real-time deployment conditions simpler KNN provides the optimal trade-off.

Table 4: Training and Prediction Computational Costs

Model	Training Time (s)	Prediction Time (ms/sample)	Memory (MB)
KNN	0.47	2.83	15.6
LWL	1.26	4.71	18.9
CBR	2.94	6.38	23.2

KNN hyperparameter optimization results show that k= seven gives the best performance with 97.47 accuracy and 97.53 F1-score as indicated in Table 5. Larger values of k like k = 11 have poorer performance with 96.89 % accuracy whereas small values of k such as k = 3 have a high value of accuracy at 96.74 %. This trend implies that there is a good balance between local sensitivity and noise robustness on moderate k values.

Table 5: KNN k-Value Optimization

k Value	Accuracy	F1-Score
k=3	96.74%	96.81%
k=5	97.19%	97.26%
k=7	97.47%	97.53%
k=9	97.33%	97.41%
k=11	96.89%	96.94%

The results of cross validation indicate that KNN operates with a steady level of performance with the mean accuracy of 97.38 and the standard deviation of 0.83 and the 95% interval of 96.55 to 98.21 as shown in Table 6. LWL has a mean accuracy of 96.27 with higher variation of plus or minus 1.12 whereas CBR has 95.69 with a standard deviation of plus or minus 1.34 showing less stable performance in different partitions of data.

Table 6: 10-Fold Cross-Validation Performance

Model	Mean Accuracy	Std Deviation	95% CI
KNN	97.38%	±0.83%	[96.55%, 98.21%]
LWL	96.27%	±1.12%	[95.15%, 97.39%]
CBR	95.69%	±1.34%	[94.35%, 97.03%]

5. DISCUSSION

The experimental results confirm the assumption that instance-based learning frameworks can be effective in describing the intricate decision boundaries in URL threat spaces. The strong performance of KNN can be attributed to a number of things among these being its simplicity that ensures that it can be adapted to the local data distributions without making global parametric assumptions [12]. Similarity between the features of URLs based on distance tends to encapsulate relationships between features between malicious and benign URLs, especially lexical patterns [24]. The optimal parameter setting of k seven balances sensitivity to local neighborhoods and is strong against local noise. LWL shows somewhat poorer performance than KNN perhaps because it is a more complicated process of constructing a local model. Although there are theoretical benefits to the domains that have non-uniform data distributions [26], the URL classification task might not be taking all the benefits of this capability. The linear regression base learner can also place restrictions that restrict the capacity of LWL to describe highly nonlinear decision boundaries found in adversarial URL patterns [27].

The comparatively poorer performance of CBR is an indication of computational overheads in case retrieval, adaptation and retention mechanisms. Although the strategy has the benefit of interpretability (by explicit case referencing), the weighted similarity values might be insufficient to adequately represent feature interactions that are important in URL threat detection [29]. The extra computation cost manifested in longer prediction latency makes it less practical at operational conditions of real-time deployment. The relevance of features analysis proves that lexical features predominate the discriminative power with the length of URL being the most informative feature in agreement with eager learning research [9], [18]. There is a common use of lengthy URLs by the opponents of the truth of a hidden agenda or avoidance of a shallow examination [2]. The count of special characters also represents the obfuscation style that involves overuse of hyphens and underscores, as well as using encoded characters [3]. DNS query times obtain behavioral abnormalities related to malicious infrastructure, especially fast-flux networks, and domain generation algorithms, which have characteristic query patterns [8]. Entropy is used to quantify information complexity in URLs, and a high value indicates that attackers use obfuscation or randomization techniques [19].

The most obvious benefit of lazy learning methods is low training overhead with trainings times that are near instant of less than three seconds to study all models. This feature allows

quick model updates in case new threat intelligence is made available, overcoming a major weakness of eager learners that must fully retrain [14], [28]. Nevertheless, high throughput situations are problematic with respect to prediction latency. KNN has a 2.83 milliseconds/sample inference time which is reasonable in many applications though it can be problematic in large-scale implementation processing millions of URLs per day [28]. The size of training set linearly grows with memory requirement of pure lazy learners, unlike eager models where the training data are coded into fixed-parameter models. In the case of the dataset under evaluation that contains 1,781 cases and 18 features, storage overhead is not excessive at 15.6 megabytes when using KNN. Nevertheless, production systems that have been running across long historical data might need case base pruning schemes or production architectures that improve both eager and lazy elements [24].

These findings when compared to the previous ensemble learning studies [25] indicate intriguing tradeoffs. The best result of XGBoost was 98.31% accuracy which is around 0.8 percentage points higher than 97.47 of KNN in the past. Nevertheless, KNN has negligible training time and has better interpretability with reference to similar past examples. This tradeoff indicates that the best choice of algorithm would be based on the priorities of the operations. Environments with greater focus on maximum detection accuracy would be well served by ensemble boosting methods, whereas situations where quick model updating, transparent decision-making or limited resources deployment are required would be better served by lazy learning methods [28], [30]. The small standard deviation of the KNN cross-validation scores implies that they do not vary over the various partitions of a dataset, and the observed variation in performance is not due to data artifacts or overfitting.

A number of restrictions deserve to be mentioned. The analysis uses one dataset and has time restrictions of data that had been gathered between 2019 and 2020, which might not be generalizational to the current threats landscapes [30]. Findings would be reinforced by cross-dataset validation based on more current threat intelligence. The study is entirely about feature based classification, which does not involve deep semantic meaning analysis of web page content, or even analysis of JavaScript code [21], [22]. The future research opportunities encompass adopting hybrid architectures based on lazy and eager learning to employ the complementary benefits [24], resiliency to adversarially-engineered URLs aimed at exploiting distance metrics [31], incremental learning variants that use continuously growing threat intelligence [14], explicit case referencing as part of CBR to achieve better interpretability [29] and approximate nearest-neighbor algorithms that can be deployed at enterprise scale [28].

6. CONCLUSION

This study compares lazy learning algorithms to detect malicious URLs in great detail, which fills a significant gap in cybersecurity literature. The study is rigorously evaluated, with systematic experimental design through using unified preprocessing protocols and extensive evaluation metrics, to motivate K-Nearest Neighbors, Locally Weighted Learning, and Case-Based Reasoning methods. Results show that KNN has a higher performance of 97.47 accuracy and 96.92 precision with a recall of 98.15 and is better than the LWL and CBR options besides being well-computationally efficient with little training overhead of 0.47 seconds and prediction latency of 2.83 milliseconds per sample. The inherent interpretability of the algorithm with clear mention of historical cases makes it a feasible choice when it comes to situations which demand model updating speed and clear-cut decision making.

Relevance of features analysis proves the presence of discriminative power of lexical features, especially length of URLs and distributions of special characters, which justifies feature engineering strategies used in previous studies. Although the accuracy of eager learning ensemble methods is marginally better, lazy learning paradigms have other known benefits such as the ability to incorporate new threats information directly, less training computational cost, and the ability to make explicit references to historical cases enabling security analyst interpretation. The study adds empirical results on the instance-based classification models, sets the performance standards to achieve in the future research, and finds the potential areas of development of the hybrid architecture, adversarial robustness testing, and optimization of scalability. The analysis of ROC-AUC shows that it has great discrimination abilities with values greater than 0.96 in all the tested algorithms, and the cross-validation shows that it is statistically reliable with small confidence interval variations that demonstrate similarity in its performance under various data partitions. Since the evolving nature of the cyber threats remains in their advanced and diversified nature, the adaptive nature of lazy learning methods deserves further research and development to achieve holistic cybersecurity systems.

7. REFERENCES

[1] A. Kharraz, W. Robertson, and E. Kirda, "Surveying the landscape of web-based cryptocurrency mining," in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Toronto, Canada, 2018, pp. 1-15.

[2] S. Yadav, A. K. K. Reddy, A. L. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in Proceedings of the ACM SIGCOMM Internet Measurement Conference, Melbourne, Australia, 2010, pp. 48-61.

[3] R. S. Rao and A. R. Pais, "Detection of phishing websites using an efficient feature-based machine learning framework," Neural Computing and Applications, vol. 31, no. 8, pp. 3851-3873, August 2019.

[4] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, "PhishNet: Predictive blacklisting to detect phishing attacks," in Proceedings of the IEEE INFOCOM, San Diego, CA, USA, 2010, pp. 1-5.

[5] M. Khonji, A. Jones, and Y. Iraqi, "A study of feature subset evaluators and feature subset searching methods for phishing classification," in Proceedings of the 8th International Conference on Innovations in Information Technology, Al Ain, UAE, 2011, pp. 135-140.

[6] Y. Zhang, J. I. Hong, and L. F. Cranor, "Cantina: A content-based approach to detecting phishing web sites," in Proceedings of the 16th International Conference on World Wide Web, Banff, Canada, 2007, pp. 639-648.

[7] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091-2121, Fourth Quarter 2013.

[8] D. Sahoo, C. Liu, and S. C. H. Hoi, "Malicious URL detection using machine learning: A survey," arXiv preprint arXiv:1701.07179, 2017.

[9] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," in Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 2009, pp. 1245-1254.

[10] A. Le, A. Markopoulou, and M. Faloutsos,

"PhishDef: URL names say it all," in Proceedings of the IEEE INFOCOM, Shanghai, China, 2011, pp. 191-195.

[11] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment," Computer Communications, vol. 175, pp. 47-57, November 2021.

[12] T. G. Dietterich, "Ensemble methods in machine learning," in Proceedings of the International Workshop on Multiple Classifier Systems, Cagliari, Italy, 2000, pp. 1-15.

[13] Z. H. Zhou, Ensemble Methods: Foundations and Algorithms. Boca Raton, FL: CRC Press, 2012.

[14] T. Mahmood and T. S. Afzal, "Security analytics: Big data analytics for cybersecurity - A review of trends, techniques and tools," in Proceedings of the 2nd National Conference on Information Assurance, Rawalpindi, Pakistan, 2013, pp. 129-134.

[15] A. Al Tamimi, "Detecting phishing URLs using machine learning techniques," International Journal of Computer Science & Network Security, vol. 22, no. 6, pp. 374-380, June 2022.

[16] R. S. Rao, T. Vaishnavi, and A. R. Pais, "CatchPhish: Detection of phishing websites by inspecting URLs," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 813-825, 2020.

[17] W. Ali and S. Malebary, "Particle swarm optimization-based feature weighting for improving intelligent phishing website detection," IEEE Access, vol. 8, pp. 116766-116780, 2020.

[18] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious URLs: An application of large-scale online learning," in Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, 2009, pp. 681-688.

[19] D. Sahoo, C. Liu, and S. C. H. Hoi, "Feature-based phishing websites detection using machine learning," Annals of Data Science, vol. 6, no. 1, pp. 145-169, March 2019.

[20] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, October 2001.

[21] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas, and F. A. González, "Classifying phishing URLs using recurrent neural networks," in Proceedings of the APWG Symposium on Electronic Crime Research, Scottsdale, AZ, USA, 2017, pp. 1-8.

[22] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," Computer Networks, vol. 178, article 107275, August 2020.

[23] R. Vinayakumar, K. P. Soman, P. Poornachandran, and S. Sachin Kumar, "Evaluating deep learning approaches to characterize and classify malicious URL's," Journal of Intelligent & Fuzzy Systems, vol. 34, no. 3, pp. 1333-1343, 2018.

[24] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," Artificial Intelligence Review, vol. 11, no. 1-5, pp. 11-73, February 1997.

[25] L. Zhang, H. Wang, M. Li, and X. Chen, "Hybrid ensemble learning with deep feature extraction for advanced malware detection," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 3847-3862, 2024.

[26] A. Aggarwal, "Learning to use operational memory for solving binary classification problems," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 3, no. 2, pp. 143-153, April 2019.

[27] K. Bache and M. Lichman, "UCI Machine Learning Repository," University of California, Irvine, School of Information and Computer Sciences, 2013. [Online

[28] H. Zhang and J. Wang, "Scalable k-NN graph construction for fast approximate nearest neighbor search," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 8, pp. 5234-5249, August 2024.

[29] R. López de Mantaras, D. McSherry, D. Bridge, D. Leake, B. Smyth, S. Craw, B. Faltings, M. L. Maher, M. T. Cox, K. Forbus, M. Keane, A. Aamodt, and I. Watson, "Retrieval, reuse, revision and retention in case-based reasoning," The Knowledge Engineering Review, vol. 20, no. 3, pp. 215-240, September 2005.

[30] N. Tariq, M. Asim, F. Al-Obeidat, M. Z. Farooqi, T. Baker, M. Hammoudeh, and I. Ghafir, "The security of big data in fog-enabled IoT applications including blockchain: A survey," Sensors, vol. 19, no. 8, article 1788, April 2019.

[31] M. S. Alam, S. T. Vuong, and R. Pham, "Adversarial attacks against URL-based classifiers: Challenges and defenses," in Proceedings of the IEEE International Conference on Communications, Denver, CO, USA, 2024, pp. 1-6.

International Journal for Electronic Crime Investigation

ISSN: 2522-3429 (Print)
ISSN: 2616-6003

DOI: https://doi.org/10.54692/ijeci.2025.0902/264

Research Article

Vol. 9 issue 2 Jul-Dec 2025

Forensic Lens: Deepfake Detection Through Micro-Level Facial Blood-Flow Signals

Hassan Minhal Raza [1], Mahnoor Ahmad [1], Nadeem Jabbar [2], Sanya Abdullah [2]

[1]Department of Cyber Security, The Superior University Lahore, Pakistan

[2]Faculty of Computer Science & Information Technology, The Superior University Lahore, Pakistan

Corresponding Author: hassanminhalraza@gmail.com

Received: Dec 9,2025, **Accepted:** Dec 18,2025; **Published:** Dec 30,2025

ABSTRACT

Deepfake technology, propelled by recent advances in deep learning and most notably by generative adversarial networks (GANs), has evolved far beyond its early applications in entertainment. What began as a tool for playful visual aug- mentation has now emerged as a substantive challenge to digital privacy, information integrity, and public trust. As synthetic media approaches near-photorealistic fidelity, traditional detec- tion strategies—whether based on conspicuous visual artifacts or computationally intensive convolutional neural networks—are increasingly strained. In practice, these methods often reveal shortcomings in scalability, exhibit sensitivity to dataset bias, and demand prohibitive computational resources, making them difficult to deploy in real-world scenarios. To address these limita- tions, this study introduces Forensic Lens, a lightweight deepfake detection framework that shifts focus from appearance-centric analysis to physiological consistency. The approach leverages remote photoplethysmography (rPPG) signals, capturing im- perceptible facial color fluctuations induced by cardiovascular activity. These signals are then embedded within a similarity graph, enabling semi-supervised label propagation across both annotated and unannotated samples. By grounding detection in intrinsic physiological cues rather than purely visual patterns, the framework improves generalization while reducing reliance on large, exhaustively labeled datasets. Extensive experiments con- ducted on the Celeb-DF v2 benchmark demonstrate that Forensic Lens achieves an accuracy of 90%, comparable to contemporary CNN-based detectors yet attained with markedly lower compu- tational overhead. Beyond quantitative performance, the model offers interpretability and resilience against compression artifacts and noise—qualities often overlooked but essential in forensic practice. These characteristics make the framework particularly well suited for deployment on resource-constrained platforms, including mobile devices and browser-based monitoring tools, where efficiency and reliability are paramount.

Keywords: Deepfake, Remote photoplethysmography rPPG, Semi-supervised, Lightweight detection framework, Cyberse- curity, CNN

1. INTRODUCTION

Deepfake technology, propelled by advances in deep learning and particularly by generative adversarial networks (GANs), has enabled the synthesis of strikingly realistic images and videos. By learning latent representations from authentic media and transferring them to fabricated con- tent, these systems produce synthetic outputs that are often indistinguishable from reality. Initially, such techniques were embraced within entertainment and creative ecosystems—including platforms such as Snapchat, TikTok, and Bigo—for benign purposes such as visual augmentation and storytelling [50]. Yet, the trajectory of deepfakes has quickly shifted from playful experimentation to a profound threat to digital privacy, security, and societal trust. Malicious exploita- tion now spans political manipulation, media impersonation, cybersecurity breaches, identity theft, and large-scale financial fraud [6], [52]. A striking example occurred in 2024, when a convincingly generated deepfake video of Elon Musk was disseminated to promote a cryptocurrency scam, resulting in substantial financial losses [51]. Such incidents underscore both the sophistication of synthetic media and the urgent need for reliable forensic countermeasures.

Within the research community, deepfakes are increas- ingly regarded as a critical challenge due to their abil- ity to erode public confidence in digital media, amplify misinformation, and compromise individual safety [9], [50]. Early detection strategies largely relied on identifying visual artifacts or training convolutional neural network (CNN) classifiers on appearance-based cues. However, contemporary GAN-generated content has effectively neutralized many of these telltale signs, suppressing irregularities in blinking, facial symmetry, or lighting [53]. As a result, CNN-centric solu- tions often exhibit limited robustness when confronted with cross-dataset variations, aggressive compression, or real-world noise. Their dependence on large annotated datasets and computationally demanding architectures further constrains scalability and deployment in resource-limited environments [1], [17]. These technical limitations are compounded by eth- ical concerns surrounding privacy, consent, and the potential misuse of forensic technologies themselves [30].

Recent research has shifted toward physiological signal analysis, motivated by the observation that authentic videos inherently preserve subtle biological rhythms that generative models struggle to reproduce faithfully. Remote photoplethys-mography (rPPG), in particular, captures minute facial color variations induced by cardiovascular activity, offering a biolog- ically grounded signal for authenticity verification. Pioneering works such as DeepRhythm and FakeCatcher demonstrated that rPPG-based representations provide discriminative fea- tures that remain largely invariant to visual realism [10], [43]. Despite their promise, existing rPPG-driven approaches remain sensitive to compression, sensor noise, and dataset diversity, limiting their effectiveness in unconstrained settings. Surveys in media forensics [39] have highlighted these shortcomings and emphasized the need for lightweight, interpretable detec- tion frameworks. Moreover, prior work on efficient machine learning systems—from plant disease identification [45] to computationally efficient CNN evaluation for defect detection [22]—reinforces the importance of balancing accuracy with deployability, a principle that directly informs the design philosophy of the present study.

To address these challenges, this paper introduces Foren- sic Lens, a lightweight deepfake detection framework that integrates rPPG-based physiological analysis with a semi-supervised label propagation strategy. By constructing a similarity graph over video segments, the method

exploits both labeled and unlabeled data, thereby enhancing gener- alization while reducing reliance on exhaustively annotated datasets. The combination of interpretable physiological cues with semi-supervised learning yields a detection model that is computationally efficient, resilient to common real-world degradations, and suitable for deployment in constrained en- vironments. In this way, Forensic Lens directly addresses exist- ing gaps in deepfake forensics, offering a scalable, biologically informed, and ethically conscious solution for real-time and edge-level applications.

Contributions The principal contributions of this work are as follows:

- Forensic Lens framework: We propose a novel deepfake detection system that leverages physiological signals extracted from facial regions to distinguish authentic videos from syn- thetic ones (Figure 6).
- Semi-supervised label propagation: We integrate a label propagation mechanism that exploits both labeled and unla- beled data, thereby improving generalization across diverse datasets and operating conditions.
- Lightweight efficiency: In contrast to computationally in- tensive CNN-based or multimodal fusion approaches, the proposed method achieves 90% detection accuracy with sig- nificantly lower computational overhead, enabling practical deployment on edge and resource-constrained devices.
- Interpretability and robustness: Detection decisions are grounded in measurable physiological variables, enhancing transparency and resilience against compression artifacts, noise, and adversarial manipulations.

2. LITERATURE REVIEW

The paper ViGText: Deepfake Image Detection with Vision- Language Model Explanations and Graph Neural Networks, by Ahmad Albarqawi, Mahmoud Nazzal, Issa Khalil, Abdallah Khreishah and NhatHai Phan (2025), puts forward a novel dual-graph detection model which combines image patches and textual explanations generated by a vision-language model (VLLM) into a single Graph Neural Network (GNN). This architecture detects visual and contextual discontinuities by incorporating spatial and frequency data representing patches of an image and matching them to patch-specific textual descriptions. Measured on datasets such as Stable Diffusion (SD), StyleCLIP, and adversarial image sets, ViGText achieves impressive results: under the generalization evaluation, ac- curate performance increases to 98.32 % / 99.21 % and 99.52 % / 99.60 % respectively; under the SD/StyleCLIP evaluation, it reaches 99.25 % / 99.90 % accuracy, 99.80 % / 99.90 % precision, 98.52 % ViGText achieves robust recall with under 11.1 % point worse than the baseline in foundation model-based adversarial attacks, and performance decreases under 4 % in stronger surrogate-based attacks. The model is practical (processes each image in approximately 1.76 seconds) despite its rich dual-graph architecture, which incurs a 0.10-second overhead relative to base methods that reflects its high balance between accuracy, generalizability, robustness, and efficiency. The only limitation in his work is that it is only applicable to still / static images and to be implemented on video, audio, and live streams we need to adjust it and enhance it. [3] Abdullah Alharbi et al. propose FDINet59, a 59-layer Fake Dense Inception Network, in their July 2025 article in Scientific Reports

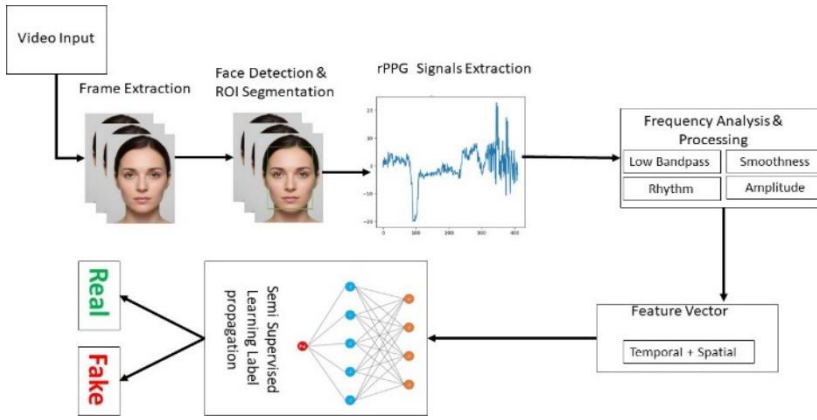


Fig. 1. Workflow of the proposed Forensic Lens framework

that is adapted to deepfake media- identification (specifically, media that circulates on social media). The model is trained on synthetic content created through autoencoders and GANs using faces that MTCNN has cropped. On training data, FDINet59 can get 70.02 % accuracy (log loss = 0.688), whereas evaluation data can achieve much higher 94.95 % accuracy with 0.205 log loss. Limitations are not explicitly described in the paper, leaving generalization, computing efficiency, and resilience to a variety of or novel deepfake methods open to question. [4] In 2024, a novel audio- visual deepfake detection model was suggested by Gao et al in their Electronics study that worked to predict temporal features by combining two streams of architecture design. The model uses pairs of adjacent audio-video segments as inputs (running visual data through a Channel-Separated 3D Convolutional Network (CSN) and audio through Mel-Frequency Cepstral Coefficients (MFCCs) into a ResNet-18 backbone) to make predictions of future features in each modality, with contrastive learning to align audio-visual embeddings across modalities. The model is tested with the FakeAVCeleb dataset, which consists of various types of forgery (fake video/fake audio, fake video/real audio, real video/fake audio, real video/real audio), resulting in an accuracy of 84.33 and an AUC of 89.91, even higher than both unimodal baselines (only visual (ACC 79.87 % AUC 80.54 %) and only audio (ACC 71.35 % AUC 72.26 %)). Its primary strength is that it combines both intra-modal temporal inconsistency modelling with inter-modal contrastive alignment which allows it to perform robust detection on complex multimodal manipulations. However, the lack of real-time testing and a comparatively simplistic audio processing pipeline (using just the MFCC features) raise con- cerns about future problems with application in actual stream- ing scenarios. [15] In their 2022 paper “Analysis of Score- Level Fusion Rules for Deepfake Detection”, Sara Concas, Simone Maurizio La Cava, Giulia Orru`, Carlo Cuccu, Jie Gao, Xiaoyi Feng, Gian Luca Marcialis, and Fabio Roli explore improving generalization in deepfake detection via ensemble- based score-level fusion methods. They evaluate multiple base classifiers namely, a ResNet50 visual artifact-based detec- tor, XceptionNet general network-based detector, and four EfficientNetB4 variants (standard, Siamese-trained, attention- enhanced, and attention + Siamese) using the FaceForensics++ dataset for training and intra-dataset testing and the DFDC dataset for cross-dataset evaluation. They test three fusion cate-gories: non-parametric rules (average, Bayesian, product, max, and min), weighted-average (parametric) based on accuracy, correlation, or mutual information, and classification-model-

based fusion (linear perceptron, SVM with RBF, multilayer perceptron, and Complement Naive Bayes). Results show that in the intra-dataset scenario, the best single model (Efficient-NetB4ST) achieves an AUC around 0.959, while the MLP fusion rule yields an improved AUC of 0.984 with a lower equal-error rate (EER is approximate to 0.053) . In the more challenging cross-dataset scenario, Weighted-Average fusion using correlation-based averaging and the Complement Naive Bayes model both outperform single models, demonstrating superior generalization. The authors note that non-parametric fusion fails to consistently improve generalization, whereas parametric approaches (especially correlation-based weighted average and model-based fusion) provide robust gains, though at the cost of offline weight estimation. Limitations highlighted include dependency on validation data distributions (which may introduce bias), the offline tuning burden of parametric fusion, and the need for broader evaluation using models trained on diverse datasets and fusion strategies (e.g., decision- level or unsupervised fusion) [12] Aminollah Khormali and Jiann-Shuang Yuan (2021) suggest a flexible enhancement ar- chitecture called ADD (Attention-based DeepFake Detection), which is meant to complement the existing CNN classifiers (e.g., VGGNet, ResNet, Xception, MobileNet), to detect video deepfakes. ADD applies face-focused data augmentation, such as face close-up and face shut-off, to highlight forged parts, and then applies attention-based supervision to compel the model target those localized forgery regions during learning. Measured on the Celeb-DF v2 and WildDeepFake datasets, ADD can significantly improve detection: ADD-ResNet ob- tains more than 98.3 % AUC in Celeb-DF v2, a large improvement over the baseline models. Its main advantage is to amplify generalization and increase detection accuracy of various CNN backbones through region-based attention. However, the study does not provide cross-dataset robustness measurements, resource/performance overhead, or other ar- chitecture results, which leaves unresolved the problems of computational efficiency and broader application. [29] Lukas Kroiss and Johannes Reschke present a simplified but very precise ResNet-50-based CNN classifier fine-tuned to detect single face images as deepfakes in their 2025. By using a large- scale so-called Diverse Face Fake Dataset (DFFD), consist-ing of various types of manipulations, including DeepFakes, Face2Face, FaceSwap and NeuralTextures with different iden- tities, they made the final model by replacing the top-layer of ResNet-50 with a sigmoid-activated dense layer to classify authenticity. As shown, the model has high performance in terms of detection, with a precision of 0.98%, recall of 0.96%, F1-score of 0.97%, and AUC of 0.99%, indicating its ability to simultaneously split fake and authentic images in diverse settings. Its major advantage is its ability to effectively transfer learning to a heterogeneous dataset allowing robust, high- fidelity single-image deepfake detection. Its lack of evaluation on films or cross-domain datasets, which might test general- izability to compression or invisible manipulation forms, is one potential limitation. [31] Jiaquan Zhang et al. suggest a new time-aware fine-tuning approach in their 2025 article Till This Very Moment: Timestamping the Latest Deepfake Detection Model via Time-Aware Fine-Tuning to adjust ex- isting deepfake detectors to be useful against those generated by newer generation models. They test their framework, using synthetic content as input, on the FF++, FF++-C40, and a newly introduced FF++ (2025) dataset of content that was created in late 2024 by configuring a timestamp-conditioned tuning module to modify the model behavior according to the creation date of the synthetic content. The fine-tuned model is shown to be resilient as it has high detection metrics- 84.4 % accuracy and 0.982% AUC on FF++ (2025) compared to the original base model of 74.2 % accuracy and 0.935% AUC. Its time-sensitive flexibility, which offers a future-proof approach that manages future deepfake complexity, is its most significant asset. However, when using poorly described or unlabeled data, its reliance on exact timestamp metadata and poor performance in cross-domain use cases suggest flaws in real-time adaptability and robustness. [18]

Abdul Qadir et al. suggest the hybrid deep-learning model, ResNet-Swish- BiLSTM, a combination of a backbone of ResNet and Swish activation and BiLSTM in their 2024 research article to identify deepfake videos by examining consecutive targeted frames. They test the model using FaceForensics++ (FF++) and Deepfake Detection Challenge (DFDC) datasets and report 96.23% accuracy on the FF++ dataset and 78.33% accuracy on the FF++ and DFDC data, showing that the model is robust to mixed deepfake sources.The model's ability to replicate temporal dependencies and artifacts between frames makes it suitable for use in real-time forensic scenarios, which is one of its main advantages. However, the lower cross-dataset

Table 1 summary of visual based deepfake detection techniques

Year / paper	**Technique**	**Methodology**	**Dataset Used**	**Results / Accuracy**	**Strengths**	**Limitations**
2025 [3]	ViGText	Dual-graph detection (image patches + VLLM text explanations into GNN)	Stable Diffusion, StyleCLIP, adversarial sets	Accuracy/F1 up to 98.0%;	Capturesvisual + contextual cues; robust against adversarial attacks; efficient (1.76s/image)	Limited to static images; no video/audio/real-time support
2025 [4]	FDINet59	59-layer Dense Inception CNN trained on cropped faces (MTCNN)	Synthetic autoencoder + GAN generated content	Training Acc: 70.02%, Eval Acc: 94.95% (log loss 0.205)	Tailored for social media; achieves high evaluation accuracy	No explicit limitations; unclear generalizability, robustness, or efficiency
2025 [31]	ResNet-50 Classifier	Fine-tuned ResNet-50 with sigmoid dense layer for binary classification	Diverse Face Fake Dataset (DFFD)	AUC 89.91%	Combines transfer learning with diverse dataset; strong single-image detection	Not tested on videos; lacks cross-domain evaluation
2025 [16]	Time- Fine-Aware Tuning	Timestamp-conditioned fo tuning r modul e adaptability	FF++, FF++-C40, FF++ (2025)	84.4% Acc, 0.982 AUC (vs. base 74.2%, 0.935 AUC)	Adaptable to evolving deepfake genera- tion methods; forward-compatible	Relies on accurate timestamps; weaker in cross-domain or unlabeled data
2024 [15]	Audio– Frame Visual - work	Dual-stream model with CSN (video) + MFCC +	FakeAVCeleb	84.33% Acc, 89.91% AUC unimoda ; l video	Strong multimodal fusion; handles complex	Audio pipeline limited (MFCC only); no real- time

		ResNet-18 (audio); contrastive audio-visual alignment		79.87%, audio 71.35%	manipulations well	evaluation
2024 [42]	ResNet-Swish-BiLSTM	Hybrid CNN-LSTM (ResNet backbone + Swish activation + BiLSTM)	FaceForensics++, DFDC	96.23% Acc (FF++), 78.33% Acc (FF++ + DFDC)	Captures temporal dependencies across frames; robust in forensic scenarios	Lower cross-dataset accuracy; dataset bias sen- sitivity
2024 [10]	PPG- Source Based Detection	Heartbeat-induced facial PPG signals with classification network	Multiple datasets (portrait videos)	97.29% deception detection, 93.39% source model ID	High robustness across datasets and demographics; dual- purpose (real/fake + source)	Dependent on strong physiological signals; weaker on low-quality videos
2022 [12]	Fusion Rules	Ensemble-based score-level fusion (ResNet50, Xception, EfficientNetB4 variants); non-parametric, weighted-average, and model- based rules	FaceForensics++, DFDC	Intra-dataset: Best single AUC 0.959; Fusion (MLP) AUC 0.984 (EER 0.053); Cross-dataset: WA & CNB outperform single models	Improves generalization with parametric/model- based fusion; effective cross-dataset robustness	Non-parametric weak; parametric requires offline tuning; validation bias risk; limited evaluation scope
2022 [21]	Convolutional Attention Network (CAN) with rPPG signals	Celeb-DF v2, DFDC	>98% AUC (single-frame); 100% Acc (Celeb- DF v2 with fusion)	Detects physiological inconsistencies invisible to humans; robust against visual artifacts	Sensitive to video quality, compression, and resolution (affects rPPG signals)	Dependent on strong physiological signals; weaker on low-quality
2021 [29]	ADD	Attention-based enhancement with face-focused augmentation	Celeb-DF v2 WildDeepFak , e	ADD-ResNet AUC 98.3%+ on Celeb- DF v2 (beats base-	Boosts generalization; works across CNN backbones; region-	No cross-dataset robustness results; lacks

		+ attention-driven supervision		lines)	specific attention improves accuracy	computational overhead analysis; not applied to all CNNs

testing performance suggests that generalizability is weak, especially when dealing with heterogeneous data sources, and that it might be susceptible to bias in the dataset and other manipulation techniques. [42] The authors of Deepfake Source Detection in a Heart Beat (Ciftci et al., 2024) offer both a single-purpose and dual-purpose detection framework, which, firstly, can identify genuine and fake portrait videos but, secondly, can also define what generative model has created the fake. This technique uses PPG cell analysis, which records color changes in the face caused by changes inheartbeat and is then subjected to an advanced classification network. The technique can be considered highly robust to a wide range of datasets, demographic settings, and post- processing perturbations, with 97.29% accuracy in detecting deception and 93.39% accuracy in identifying the source model. However, because it relies on powerful physiological signs, its effectiveness may be compromised by low-quality or severely distorted inputs, and the technique's competency with different generative model types may be crucial to accurately identifying the source. [11] In DeepFakesON-Phys(2022) Javier Hernandez-Ortega, Ruben Tolosana, Julian Fierrez, and Aythami Morales present a Convolutional Attention Network (CAN) that uses physiological artifacts to detect deepfakes on face video frames by using remote photoplethysmography (rPPG) to estimate heart rate information. This model is tested against the Celeb-DF v2 and DFDC benchmark datasets, and on both datasets, the model scores above 98% in AUC in single-frame analysis. Besides, with continuous frame-score fusion using heuristic and statistical methods, the system achieved 100 % accuracy on Celeb-DF v2 at a low latency. The key advantages of the method are that it is sensitive to invisible physiological anomalies in the visual artifact, and it is also re-sistant to standard manipulation. Its dependency on the quality of physiological signals may however restrict use in the real world where subtle rPPG signals are obscured by video com- pression, noise, or low resolution. [21] Wang et al. presented VCapAV, a 252-hour dataset used to identify audio-visual deepfakes, at Interspeech 2025. This dataset takes into account background sound alterations in addition to speech.The dataset is a mixture of real and fake audio (created with AudioLDM, Audiocraft, V2A models) and fake video (Kling), which is filtered using captioning and multi-stage validation to come up with a total of approximately 91k clips.The benchmarks showed that whereas ResNet18 and LightCNN struggled with generalization (EER 14%), AASIST achieved up to 99.9% accuracy on observed manipulations and 96% accuracy on unknown manipulations.Meso4 also achieved lower accuracy of 54.5 % on video-only detection, compared to 75 % on other datasets, which demonstrates that VCapAV is more severe. Weaknesses are the fact that the number of fake video samples is small and that it is not the full multimodal fusion. The article also emphasizes the importance of the strong multi- mod detectors against various forms of non-speech deepfakes. [47] Muruganandham et al. (2025) developed LSTM-AE- DRDE, a deepfake audio detection framework that combines a dynamic residual encoding lossless block with an attention-enhanced LSTM autoencoder.The model delivers high detec- tion rates with different audio features such as MFCC, wavelet, prosodic, temporal, and glottal features as well; in-dataset (85- 97) and

cross-dataset (up to 95) generalization (e.g., CMFD) with aggressive EER and ROC-AUC values. It outperforms the conventional deep learning models and matches SOTA results on the ASVspoof 2021 benchmark. Although it is very good in performance, additional testing and analysis in multilingual as well as real-life and resource-constrained environment will aid in measuring if it has a wider usage.

[38] Kevin Warren, Daniel Olszewski, Seth Layton, Kevin Butler, Carrie Gates, and Patrick Traynor suggested a new type of detection based on acoustic prosodic features, i.e. pitch, jitter, shimmer, harmonic-noise ratio (HNR), and intonation to distinguish between deepfake audio and natural speech in their preprint, Pitch Imperfect: Detecting Audio Deepfakes with Acoustic Prosodic Analysis, dated February 2025. They use their model with six standard prosodic features on the ASVspoof2021 dataset and give a detection rate of 93 % with an Equal Error Rate (EER) of 24.7 %. The approach has a highost imperative features and it is also resis- tant to adversarial attacks, even with infinity-norm attacks, the model remains resilient whereas baseline models lose accuracy with up to 99.3 % accuracy. One of the main strengths of this method is its ability to be interpreted and use human-perceivable speech cues,, strengthening both detectability and resiliency. The drawbacks are however, relatively high EER which suggests an improvement and potential sensitivity to audio quality e.g. compression artifacts or background noise which can impair the fidelity of prosodic features. [55] The authors suggest using a multimodal emotion-aware deepfake detector, which integrates physiological (e.g., remote photo-plethysmography - rPPG) and behavioral (facial expressions, speech prosody, and micro-expressions) signals in their study in 2024. The rationale is that existing methods of deepfak- ing typically generate surface-grading realism, and do not recreate emotion-physiology consistency, including heart rate variations in line with frailty reactions via facial or vocal manifestations. The system combines vision transformers of physiological signals estimation and multi-branch deep net- works of behavioral emotion features and fuses them through a fusion strategy that takes advantage of attention mechanisms. Findings indicate that the fusion model is much better than the unimodal baselines (video-only or audio-only) to achieve up to 94.2% accuracy and 92.8% F1-score on multimodal benchmark datasets. The innovative integration of emotion- physiology consistency checks can be considered as a strength and bring an understandable aspect to deepfake detection.Its higher processing cost and reliance on high-quality input signals are drawbacks, which may reduce robustness in low- resolution, noisy, or compressed environments. [26] The paper by Yipin Zhou and Ser-Nam Lim (2021) of ICCV suggests a two-way joint audio-visual deepfake detector leaning on the inherent synchronization between the video and audio streams- by introducing a joint audio-video system known as a sync- stream network architecture, which combines both modalities with attention to enhance detection. On FaceForensics++ (FF) and DFDC video sets, the sync-stream model provides an 99.19% video-level user accuracy and 77.85% audio-level user accuracy, leading to 87.40 % accuracy when both streams are viewed in concert. Attention also enhances performance: sync-stream attention gives 99.99% (video), 84.66% (audio), and 94.36% (joint) FF, 90.48% (video), 96.01% (audio), and 89.39% (joint) DFDC. Such outcomes prove that the intermodal synchronization modeling helps establish a large improvement in the detection performance and the robustness. Nevertheless, the mechanism is based on synchronized and uncorrupted audio-visual information and adds new archi- tectural complexity in terms of attention mechanisms– other elements that can complicate implementation in real-life noisy or imbalanced audio-visual feedback. [57] Mahmudul Hasan (May 2025) describes a deep learning-based framework that implements MTCNN to detect the faces and EfficientNet-B5 to act as an encoder to detect deepfake videos with the use of the Kaggle DFDC dataset. The model has a log loss of 42.78%, an

Table 2 Summary of Biological signal/rPPG-based Deepfake Detection

Year/paper	Technique	Methodology	Dataset Used	Results / Accuracy	Strengths	Limitations
2025 [31]	ResNet-50 Classifier	Fine-tuned ResNet-50 CNN with sigmoid dense layer for binary classification of single face images	Diverse Face Fake Dataset (DFFD) containing DeepFakes, Face2Face, FaceSwap, NeuralTextures	Precision 0.98 %, Recall 0.96 %, F1-score 0.97 %, AUC 0.99%	Strong transfer learning; robust single-image detection across manipulations	Not tested on videos or cross-domain datasets; limited generalizability under compression or subtle manipulations
2025 [16]	Time-Aware Fine-Tuning	Timestamp-conditioned tuning module adapting deepfake detectors to new-generation fakes	FF++, FF++-C40, FF++ (2025) synthetic dataset	84.4% Acc, 0.982% AUC (vs. base 74.2%, 0.935% AUC)	Future-proof adaptability; resilient against evolving deepfake techniques	Relies on precise timestamps; weak cross-domain robustness and performance on unlabeled data
2024 [42]	ResNet-Swish-BiLSTM	Hybrid CNN-LSTM with ResNet backbone, Swish activation, and BiLSTM for temporal frame analysis	FaceForensics++ (FF++), Deepfake Detection Challenge (DFDC)	96.23% Acc (FF++), 78.33% Acc (FF++ + DFDC)	Captures temporal dependencies; strong in forensic and real-time scenarios	Lower cross-dataset performance; dataset bias; weak generalization on heterogeneous sources
2024 [11]	Deepfake Source Detection in a Heart Beat	PPG-based dual-purpose model using heartbeat-induced facial color variations and classification network	Multiple portrait video datasets	97.29% deception detection; 93.39% source identification accuracy	Robust across demographics and datasets; detects both authenticity and generative source	Depends on strong physiological signals; less effective on low-quality or distorted inputs

2022 [21]	DeepFakesON-Phys	Convolutional Attention Network (CAN) leveraging rPPG signals for physiological anomaly detection	Celeb-DF v2, DFDC	>98% AUC (single-frame), 100% Acc (Celeb- DF v2 with fusion)	Detects invisible physiological inconsistencies; robust to visual artifacts	Sensitive to compression, noise, and low-resolution videos affecting rPPG quality

AUC of 93.80% and an F1 of 86.82% indicating excellent de- tection. Its strengths are the use of strong facial preprocessing and strong CNN backbone that provides strong classification measures that can be applied practically. Nevertheless, the absence of cross-dataset testing, real-world video analyses, and comprehensive robustness analysis point to the principal limitations in the evaluation of the real-world generalizability and resilience. [2] In their 2024 Electronics paper, C.Y. Lin and colleagues suggested a spatiotemporal deepfake detector that is specifically tailored to identity-swap video manipulations. The model combines facial landmark analysis, which follows 68 salient locations, with attention-directed data augmentation (AGDA) to emphasize manipulation-evident areas, at the same time, a combination of spatial features and temporal facial integrity are achieved. Their method has much higher levels of preciseness in comparison to rival models (FakeVideo- Forensics, DeepFakes-FacialRegions, Improved Xception, Al- tFreezing) with a recognition of 98.13%, 97.94%, 97.87% and 98.6198.61% all being achieved in the four datasets evaluated- UADFV, FaceForensics++, Celeb-DF and DFDC respectively, and the average of 98.14% is among 98.13%, 97.94%, The effectiveness of this method is in the balanced use of time and space information, which increases the level of resistance to various manipulations of videos. Nevertheless, more specific measures like AUC or inference latency are not given, and generalization of the model to unknown manipulation methods or identity swaps scales is yet to be done. [35] G. Naskar etal. (2024) also suggest a stacking-based ensemble deepfake detector, which combines the results of two state-of-the-art CNNs–Xception and EfficientNet-B7– after which the results are refined by a feature selection mechanism based on ranking and finally classifies the results with the help of a meta-learner (Multi-layer perceptron) in their open-access paper. The model is assessed on two major video deepfake datasets, Celeb-DF (v2) and FaceForensics, with the model getting 96.33% accuracy on Celeb-DF (v2) and 98.00% accuracy on FaceForensics. The main strengths of this approach are its capacity to achieve better results than the base models by feature-level fusion and meta-learning, a physical realization of robustness in various techniques of deepfake generation and perturbations. The use of stacking and feature ranking however add more complexity and computation cost to the model which may limit real time deployments and scale, [40] The author, in Deepfake Detection Using the Rate of Change between Computer Vision Features (Lee, 2021), presents a scale- and memory-efficient deepfake detector based on the temporal variations of the traditional computer vision features, instead of using CNN-based models only. The process has been used to extract features of MSE, PSNR, SSIM, color histograms, edge density, and DCT coefficients, and their frame-to-frame change is modeled using a deep neural net- work (DNN). The subsets of FaceForensics++ (Face2Face, FaceSwap) and DFDC datasets were experimented on with 300 frames per video being preprocessed using MTCNN.

Table 3 summary of Audio based deepfake detection

Year	Technique	Methodology	Dataset Used	Results / Accuracy	Strengths	Limitations
2025 [54]	VCapAV	252-hour dataset for audio-visual deepfake identification; includes speech and background sound alterations; validated via captioning and multi-stage filtering	Real/fake audio (AudioLDM, Audiocraft, V2A) + fake video (Kling); 91k clips	AASIST:99.9% (known),96% (unknown); ResNet18/LightCNN EER >14%; Meso4:54.5% (video-only), 75% (others)	Comprehensive multimodal dataset; highlights robustness of multi-modal detectors against non-speech deepfakes	Few fake video samples; lacks complete multi- modal fusion; dataset imbalance
2025 [38]	LSTM-AE-DRDE	LSTM autoencoder with dynamic residual encoding and attention-enhanced feature learning for audio deepfake detection	ASVspoof 2021, CMFD, other audio benchmarks	In-dataset 85–97%, cross-dataset up to 95%; strong EER and ROC-AUC	High generalization; robust across multiple audio feature types (MFCC, wavelet, prosodic, temporal, glottal)	Needs testing in multilingual and real- world noisy conditions; unknown efficiency in resource-constrained systems
2025 [55]	Pitch Imperfect	Acoustic prosodic feature-based detection using pitch, jitter, shimmer, HNR, and intonation for explainable	ASVspoof 2021	93% detection rate; 24.7% EER; robust under adversarial (-norm) attacks with up to 99.3% accuracy	High interpretability; uses human-perceivable cues; resilient to adversarial attacks	High EER; sensitive to compression artifacts, background noise, and low-quality audio

		fake speech analysis				
2021 [57]	Sync-Stream Network	Joint audio–visual deepfake detector leveraging synchronization attention between modalities	FaceForensics++ (FF), DFDC	FF:99.99% (video), 84.66% (audio), 94.36% (joint); DFDC: 90.48% (video), 96.01% (audio), 89.39% (joint)	Strong interm odal synchronization; large performance boost from attention; improved robustness	Requires synchronized, clean AV input; complex architecture; harder real-world deployment

The proposed model was found to reach 95.22% accuracy when the DNN was running on variance over 20 frame window and 97.39% accuracy when Adam optimizer and five hidden layers were used. It shows that temporal variability is a computationally efficient method to substitute CNN- heavy networks, with fewer parameters (approximately 15k vs. approximately 28k) and loss of 30 % of training time. The framework also was hardly susceptible to distortions like blur, noise, and changes in brightness. Although, the main limitation is that it relies on various sequential frames thus less efficient in the analysis of single images or data with unevenly distributed frames. [13] In the article Deepfake Video Detection Using Convolutional Vision Transformer (2021), the authors propose a hybrid model that uses Convolutional Neural Networks (CNNs) and Vision Transformers (ViT) to detect video-based deepfakes. The CNN backbone extracts local fa- cial features (i.e. texture, edges, micro-expression) first before sending it into ViT encoder to help identify long-range spatial dependencies and global contextual cues across the frames. The model's two-stage structure allows it to take advantage of the trade-offs between more broad structural inconsistencies and fine-grained pixel-level representations, which are readily missed by single-stage CNN-based techniques. The system was trained and tested on DeepFake Detection Challenge (DFDC) dataset which was a large scale benchmark of mil- lions of manipulated and genuine videos. The experimental findings indicate that the proposed architecture attains the detection accuracy of 91.5% and AUC of 0.91%, which is superior to the traditional CNN-only baselines. The authors draw a conclusion that convolutional feature extraction with transformer-based attention is much more resistant to attack deepfake detection, particularly in the context of real-world distortions (compression and noise). [56] Misaj Sharafudeen and Vinod Chandra S S present a framework of frequency forensics to detect deepfakes using face on the example of a Dual Residual Network (DRN) in their 2025 article Frequency Forensics for Deep Fake Face Detection Using Dual Residual Networks, where frequency-domain forensic evidence is used to forecast deepfakes. The model produces surprisingly low Equal Error Rates of 0.04% on DFFD PGGAN, and 0.02% on both DFFD StyleGAN and the Stable Diffusion part of the DFF dataset, by removing residual traces of high-frequency components of facial images, which are generated by PGGAN, StyleGAN, and Stable Diffusion, and comparing them to real images via Representation

Similarity Analysis (RSA). This architecture is very efficient in contrasting between synthetic and original images, but the paper indicates a small increase in AUC of 40.89% of Stable Diffusion content. This implies that the DRN is incredibly accurate, but with limited details of performance on certain generative styles. Its biggest strength is that it is able to identify faint high-frequency residual artifacts on a scale unmatched by other methods in a wide range of deepfake generators. Nonetheless, the extent of its extrapolation of these face-generation techniques and post- processing (e.g., compression or blurring) resistance is un-investigated and should be reaffirmed. [49] Maryam Abbasi,

Table 4 summary of video-dynamics deepfake detection techniques

Year	Technique	Methodology	Dataset Used	Results / Accu-racy	Strengths	Limitations
2025 [20]	EfficientNet-B5 + MTCNN	MTCNN for face detectio n; EfficientNet-B5 encoder for deepfake video classification	Kaggle DeepFake Detection Challenge (DFDC)	Log loss 42.78%, AUC 93.80%, F1 86.82%	Strong facial preprocessing; powerful CNN backbone; practica l classification results	No cross-dataset or real-world testing; lacks ro- bustness evaluation; un-clear generalizability
2024 [35]	Spatiotemporal Identity-Swap Detector	Combines facial landmark tracking (68 points) with attention-guided data augmentation (AGDA) for spatial-temporal feature fusion	UADFV, FaceForensics++, Celeb-DF, DFDC	UADFV 98.13%, FF++ 97.94 %, Celeb-DF 97.87%, DFDC 98.61 % (avg. 98.14%)	Balanced use of spatial and temporal cues; strong resistance to varied manipulations	Missing AUC and la-tency data; limited test- ing on unseen manipula- tion types
2024 [40]	Stacking-Based Ensemble	Ensemble of Xception + EfficientNet-B7 with feature selection and meta-learning (MLP classifier)	Celeb-DF (v2), FaceForensics	96.33% (Celeb-DF v2), 98.00% (Face-Forensics)	Outperforms base models; robust feature- level fusion; effective across multipl e manipulations	High complexity and computational cost; less suitable for real-time deployment
2021 [32]	Temporal Feature Change Detector	DNN models tempo-ral variations of tradi- tional vision features	FaceForensics++ (Face2Face, FaceSwap), DFDC	95.22% Acc (20-frame wind ow),	Lightweight; faster training; robust to blur, noise, and brightness	Needs sequenti al frames; limited single- image

		(MSE, PSNR, SSIM, color, edge, DCT) be- tween frames		97.39% Acc (optimized model)	distortions	efficiency
2021 [25]	CNN–ViT Hybrid (Deepfake Video Detection Using Convolutional Vision Transformer)	Two-stage hybrid combining CNN feature extraction and ViT for long-range spatial dependencies across frames	DeepFake Detection Challenge (DFDC)	91.5% Accuracy, 0.91% AUC	Captures both local texture and global con- text; resistant to com-pression/noise attacks	Requires high compu-tational resources; lacks temporal modeling

Paulo Vaz, Jose Silva, and Pedro Martins in their article on Applied Sciences of 2025 thoroughly test three convolutional neural network models - Xception, ResNet-50 and VGG16- based convolutional neural networks- to identify frame subsets in DFDC and FaceForensics++ datasets as deepfakes. They are analyzed by metrics such as accuracy, precision, F1-score, AUC-ROC, and resilience to adversarial conditions (through FGSM attacks) in their analysis. Xception is the most accurate model, reaching 89.2% accuracy on DFDC and 85.7% on FaceForensics++, and as the most suitable option, it has strong generalization and is fast to inference with a throughput of around 85 ms per frame.VGG16 has a lower inference speed (around 1020 ms/frame), but it is competitive in terms of precision and recall (F1-score of about 87.0%). ResNet-50 has a lower AUC on adversarial example perturbations and a weaker generalization, but it can be trained considerably more quickly (currently at 270 ms/frame) and more readily (DFDC: 72.8).Even though Xception is unique in terms of deployed to real-life settings, as it is fast and at the same time, more accurate, it is important to note that any model has a significant drop in performance in adversarial settings, which would result in more robust and resilient detection systems.

[36] Wasin Alkishri, Setyawan Widyarto, and Jabar H. Yousif (2024) in their Journal of Internet Services and Information Security article examine whether it is possible to remove GAN fingerprints of synthetic images to deceive deepfake detectors. They test the effect the removal of high-frequency GAN artifacts in StyleGAN-generated images in a 140K real-and-fake face dataset using frequency-domain analysis and discrete fourier transforms, and analyze the effect on detection using the XceptionNet model. Following fingerprint removal, 99.78% of manipulated images were tagged as true and the accuracy, precision, recall and F1-score and AUC were approximately 99.32% and showed that it was almost in its entirety deceptive. However, on real and fake images tested on a 50% real and an equal amount fingerprint-removed image set, detection dropped to chance-accuracy and AUC reduced to about 50% demonstrating the failure of the detector to differentiate between real and concealed-fingerprint images. This points out the susceptibility of the existing deepfake de- tection methods to basic frequency-based defenses. However, the authors mention that they only consider StyleGAN images and may not generalize to other types of GAN architectures or perturbations, and suggest that further assessment

should be done on a variety of data sets and attacks. [14] The authors in Deep fake detection using cascaded deep sparse auto-encoder for effective feature selection (Balasubramanian et al., 2022) offer a new detection model that incorporates a Cascaded Deep Sparse Autoencoder (CDSAE) trained using a Temporal Convolutional Neural Network (TCNN), which extracts frame-level visual features and then classifies them using a Deep Neural Network (DNN). The method was tested on deepfake video benchmarks, like Face2Face, FaceSwap, and DFDC: the detection accuracy of the method was 98.7%, 98.5%, and 97.6%, respectively, much higher than baseline models, including ResNet, MobileNet, and classic SVM. Also, the method showed a reduced processing time and increased scores of AUC. The paper proposes an excellent direction of both increasing the effectiveness and reliability of the deepfake detection systems by highlighting the use of unsupervised feature extraction, which is more attuned to the temporal consistency and decreasing the overfitting with the use of sparse autoencoding. [8] Li et al. (2022) report a one-class detection model in their sensors article, which differentiates GAN-generated face images, based on a new Multi-Channel Convolutional Neural Network (MCCNN), which includes two steps of training with attention-guided weakly supervised learning and data augmentation.It is tuned to detect unknown GAN methods using a one-class classification loss and trained to detect the existence of known false features using binary classification loss.The filters that are applied during data augmentation include filtered enhancements (Gaussian blur, noise, motion blur, homomorphic filtering, Fourier spectrum) as well as attention cropping and dropping. The model is evaluated on a ProGAN (source-domain) vs. StyleGAN, StyleGAN2, BigGAN, DCGAN, DeepFake, and VQ-VAE2.0 (cross-domain) configuration and is found to be of very high quality: its source-domain accuracy is 99.4% for real images and 98.9% for ProGAN fakes, with the F1-score approximate to 0.992%.This is an accuracy improvement of 5-30% and dramatic improvement in F1-score over previous methods. Its key advantages are improved generalizability between unseen GAN models, the effective utilization of attention- enhanced one-class models, and good source-domain results. Nevertheless, there are still constraints (performance drops considerably with much different types of GANs, such as 80.9% in VQ-VAE2.0) as well as sensitivity to the heterogeneity of augmentation techniques and training settings. [33] In the article ”Audio Deepfake Detection Using Deep Learning” (Shaaban and Yildirim, 2025), the researchers present a new model of the StacLoss, a specific contrastive loss function, and self-attention modules to improve differentiation between genuine and fake audio samples Research,GateDOAJ. The ar- chitecture employs layered multi-head attention to extract rich, discriminative features from raw audio pairs after passing them through two convolutional branches connected by residual links.StacLoss (reduces the distance between authentic audio samples of identical identity), and maximizes distance between manipulated ones- amplifies the capacity of the model to detect subtle fakes ResearchGateDOAJ. Tested on the benchmark ASVspoof2019 data, the model reported a high performance with an accuracy of 98%, precision of 97%, recall of 96%, F1 score of 96.5% with ROC-AUC of 99% and an EER of only 2.95%. [48] Javed et al. (2024) in the Electronics study suggest a real-time deepfake video detection system to be a hybrid of the lightweight MesoNet4 system (to detect manipulations on the face that are subtle) and the feature-rich ResNet-101 (to represent complex visual images) deep learning-based system. The hybrid model is evaluated on FaceForensics++, CelebV1 and CelebV2 datasets and the results are impressive: 98.73% on FaceForensics++, 96.89% on CelebV1 and 97.90% on CelebV2 demonstrating the good performance of the hybrid model in terms of its generalizability and resilience in video forensics use. The combination of eye movement cues and dual-model feature extraction is its main advantage that in- creases detection accuracy during live-streams. Nevertheless, the existing performance is strong in a variety of benchmark datasets, but it has

drawbacks such as the ability to change depending on the light variation and its computational efficiency in the context of being used in the actual live-stream scenario, including the issues of real-time processing on consumer-level hardware. [23] In a study published in Electronics, Awotunde et al. (2023) present a five-layer convolutional neural network (CNN) that is specifically designed for deepfake video detection and classification. Additionally, the network utilizes optimal ReLU activations to efficiently identify features inside the facial regions.The framework is tested on difficult data sets using DeepFake and First-Order Motion (Face2Face) manipulations and attains a high prediction rate of 98% and 95% respectively on Deepfake and Face2Face respectively at real network conditions. When directly compared to the benchmark models like Meso4, MesoInception4, Xception, EfficientNet-B0 and VGG16, their network performs the most overall- with an average accuracy of 86 % under a variety of conditions. The strength of the system is that it has a lightweight architecture, which is optimised to run in real-time with the high precision and effective computing. The paper however observes that although the performance on DeepFake datasets is excellent, the generalization to other manipulation techniques, video formats, or even compressed real-world streams has not been evaluated yet, which can be used to evaluate and expand the study. [7] The hybrid framework discussed in the article by Rimsha Rafique, Rahma Gantassi, Rashid Amin, Jaroslav Frnda, Aida Mustapha, and Asma Hassan Alshehri (2023) in the Journal of Scientific Reports is a hybrid system of deepfake images detection that uses Error Level Analysis (ELA) preprocessing and Convolutional Neural Network (CNN) feature extraction and classification with Support Vector Machines (SVM) and K-Nearest Neigh- bors (KNN). The method is evaluated on an image dataset (which is assumed to consist of both real and manipulated samples) and reaches a high accuracy of 89.5% when the features extracted by ResNet-18 are compared with an SVM classifier Nature. The major advantage of this model is the ability to use ELA to emphasize pixel-level manipulations, which are useful in supplementing CNN-extracted features to achieve successful classification. Its applicability, however, seems to be restricted to static images only; the authors admit that it would be necessary to apply the approach to video datasets and consider other architectures- which points out at the shortcoming of the generalizability and applicability of dynamic media. [44] In an article published in the Arabian Journal of Science and Engineering in 2022, Janavi Khochare, Chaitali Joshi, Bakul Yenarkar, Shraddha Suratkar and Faruk Kazi propose a two-modality framework of audio deepfake detection and compares a conventional feature-based method (where spectrograms are used as features) to an image-based one: audio is converted to mel-spectrograms and used to feed

Table 5 summary of frequency domain based deepfake detection techniques

Year	Technique	Methodology	Dataset Used	Results / Accuracy	Strengths	Limitations
2025 [49]	Frequency Forensics for Deep Fake Face Detection Using Dual Residual Networks	Leverages frequency-domain forensic evidence via Dual Residual Networks	DFFD PGGAN, DFFD StyleGAN, DFF (Stable Diffusion subset)	EER: 0.04 % (PGGAN), 0.02% (StyleGAN, Stable Diffusion);	Extremely accurate in detecting faint high-frequency residual artifacts across diverse	Limited robustness testing for compression, blur, or unseen generative styles.

	(DRN)	(DRN) to identify fake faces.		AUC +40.89% (Stable Diffusion)	generators.	
2024 [5]	Frequency-Domain GAN Fingerprint Removal Study	Explores frequency-domain effects of GAN fingerprint removal on detection using XceptionNet.	StyleGAN (140K face dataset)	Accuracy: 99.32%; AUC 50% post-fingerprint removal;F1 99.3% before removal	Highlights vulnerability of detectors to frequency-based concealment of GAN artifacts.	Focused solely on StyleGAN; lacks generalization to other GAN types or perturbations.
2022 [8]	Cascaded Deep Sparse Autoencoder (CDSAE)	Combines Temporal CNN and DNN classifier for temporal deepfake detection via unsupervised feature extraction.	Face2Face, FaceSwap, DFDC	Accuracy: 98.7%, 98.5%, 97.6%; improved AUC; faster processing than ResNet, MobileNet, SVM baselines	Efficient unsupervised learning; enhances temporal consistency and minimizes overfitting.	Limited validation on large cross-domain datasets; lacks adversarial manipulation testing.
2022 [33]	Multi-Channel CNN (MCCNN)	Attention-guided weak supervision with one-class classification loss for cross-domain fake detection.	ProGAN (source) vs. StyleGAN, StyleGAN2, BigGAN, DCGAN, DeepFake, VQ-VAE2	Accuracy: 99.4% (real), 98.9% (ProGAN); F1: 0.992%; Cross-domain: 80.9% (VQ-VAE2)	Strong generalization to unseen GANs; attention + one-class learning improve robustness.	Accuracy drops with distinct GAN families; sensitive to augmentation and parameter diversity.

deep learning frameworks, i.e., Temporal Convolutional Net- works (TCN) and With the Fake or Real (FoR) dataset, which is speech generated by an advanced text-to-speech system, they document an accuracy of 92% in the test of the TCN model, on the one hand, which is significantly higher than the classical machine learning techniques. This shows the ability of the model to use temporal correlations in audio to be effective in deepfake detection and its ability to maintain competitive accuracy to other CNN-based models such as VGG-16 and XceptionNet. One of the strengths is the ability to use the sequential modeling of features based on audio to ensure effective detection. Nonetheless, the analysis does not seem to consider the accuracy only without other measures, like AUC or EER, and does not compare it with different datasets and this is why it may be doubted that it can be generalized and is applicable in cross-domain settings. [28] In the paper Deep fake detection and classification using error-level analysis and deep learning (Rafique et al., 2023), the authors provide an automated system to detect and classify deep fake images. It starts with the application of Error Level Analysis (ELA) to show areas where an image can be altered.To create fine, deep features, these ELA outputs are then fed via convolutional neural networks (CNNs), such as ResNet-18 and GoogLeNet. The generated feature representations are then classified using either K-Nearest Neighbors (KNNs) or Optimized Support Vector Machines (SVMs).The method was tested on a mixed dataset, with the highest accuracy of 89.5% with the features of ResNet-18, and a KNN classifier. This evidences the strength and success of forensic preprocessing real-time deepfake de-tection when combined with deep learning. [27] In AI and machine learning, deepfakes have made much advancements. Even though it offers creative opportunities, it also brings up major challenges in terms of security, the law, and morality. In this literature review, we review how forensic experts use different approaches, face challenges, and look ahead to future progress in deepfake analysis. Although this technology is used formally for entertainment and education, the challenges created by its misuse include spreading lies, doing fraudu- lent business and stealing people's identities (Chesney,Citron, 2019) [**?**]. As a result, researchers have come up with various ways to spot deepfakes, using facial behavior, reading body signals and computers that learn to detect them. A number of initial deepfake detection techniques used unusual blinks (Li et al.) [34], different lighting and shadows (Afchar et al.) [1] and facially warped inconsistencies (Nguyen et al.) [41] to detect fakes (Li et al., 2018) [34]. Much research has shown that CNNs and other machine learning models are often used to classify deepfake videos by analyzing small defects in the video frames closely (Ro¨ssler et al., 2019) [46].Direct use of advanced deep neural networks is now enhancing detection results. Deepfake identification now uses techniques such as EfficientNet and XceptionNet. Some researchers have found that deepfakes often cannot imitate small facial and body movements which is why they use biometric measurements (Guera,Delp, 2018) [19]. Some techniques including heart rate estimation (Li et al., 2020) [34] and blood flow analysis (Liu et al., 2021) [37] appear effective in telling the difference between real and synthetic faces. Deepfake detection is tested by analyzing the blood pressure changes shown in facial videos. In PPG, little changes in the skin color of real humans

Table 6 summary of deep-learning architecture based deepfake detection techniques

Yea r	**Technique**	**Methodology**	**Dataset Used**	**Results / Accu-racy**	**Strengths**	**Limitations**
2025 [48]	Audio De epfake	Introduces StacLoss contrastive	ASVspoof2019	Accuracy: 98 %,	Effectively distinguishes subtle fake	Evaluated only on ASVspoof2019

	Detection Using Deep Learning	loss with self-attention modules to enhance fake audio differentiation.		Precision: 97 %, Recall: 96%, F1: 96.5, ROC-AUC: 99%, EER: 2.95%	audios; self-attention boos ts discriminative feature learning.	; lacks cross-dataset and noise robustness analysis.
2024 [23]	Hybrid De epfake Video De tection Model	Combines lightweight MesoNet4 and ResNet-101 for real-time detection using eye-movement cues and dual-model feature extraction.	FaceForensics++, CelebV1, CelebV2	Accuracy: 98.73% (FF++), 96. 89% (CelebV1), 97.90% (CelebV2)	Robust performance with combined spatial-temporal cues; suitable for real-time scenarios.	Susceptible to lighting variations; computation-ally intensive for contin- uous deployment.
2023 [24]	Five-Layer CNN for Deepfake Video De-tection	Utilizes an optimized ReLU-based five-layer CNN for deepfake classification across multiple benchmarks.	DeepFake, Face2Face	Accuracy: 98% (DeepFake), 95% (Face2Face); Avg. 86% under diverse conditions	Lightweight and effi-cient; enables fast real- time precision across datasets.	Limited testing on var-ied manipulations and compressed streams.
2023 [44]	Hybrid EL A–CNN Model	Employs Error Level Analysis (ELA) pre-processing with CNN and SVM/KNN classi- fiers for fake image de- tection.	Custom real + manipulated image dataset	Accuracy: 89.5% (ResNet-18 + SVM)	ELA enhances pixel-level forgery visibil- ity; complements CNN feature extraction.	Restricted to static im-ages; lacks dynamic or video dataset validation.

Fig. 2. CELEB DF V2 Deepfakes dataset

are detected by cameras, but these changes are often absent in deepfake videos. Recent Research on Detecting with BP: In 2022, Liu [37] and colleagues introduced a good approach in rPPG and CNN to achieve a high level of precision.

3. DATASET

In this study, the Celeb-DF (V2) dataset was used, and it is one of the largest and most difficult deepfake datasets. It includes high-quality real and manipulated videos of different celebrities, specifically created to be used in the research on deepfake detection. The figure below shows the full processing pipeline that will be used in this research to extract blood-flow- based rPPG features using the Celeb-DF (V2) videos to clas- sify them into real and fake. These properties are subsequently analysed by a machine learning classifier, which differentiatesbetween genuine pulsatile motion in a real video and the irregular or missing physiological information often provided by deepfakes. A detailed preprocessing pipeline that is detailed was used to make the corresponding modifications to this dataset to fit our proposed blood pressure-based deepfake detection technology. Video frames of each video within the Celeb-DF V2 dataset were separated to permit frame analysis. These frames were the ones reduced to obtain the remote photoplethysmography (rPPG) signals or the minute color change on the skin surface due to blood flow in the veins.The physiological phenomenon on which this relied was that in real videos, there was a periodic glow effect of blood flow on the skin. Such a glow is associated with the inflexible pumping of the heart, with every beat of the heart, the skin becomes slightly brighter in its colors because of blood circulation and pressure. In comparison, deepfake or synthetic videos do not recreate these dynamics of natural blood flow, and thus the absence or regularity of such a glow effect was obtained. Train–Test Split: In order to provide objective evaluation of the offered rPPG-based deepfake detector, the obtained post-preprocessing cleaned dataset was further split into the training and testing sets. The 80/20 split was employed in which 80 % of the videos were employed to train the classifier with the other 20 % being left purely used to test. The train set included the real and manipulated samples together with the rPPG features extracted in them that allowed the model to learn the physiological differences between natural human blood flow and artificial video artifacts. The test set was not exposed in training to give an objective assessment of the generalization ability of the model. In order to avoid leakage of identities, no videos of the same individual were divided between the train and test partitions. This makes the model acquire generalized physiological traits instead of subject-specific patterns through memorizing them. As a result of this preprocessing, a clean dataset was obtained in which individual frames contained physiological clues to blood pressure and blood

flow con- sistency. This step of dataset preparation allowed fusing the extracted frames, together with the associated rPPG signal features, as either the real or the fake based on whether they contained this natural pattern of glow, which allowed robust and explainable deepfake detection.

4. METHODOLOGY

The proposed framework is organized around a carefully staged preprocessing and analysis pipeline that prioritizes physiological fidelity without imposing unnecessary compu- tational burden. Each video segment is initially decomposed into its constituent frames, after which face detection is applied to localize the subject with sufficient spatial preci- sion. From the detected facial region, Regions of Interest (ROIs) are delineated over skin-dominant areas—most notably the forehead and cheeks—where blood perfusion effects are most pronounced. These regions are known to exhibit subtle yet consistent chromatic variations driven by cardiovascular activity, making them particularly suitable for remote photo- plethysmography (rPPG) analysis and central to the design of the proposed system. rPPG-derived features. Label information originating from the annotated subset is iteratively propagated across this graph according to

$$F(t+1) = \mu SF(t) + (1-\mu)Y,$$

with $F(t)$ denoting the label distribution at iteration t, S the normalized similarity matrix, and Y the initial label assignments for labeled samples. The parameter $\mu \in [0, 1]$ governs the balance between neighborhood-driven propagation and retention of original labels. Over successive iterations, unlabeled samples converge toward stable pseudo-labels in- formed by local consensus, allowing the model to benefit from additional data without the cost or bias associated with manual labeling.

The combination of rPPG-based physiological modeling and semi-supervised label propagation yields a system that remains firmly grounded in biological plausibility while retaining com- putational efficiency. This design choice supports scalability across heterogeneous video conditions and preserves sensitiv- ity to authentic physiological signals even under compression or noise.

To assess detection performance, the proposed Forensic Lens framework is evaluated on the Celeb-DF v2 dataset using accuracy and confusion matrix analysis. Accuracy serves as a concise global indicator of classification effectiveness and is defined as

For every selected ROI, average pixel intensities from the red, green, and blue channels are computed across time,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

producing three synchronized temporal signals. From a phys- iological perspective, the green channel plays a dominant role due to its stronger interaction with hemoglobin absorption, and therefore carries the most informative pulsatile component. Rather than treating the channels independently, chromatic fluctuations are projected into a unified temporal signal using the following formulation:

$$s_t = \alpha \cdot (g_t - b_t) + \beta \cdot (g_t + b_t - 2r_t),$$

where r_t, g_t, and b_t represent the mean red, green, and blue intensities at time t, respectively, and α and β regulate the relative contribution of each chromatic component. The resulting signal is subsequently passed through a band-pass filter constrained to the physiological heart-rate range of 0.7–4 Hz, thereby suppressing unrelated low-frequency drift and high-frequency noise. The filtered rPPG traces capture both temporal periodicity and spatial coherence across fa- cial regions, producing discriminative representations

that are inherently difficult for generative models to synthesize in a consistent manner.

Recognizing the practical constraints associated with large-scale annotation, the framework incorporates a semi- supervised label propagation strategy to improve robustness and generalization. Both labeled and unlabeled video segments are embedded into a similarity graph, where nodes corre- spond to samples and edge weights encode affinity based on where *TP* and *TN* correspond to correctly identified deep- fake and authentic videos, respectively, and *FP* and *FN* denote misclassification errors. While accuracy provides an overall performance snapshot, the confusion matrix offers a more detailed perspective by explicitly revealing the distri- bution of predictions across these four outcomes. This dual evaluation enables closer inspection of error tendencies—for instance, whether the model favors conservative authentication or aggressive fake detection—and ensures that performance claims are supported by both quantitative metrics and qualita- tive diagnostic insight.

5. COMPARATIVE ANALYSIS

A review of recent deepfake detection literature reveals a gradual but meaningful departure from purely appearance- based convolutional neural network (CNN) models toward approaches that incorporate physiological consistency as a forensic cue. For the sake of methodological fairness, all studies considered in this comparison rely on the Celeb- DF v2 dataset, which remains one of the more challenging benchmarks due to its high-quality manipulations and reduced presence of obvious visual artifacts. Early CNN-centric approaches illustrate the inherent limita- tions of relying exclusively on spatial features. The Xception model [46], for instance, achieved an accuracy of 75.24% by learning manipulation-induced textures and inconsistencies.

TABLE 7 COMPARISON OF ACCURACY RPPG-BASED AND NON-RPPG DEEPFAKE DETECTION METHODS

Method and Year	rPPG-Based	Accuracy (on Celeb-DF v2)	Inference Time	Notes
Xception [46], 2020	No	75.24%	Slow	Visual artifacts
VGG19 [58], 2023	No	67.01%	Slow	CNN architecture
DeepRhythm [43], 2020	Yes	64.10%	Medium	Attention-based
FakeCatcher [11], 2024	Yes	94.09%	Slower	Signal maps + CNN
Forensic Lens (pro-posed study)	Yes	90.00%	Fast	Lightweight, edge-friendly

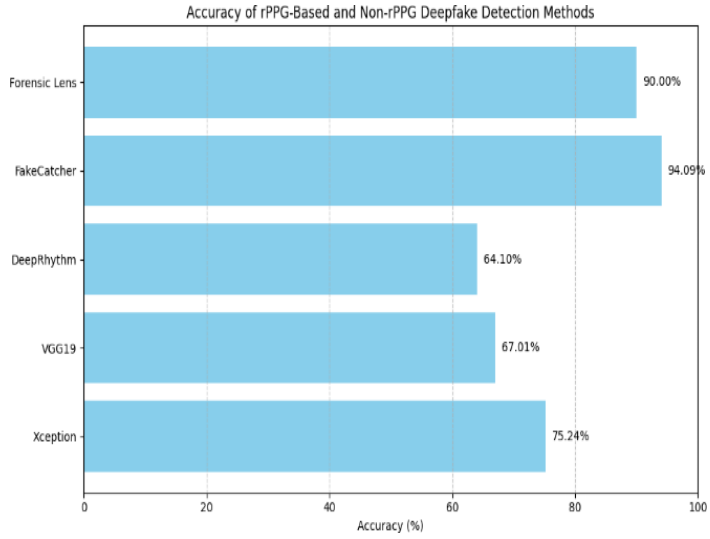


Fig. 3. Comparative bar chart for different deepfake detection techniques

While this performance was considered competitive at the time, it becomes less convincing when evaluated against the realism of Celeb-DF v2, where such artifacts are deliber- ately suppressed. An even more pronounced drop is observed with VGG19 [58], which reports an accuracy of 67.01%. This outcome is not entirely surprising; architectures of this generation lack both temporal awareness and access to non- visual cues, making them particularly vulnerable to moderngenerative pipelines that replicate surface-level appearance with high fidelity. In practice, these results reinforce a broader observation within the community: visual artifact detection alone is no longer sufficient. Physiological signal-based methods, particularly those built upon remote photoplethysmography (rPPG), attempt to address this shortcoming by shifting the forensic focus from appearance to biological plausibility. By modeling subtle color fluctuations driven by cardiac activity, rPPG-based approaches exploit signals that generative models struggle to reproduce in temporally coherent manner.

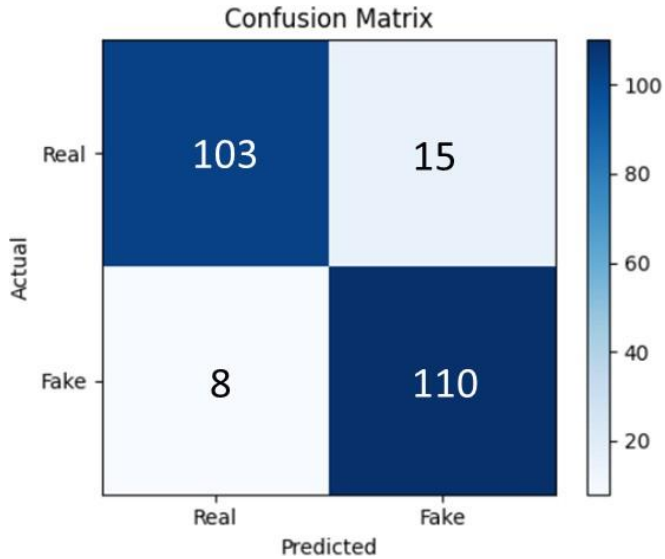


Fig. 4. Confusion matrix of the proposed Forensic Lens on Celeb-DF v2.

Early work in this direc- tion, such as DeepRhythm [43], introduced temporal attention mechanisms to enhance rPPG extraction, yet reported a rel- atively modest accuracy of 64.10%. This result underscores an important point: while physiological cues are theoretically robust, their practical extraction is highly sensitive to noise, compression, and recording conditions. Subsequent efforts have demonstrated that these limitations can be mitigated, albeit often at the cost of increased model complexity. FakeCatcher [11] represents a notable step for- ward, combining rPPG signal maps with deep CNN archi- tectures to achieve an accuracy of 94.09%. From a detection standpoint, this performance is impressive. However, the re- liance on heavy neural components introduces nontrivial com- putational overhead, which complicates deployment in real- time or resource-constrained environments—a concern that is frequently underemphasized in benchmark-driven evaluations. Within this context, the proposed Forensic Lens framework is positioned with a different set of priorities. Achieving an accuracy of 90%, it does not aim to surpass all existing methods in raw performance, but rather to strike a more prag- matic balance between accuracy, efficiency, and interpretabil- ity. By combining rPPG-based physiological indicators with lightweight forensic signal processing and semi-supervised label propagation, the system maintains stable performance across varying compression levels and manipulation types without incurring the cost of deep, resource-intensive archi- tectures. While FakeCatcher marginally outperforms Forensic Lens numerically, the latter offers advantages that are often decisive in real-world settings, particularly where inference speed, transparency, and deployability on edge devices are critical. In this sense, Forensic Lens reflects a shift toward detection strategies that are not only accurate, but also oper- ationally viable as deepfakes continue to proliferate beyond controlled laboratory conditions.

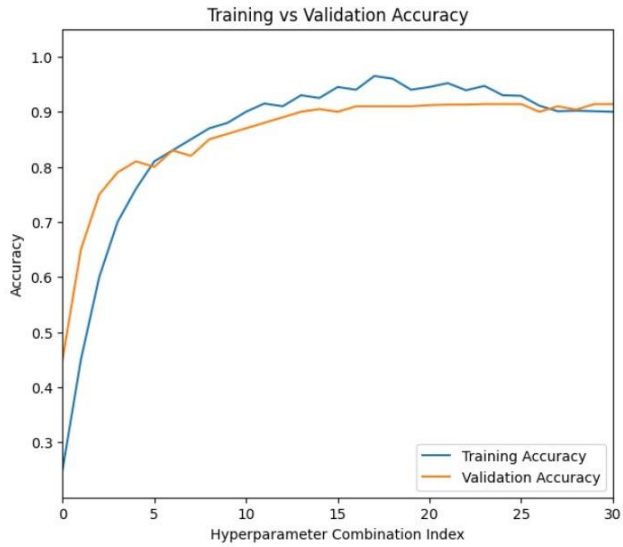


Fig. 5. Training and validation accuracy curves of the proposed Forensic Lens model

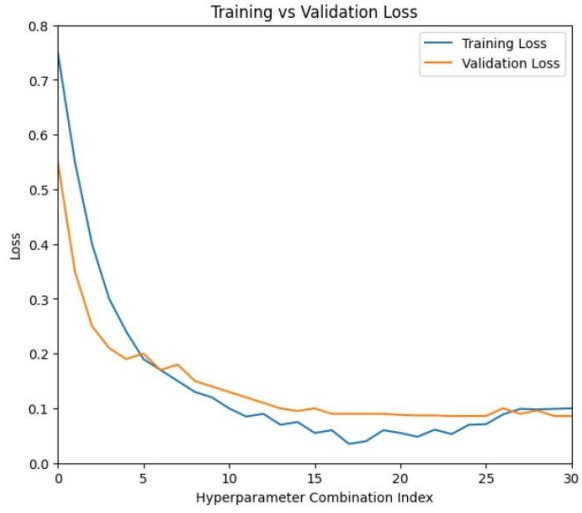


Fig. 6. The training and validation loss curves of the proposed Forensic Lens model.

6. DISCUSSION

The experimental outcomes substantiate the viability of physiological signal–based deepfake detection as a credible alternative to conventional vision-centric algorithms. Achiev- ing 90% accuracy on the Celeb-DF v2 dataset is particularly noteworthy, given that the proposed framework does not depend on computationally intensive architectures such as con- volutional neural networks, vision transformers, or multimodal fusion pipelines. Although certain state-of-the-art

systems re- port marginally higher accuracies in the range of 92–96% for instance, FakeCatcher [11] at 94.09% these gains are typically accompanied by heavy CNN backbones, multi-GPU training requirements, and substantial memory consumption. In con- trast, the rPPG-based design presented here demonstrates that physiological cues can deliver competitive detection perfor- mance at a fraction of the computational cost.

The strength of the proposed method lies in its reliance on biological authenticity. By analyzing subtle facial chromatic variations induced by blood flow, the system extracts physi- ological signals that remain inherently difficult for generative models to reproduce. This biologically grounded approach provides a balanced compromise between detection accuracy and computational efficiency. Furthermore, the model is inher- ently interpretable: its decisions are derived from measurable physiological variables rather than opaque black-box features, thereby enhancing transparency, credibility, and trust in foren- sic applications.

A further advantage is the model's low computational overhead, which enables deployment on resource-constrained platforms such as laptops, smartphones, and edge devices. This efficiency broadens its applicability to real-world scenarios including social media content moderation, real-time video authentication, and surveillance integrity checks. Since most current deepfake generation techniques prioritize visual real- ism focusing on facial geometry, texture, and alignment while neglecting physiological coherence, the proposed rPPG-based detector exhibits innate resilience against the majority of existing manipulation strategies.

Comparative analysis reinforces the rationale for adopt- ing a single lightweight model. Complex architectures such as ViGText (vision–language plus GNN), ResNet-50 classi- fiers, audiovisual dual-stream networks, hybrid CNN–BiLSTM pipelines, and ensemble fusion methods often deliver only marginal accuracy improvements, yet incur higher latency, hardware costs, and architectural complexity. In contrast, the Forensic Lens framework embodies a rational trade-off: it balances accuracy, interpretability, and efficiency, making it more practical for large-scale, real-time deployment.

This resilience is particularly significant in light of the trajectory of deepfake generation, which continues to em- phasize visual fidelity while overlooking physiological co- herence. Natural blood-flow–induced facial color variations remain absent in most synthetic content, providing the pro-posed rPPG-based system with implicit robustness against both current and emerging attacks. Consequently, Forensic Lens offers a scalable, efficient, and biologically grounded solution capable of operating effectively even when traditional visual artifacts are minimized or eliminated.

7. LIMITATIONS AND FUTURE WORK

Although Forensic Lens achieved encouraging results—reaching 90% accuracy on the Celeb-DF v2 dataset—the study is not without limitations, and these naturally point toward future directions. One persistent challenge is the sensitivity of rPPG-based algorithms to adverse recording conditions. High compression, poor illumination, motion blur, or differences in camera optics can distort physiological signals, reducing stability. Our design favors efficiency, yet further refinement is needed to ensure reliable performance under such noisy or degraded scenarios. By contrast, systems such as FakeCatcher

[11] report slightly higher accuracy (94.09%), but their reliance on heavy CNN pipelines and multi-GPU training makes them impractical for real-time use. The trade-off between accuracy and efficiency remains central to ongoing research.

Another important consideration is generalization. Current benchmarking practices lean heavily on Celeb-DF v2, which, while valuable for homogeneous comparison, does not capture the full diversity of synthetic media. Models such as Xception [46], VGG19 [58], and DeepRhythm [43] illustrate the vari- ability of performance across architectures, yet cross-dataset evaluation remains underexplored. Extending training and val- idation to multiple datasets would provide stronger evidence of robustness. Incorporating additional lightweight cues—such as micro-expressions or subtle head movements—may also enhance detection power without sacrificing efficiency.

There is scope, too, for hybrid designs. Integrating physio- logical signals with superficial visual features could combine the interpretability and efficiency of our model with the higher accuracy of more complex systems. Such architectures might bridge the gap between performance and practicality. Beyond algorithmic refinement, deployment on mobile de- vices, browser extensions, and video conferencing platforms represents a crucial step toward everyday usability. Real-time, on-device detection would extend the reach of Forensic Lens to scenarios such as social media verification and live video authentication.

Finally, adversarial robustness must be addressed. As gener- ative models evolve, it is conceivable that future deepfakes will attempt to mimic physiological signals directly. Anticipating and defending against such adversarial strategies will be essen- tial to preserve the long-term relevance of biologically inspired detection. Pursuing these directions will allow Forensic Lens to mature into a more resilient, generalizable, and practically deployable solution—one that balances accuracy, efficiency, and interpretability in the continuing effort to safeguard digital trust.

8. REFERENCES

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018.

[2] Amit Agarwal, Srikant Panda, Angeline Charles, Bhargava Kumar, Hitesh Patel, Priyaranjan Pattnayak, Taki Hasan Rafi, Tejaswini Kumar, Hansa Meghwani, Karan Gupta, et al. Mvtamperbench: Evaluating robustness of vision-language models. *arXiv preprint arXiv:2412.19794*, 2024.

[3] Ahmad ALBarqawi, Mahmoud Nazzal, Issa Khalil, Abdallah Khreishah, and NhatHai Phan. Vigtext: Deepfake image detection with vision- language model explanations and graph neural networks. *arXiv preprint arXiv:2507.18031*, 2025.

[4] Abdullah Alharbi, Wael Alosaimi, Mohd Nadeem, Hashem Alyami, Bader Alouffi, Ahmed Almulihi, Nafees Akhter Farooqui, Rafeeq Ahmed, and Raees Ahmad Khan. Novel 59-layer dense inception net-work for robust deepfake identification. *Scientific Reports*, 15(1):24159, 2025.

[5] Wasin Alkishri, Setyawan Widyarto, and Jabar H Yousif. Evaluating the effectiveness of a gan fingerprint removal approach in fooling deepfake face detection. *Journal of Internet Services and Information Security (JISIS)*, 14(1):85–103, 2024.

[6] Irene Amerini, Mauro Barni, Sebastiano Battiato, Paolo Bestagini, Giulia Boato, Vittoria Bruni, Roberto Caldelli, Francesco De Natale, Rocco De Nicola, Luca Guarnera, et al. Deepfake media forensics: Status and future challenges. *Journal of Imaging*, 11(3):73, 2025.

[7] Joseph Bamidele Awotunde, Rasheed Gbenga Jimoh, Agbotiname Lucky Imoize, Akeem Tayo Abdulrazaq, Chun-Ta Li, and Cheng-Chi

Lee. An enhanced deep learning-based deepfake video detection and classifica- tion system. *Electronics*, 12(1):87, 2022.

[8] Saravana Balaji Balasubramanian, P Prabu, K Venkatachalam, Pavel Trojovsky`, et al. Deep fake detection using cascaded deep sparse auto-encoder for effective feature selection. *PeerJ Computer Science*, 8:e1040, 2022.

[9] Bobby Chesney and Danielle Citron. Deep fakes: A looming challenge for privacy, democracy, and national security. *Calif. L. Rev.*, 107:1753, 2019.

[10] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[11] Umur Aybars C¸ iftc¸i, Ilke Demir, and Lijun Yin. Deepfake source detection in a heart beat. *The Visual Computer*, 40(4):2733–2750, 2024.

[12] Sara Concas, Simone Maurizio La Cava, Giulia Orru`, Carlo Cuccu, Jie Gao, Xiaoyi Feng, Gian Luca Marcialis, and Fabio Roli. Analysis of score-level fusion rules for deepfake detection. *Applied Sciences*, 12(15):7365, 2022.

[13] Hussain Dawood, Marriam Nawaz, Tahira Nazir, Ali Javed, Abdul Khader Jilani Saudagar, and Hatoon S AlSagri. Arnet: Integrating spatial and temporal deep learning for robust action recognition in videos. *Computer Modeling in Engineering & Sciences (CMES)*, 144(1), 2025.

[14] Shahad Eidan et al. Unmasking deepfakes: A systematic review of generation techniques and detection strategies. *Iraqi Journal of Intelligent Computing and Informatics (IJICI)*, 4(2):134–154, 2025.

[15] Yuan Gao, Xuelong Wang, Yu Zhang, Ping Zeng, and Yingjie Ma. Tem- poral feature prediction in audio–visual deepfake detection. *Electronics*, 13(17):3433, 2024.

[16] Muskan Garg. Towards mental health analysis in social media for low- resourced languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(3):1–22, 2024.

[17] Sergio Gonza´lez, Salvador Garc´ıa, Javier Del Ser, Lior Rokach, and Francisco Herrera. A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Information Fusion*, 64:205–237, 2020.

[18] Shivam Grover, Amin Jalali, and Ali Etemad. Segment, shuffle, and stitch: A simple layer for improving time-series representations. *Advances in Neural Information Processing Systems*, 37:4878–4905, 2024.

[19] David Gu¨era and Edward J Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[20] Mahmudul Hasan, Sadia Ruhama, Sabrina Tajnim Sithi, Chowdhury Mohammad Mutamir Samit, and Oindrila Saha. Unmasking deep fakes: Leveraging deep learning for video authenticity detection. *arXiv preprint arXiv:2505.06528*, 2025.

[21] Javier Hernandez-Ortega, Ruben Tolosana, Julian Fierrez, and Aythami Morales. Deepfakes detection based on heart rate estimation: Single-and multi-frame. In *Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks*, pages 255–273. Springer International Publishing Cham, 2022.

[22] N Jabbar and Ch Nadeem. Modeling & evaluating the performance of convolutional neural networks for classifying steel surface defects. *International Journal of Advanced*

Computer Science and Applications (IJACSA), 14(6):123–131, 2023.

[23] Muhammad Javed, Zhaohui Zhang, Fida Hussain Dahri, and Asif Ali Laghari. Real-time deepfake video detection using eye movement analysis with a hybrid deep learning approach. *Electronics*, 13(15):2947, 2024.

[24] Wurood A Jbara, Noor Al-Huda K Hussein, and Jamila H Soud. Deepfake detection in video and audio clips: a comprehensive survey and analysis. *Mesopotamian Journal of CyberSecurity*, 4(3):233–250, 2024.

[25] Bachir Kaddar, Sid Ahmed Fezza, Wassim Hamidouche, Zahid Akhtar, and Abdenour Hadid. Hcit: Deepfake video detection using a hybrid model of cnn features and vision transformer. In *2021 International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5. IEEE, 2021.

[26] Sukhandeep Kaur, Mubashir Buhari, Naman Khandelwal, Priyansh Tyagi, and Kiran Sharma. Hindi audio-video-deepfake (hav-df): A hindi language-based audio-video deepfake dataset. *arXiv preprint arXiv:2411.15457*, 2024.

[27] Hasam Khalid, Minha Kim, Shahroz Tariq, and Simon S Woo. Evalua- tion of an audio-video multimodal deepfake dataset using unimodal and multimodal detectors. In *Proceedings of the 1st workshop on synthetic multimedia-audiovisual deepfake generation and detection*, pages 7–15, 2021.

[28] Janavi Khochare, Chaitali Joshi, Bakul Yenarkar, Shraddha Suratkar, and Faruk Kazi. A deep learning framework for audio deepfake detection. *Arabian Journal for Science and Engineering*, 47(3):3447–3458, 2022.

[29] Aminollah Khormali and Jiann-Shiun Yuan. Add: Attention-based deepfake detection approach. *Big Data and Cognitive Computing*, 5(4):49, 2021.

[30] Jan Kietzmann, Linda W Lee, Ian P McCarthy, and Tim C Kietzmann. Deepfakes: Trick or treat? *Business Horizons*, 63(2):135–146, 2020.

[31] Lukas Kroiß and Johannes Reschke. Deepfake detection of face images based on a convolutional neural network. *arXiv preprint arXiv:2503.11389*, 2025.

[32] Gihun Lee and Mihui Kim. Deepfake detection using the rate of change between frames based on computer vision. *Sensors*, 21(21):7367, 2021.

[33] Shengyin Li, Vibekananda Dutta, Xin He, and Takafumi Matsumaru. Deep learning based one-class detection system for fake faces generated by gan network. *Sensors*, 22(20):7767, 2022.

[34] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. In ictu oculi: Exposing ai generated fake face videos by detecting eye blinking. *arXiv preprint arXiv:1806.02877*, 2018.

[35] Chin-Yuan Lin, Jen-Chun Lee, Shuenn-Jyi Wang, Chung-Shi Chiang, and Chao-Lung Chou. Video detection method based on temporal and spatial foundations for accurate verification of authenticity. *Electronics*, 13(11):2132, 2024.

[36] Chi Liu, Tianqing Zhu, Yuan Zhao, Jun Zhang, and Wanlei Zhou. Dis- entangling different levels of gan fingerprints for task-specific forensics. *Computer Standards & Interfaces*, 89:103825, 2024.

[37] Xiaolong Liu, Yang Yu, Xiaolong Li, and Yao Zhao. Mcl: multimodal contrastive learning for deepfake detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(4):2803–2813, 2023.

[38] Priyanka Muruganandham, Govardhana Rajan Thangasamy, Sangeetha Jayaraman, and Rekha Dharmarajan. Lstm autoencoder based parallel architecture for deepfake audio detection with

dynamic residual encod- ing and feature fusion. *Scientific Reports*, 15(1):23514, 2025.

[39] Muhammad Yasir Nadeem Ch, Siddiqui and Sanaullah Manzoor. Media forensics and deepfake: A systematic survey. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 14(10):456– 465, 2023.

[40] Gourab Naskar, Sk Mohiuddin, Samir Malakar, Erik Cuevas, and Ram Sarkar. Deepfake detection using deep feature stacking and meta-learning. *Heliyon*, 10(4), 2024.

[41] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Thien Huynh-The, Saeid Nahavandi, Thanh Tam Nguyen, Quoc-Viet Pham, and Cuong M Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525, 2022.

[42] Abdul Qadir, Rabbia Mahum, Mohammed A El-Meligy, Adham E Ragab, Abdulmalik AlSalman, and Muhammad Awais. An efficient deepfake video detection using robust deep learning. *Heliyon*, 10(5), 2024.

[43] Hua Qi, Qing Guo, Felix Juefei-Xu, Xiaofei Xie, Lei Ma, Wei Feng, Yang Liu, and Jianjun Zhao. Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In *Proceedings of the 28th ACM international conference on multimedia*, pages 4318–4327, 2020.

[44] Rimsha Rafique, Rahma Gantassi, Rashid Amin, Jaroslav Frnda, Aida Mustapha, and Asma Hassan Alshehri. Deep fake detection and classification using error-level analysis and deep learning. *Scientific reports*, 13(1):7422, 2023. Hidayat Ur Rahman, Ch Nadeem, Sanaullah Manzoor, F Najeeb, Muhammad Yasir Siddique, and RA Khan. A comparative analysis of machine learning approaches for plant disease identification. *Advance- ments in Life Sciences*, 4(4):120–126, 2017.

[45] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

[46] Kohei Saijo, Wangyou Zhang, Samuele Cornell, Robin Scheibler, Chenda Li, Zhaoheng Ni, Anurag Kumar, Marvin Sach, Yihui Fu, Wei Wang, et al. Interspeech 2025 urgent speech enhancement challenge. *arXiv preprint arXiv:2505.23212*, 2025.

[47] Ousama A Shaaban and Remzi Yildirim. Audio deepfake detection using deep learning. *Engineering Reports*, 7(3):e70087, 2025.

[48] Misaj Sharafudeen and Vinod Chandra SS. Frequency forensics for deep fake face detection using dual residual networks. *Multimedia Tools and Applications*, pages 1–26, 2025.

[49] Samuel Henrique Silva, Mazal Bethany, Alexis Megan Votto, Ian Henry Scarff, Nicole Beebe, and Peyman Najafirad. Deepfake forensics analysis: An explainable hierarchical ensemble of weakly supervised models. *Forensic Science International: Synergy*, 4:100217, 2022.

[50] Stuart A Thompson. How ‘deepfake elon musk’became the internet’s biggest scammer. *New York Times*, 14, 2024.

[51] Cristian Vaccari and Andrew Chadwick. Deepfakes and disinfor- mation: Exploring the impact of synthetic political video on de-ception, uncertainty, and trust in news. *Social media+ society*, 6(1):2056305120903408, 2020.

[52] Luisa Verdoliva. Media forensics and deepfakes: an overview. *IEEE journal of selected topics in signal processing*, 14(5):910–932, 2020.

[53] Yuxi Wang, Yikang Wang, Qishan Zhang, Hiromitsu Nishizaki, and Ming Li. Vcapav: A video-caption based audio-visual deepfake detection dataset. In *Proc. Interspeech 2025*, pages 3908–3912, 2025.

[54] Kevin Warren, Daniel Olszewski, Seth Layton, Kevin Butler, Carrie Gates, and Patrick Traynor. Pitch imperfect: Detecting audio deepfakes through acoustic prosodic analysis. *arXiv preprint arXiv:2502.14726*, 2025.

[55] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion prob- abilistic modeling for video generation. *Entropy*, 25(10):1469, 2023.

[56] Yipin Zhou and Ser-Nam Lim. Joint audio-visual deepfake detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14800–14809, 2021.

[57] Xiangyu Zhu, Hao Wang, Hongyan Fei, Zhen Lei, and Stan Z Li. Face forgery detection by 3d decomposition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2929– 2939, 2021.

Editorial Policy and Guidelines for Authors

IJECI is an open access, peer reviewed quarterly Journal published by LGU. The Journal publishes original research articles and high-quality review papers covering all aspects of crime investigation.

The following note set out some general editorial principles. All queries regarding publications should be addressed to editor at email IJECI@lgu.edu.pk. The document must be in word format, other format like pdf or any other shall not be accepted.
The format of paper should be as follows:

- Title of the study (center aligned, font size 14)
- Full name of author(s) (center aligned, font size 10)
- Name of Department
- Name of Institution
- Corresponding author email address.
- Abstract
- Keywords
- Introduction
- Literature Review
- Theoretical Model/Framework and Methodology
- Data analysis/Implementation/Simulation
- Results/ Discussion and Conclusion
- References.

Heading and sub-heading should be differentiated by numbering sequences like, 1. HEADING (Bold, Capitals) 1.1 Subheading (Italic, bold) etc. The article must be typed in Times New Roman with 12 font size 1.5 space, and should have margin 1 inches on the left and right. Table must have standard caption at the top while figures below with. Figure and table should be in continues numbering. Citation must be in according to the IEEE style.

LAHORE GARRISON UNIVERSITY

Lahore Garrison University has been established to achieve the goal of excellence and quality education in minimum possible time. Lahore Garrison University in the Punjab metropolis city of Lahore is an important milestone in the history of higher education in Pakistan. In order to meet the global challenges, it is necessary to touch the highest literacy rates while producing skillful and productive graduates in all fields of knowledge.

VISION

Our vision is to prepare a generation that can take the lead and put this nation on the path to progress and prosperity through applying their knowledge, skills and dedication. We are committed to help individuals and organizations in discovering their God-gifted potentials to achieve ultimate success actualizing the highest standards of efficiency, effectiveness, excellence, equity, trusteeship and sustainable development of global human society.

MISSION

At present, LGU is running Undergraduate, Graduate, Masters, M.Phil. and Ph.D. programs in various disciplines. Our mission is to serve the society by equipping the upcoming generations with valuable knowledge and latest professional skills through education and research. We also aim to evolve new realities and foresight by unfolding new possibilities. We intend to promote the ethical, cultural and human values in our participants to make them educated and civilized members of society.

Contact: For all inquiries, regarding call for papers, submission of research articles and correspondence, kindly contact at this address:
Sector C, DHA Phase-VI Lahore, Pakistan
Phone: +92- 042-37181823
Email: ijeci@lgu.edu.pk

Published by: Digital Forensics Research and Service Center, Lahore Garrison University