FMDB *Transactions on Sustainable Intelligence and Security*

A Hybrid Intrusion Detection System Integrating Machine Learning and Signature-Based Techniques for Enhanced Network Security

Ayobola Victor Amubioya[1,*]

[1]Department of Computer and Information Sciences, Northumbria University, Newcastle Upon Tyne, England, United Kingdom.
ayobola.amubioya@northumbria.ac.uk[1]

Abstract: The fast growth of internet connectivity has made digital infrastructures, like enterprise networks, cloud platforms, and Internet of Things (IoT) environments, much more likely to be targeted by advanced cyber-attacks. As the threat landscape worsens, researchers need to develop advanced security systems capable of detecting both known and emerging threats. To address this issue, this study suggests a Hybrid Intrusion Detection System (IDS) that combines machine learning with traditional signature-based detection methods. The proposed system uses benchmark network datasets that have been thoroughly pre-processed to improve data quality and feature relevance. Then, different machine learning algorithms are used to model and predict anomalous network behavior that could indicate a security breach. These predictive outcomes are then used with signature-based detection to cross-validate the results, which makes detection more reliable. The hybrid approach aims to leverage the best of both methods: machine learning's ability to adapt to new threats and signature-based systems' ability to detect known attack patterns. The proposed Hybrid IDS significantly improves detection rates while lowering false positives compared to traditional intrusion detection systems, as shown by experimental results. Also, the system remains computationally efficient, making it well-suited for real-world, dynamic, large-scale network settings.

Keywords: Internet of Things; Intrusion Detection System; Internet Connectivity; Machine Learning; Digital Ecosystems; Cyber Attacks; Anomaly Detection; Signature-Based Detection; Batch Intrusion Analysis.

Received on: 28/05/2025, **Revised on:** 03/08/2025, **Accepted on:** 04/10/2025, **Published on:** 07/03/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSIS

DOI: https://doi.org/10.69888/FTSIS.2026.000666

Cite as: A. V. Amubioya, "A Hybrid Intrusion Detection System Integrating Machine Learning and Signature-Based Techniques for Enhanced Network Security," *FMDB Transactions on Sustainable Intelligence and Security,* vol. 1, no. 1, pp. 29-52, 2026.

1. Introduction

Smart Homes, Industrial Automation Systems, Healthcare Devices, and Cloud-Connected Sensors have transformed the Digital Ecosystem using Internet of Things (IoT) Devices. Today's highly distributed network environments constantly exchange large amounts of data. Although increased efficiency and automation result from the use of these systems, they also pose a wide range of previously unseen security risks. Many IoT devices run on low-powered hardware, have limited memory, and have poor security configurations; as a result, they become attractive targets for cyber attackers. Some common cyber-attacks include

[*]Corresponding author.

Denial-of-Service (DoS), Probing/Scanning, Brute-Force Authentication Attempts, and Remote-to-Local Privilege Escalation. Intrusion Detection Systems (IDS) are therefore a necessary component in protecting IoT Networks against Cyber Attacks by identifying Malicious Activity and monitoring Network Traffic. Signature-Based IDS performs well at detecting known attack patterns; however, it fails to identify Unknown Threats, including Zero-Day Threats. Machine Learning-Based Anomaly Detection Systems perform well when detecting Unknown Threats; however, they generate High False Positive Rates and Lack Transparency. Because these two types of systems have different advantages and disadvantages, there is a need for Hybrid IDS Approaches that combine the Advantages of Both Signature-Based and Anomaly-Based IDS Systems. Recent studies show that the combination of Machine Learning Classifiers and Deterministic Signature Matching Significantly Improves Detection Accuracy, Reduces False Alarms, and allows IDS Systems to Learn New Signatures from Anomalous Traffic Automatically. Therefore, the focus of this dissertation is to design and implement a Hybrid IDS for IoT Network Environments.

1.1. Statement of the Problem

Although the amount of IDS research has grown exponentially over the last few years, current detection methods still do not provide Complete and Reliable Protection for IoT Networks. Signature-Based IDS Require Frequent Updates and has no way of Detecting Zero-Day Threats. Although Anomaly-Based IDSs can identify unknown threats, including zero-day threats, They Are Often Unable to Generalise Well, Generate Excessive False Positives, and incur high computational overhead, making them unsuitable for resource-constrained IoT Environments.

1.2. Aim and Objectives

1.2.1. Aim

This paper aims to design, implement, and evaluate a Hybrid IDS for IoT Networks That Integrates Signature-Based Detection with Machine Learning-Based Anomaly Detection and deploys the system as a Functional Streamlit Web Application Capable of Performing Real-Time and Batch Intrusion Analysis.

1.2.2. Objectives

To achieve the above aim, the following objectives were pursued:

- To review existing literature on Signature-Based IDS, Anomaly-Based IDS, and Hybrid IDS Techniques within IoT environments.
- To obtain and preprocess a benchmark IDS dataset, including feature encoding, normalization, and exploratory data analysis.
- To implement and evaluate multiple machine learning algorithms, including Logistic Regression, Decision Trees, Random Forest, k-NN, and GNB for Anomaly Detection.
- To develop a deterministic Signature-Based Detection Engine that can identify known attack patterns.
- To integrate the signature-based engine with the machine learning classifier to create a hybrid detection workflow.
- To develop a self-learning mechanism that allows the system to generate and store new signatures from machine learning-detected anomalies automatically.
- To develop a Streamlit-Based Web Application to host the Hybrid IDS and allow for real-time analysis, batch processing, and Interactive Visualization.
- To Evaluate Performance Using Accuracy, Precision, F1-score, and Confusion Matrix Analysis.

1.3. Research Questions

This study seeks to answer the following research questions:

- How does the combination of signature-based detection with machine learning anomaly detection improve IDS performance in IoT networks?
- What is the comparative performance of different machine learning classifiers when used within a hybrid IDS framework?
- To what extent does the self-learning mechanism improve the system's ability to detect recurring attack patterns?
- How successful is the hybrid IDS at reducing false positives compared to standalone anomaly-based detection?
- How usable is the streamlit-based hybrid IDS for Real-Time IDS in IoT-Type environments?

2. Literature Review

2.1. Background

Modern Intrusion Detection Systems (IDSs) are critical components in protecting today's increasingly complex, evolving networks against an array of sophisticated cyber threats. In response to the continually emerging threat posed by increasingly sophisticated malicious actors, IDS technology has evolved beyond simple matching signatures. It now employs advanced hybrid and machine-learning-based analytical models. As a result, there is an ever-growing body of IDS-related research across computer security, artificial intelligence, data science, and network engineering.

2.2. Signature-Based Intrusion Detection Systems

Signature-based intrusion detection systems are among the earliest and most widely implemented forms of intrusion detection technology. These systems primarily function by comparing network traffic data to predefined collections of known malicious 'signatures' describing identified harmful network activities. Upon identifying a network activity consistent with a predefined attack signature, the system produces an alert. The primary strengths of signature-based intrusion detection systems lie in their accuracy and efficiency in identifying previously observed malicious activities, and in the low levels of false positives generated by their predefined rule sets, as they incorporate explicit characteristics or patterns of identified threats. Given the nature of their operation, they are extremely reliable in environments characterized by static threats; however, because they depend on prior knowledge of malicious threats, their inability to identify unknown attacks is a significant disadvantage in environments where cyber-attacks are becoming increasingly sophisticated. Therefore, a substantial shortcoming exists in the signature-based model.

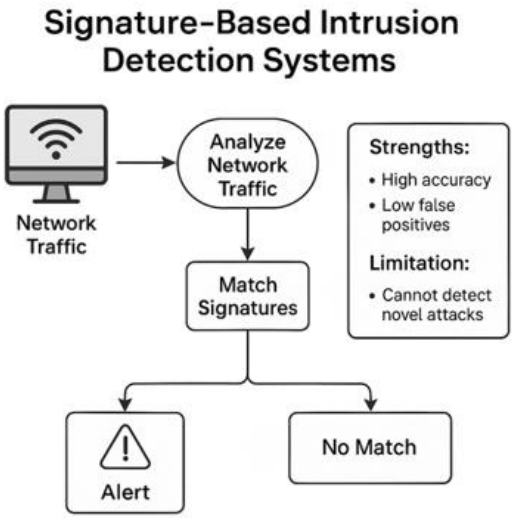

Figure 1: Strengths and limitations of the signature-based detection

As demonstrated in multiple studies, signature-based detection performs poorly at identifying unknown threats. Signature-based models can successfully detect simple versions of malware, but they are unable to identify variants that alter their internal structure. Attackers continually modify packet payloads and encrypt communications to evade signature-based detection, thereby diminishing the reliability of static IDS in today's rapidly changing environments. Additionally, the need to continuously update large signature databases creates operational challenges, including the need to deploy updates quickly and the possibility of deploying disparate rule sets across distributed IDS sensors, thereby hindering the implementation of a signature-based system. IoT environments consist of a variety of device configurations and consequently include many lightweight, resource-constrained devices. Consequently, edge-deployed signature-based IDS must operate efficiently. Signature matching becomes computationally expensive as the rule set grows [16], posing a challenge for IoT devices with limited processing power. Therefore, there is a need to create efficient, compact signature sets; however, doing so increases the number of false negatives, thereby compromising the balance between signature comprehensiveness and computational feasibility. Therefore, the applicability of signature-based IDSs in IoT deployments is complex.

Researchers have attempted to improve upon signature-based detection using machine learning techniques to generate and maintain optimal signatures. Subsequently, statistical modeling was applied to extract signatures from malicious traffic patterns automatically. Although automation enables faster signature generation, the underlying detection methodology still relies on

identifiable patterns in malicious traffic. The model assists in generating new signatures; however, it cannot identify attacks that do not follow the same structural pattern as the identified threats. These limitations further support the rationale behind integrating signature-based IDS with other detection methodologies. Papers such as Alasad et al. [2] compare the performance of signature-based and anomaly-based detection systems across various attack categories. The results clearly indicate that while signature-based systems produce few false positives for known attack categories, they fail miserably at reconnaissance and stealth attacks that depend on subtle differences in traffic patterns rather than fixed attack patterns. Signature-based IDS exhibits deterministic yet brittle behavior, making it suitable for static threat environments but not for the complex, rapidly evolving threat landscape typical of modern networks, as shown in Figure 1. Therefore, although signature-based IDS remains useful, it is inadequate and requires a complementary approach, such as anomaly detection, to address its shortcomings rather than replace it.

2.3. Anomaly-Based Intrusion Detection Systems

Although traditional anomaly-based intrusion detection systems (IDSs) can detect new attack types that do not match previously seen attack signatures, they remain inferior to signature-based IDSs due to the potential for a high number of false alarms. The use of machine learning (ML) techniques to enhance the accuracy of anomaly-based IDS has become an area of active study, where ML-based IDSs achieve superior performance compared to traditional statistical models by creating higher-dimensional representations of normal and abnormal network traffic. One of the major challenges in applying ML to IDS is the reliance on hand-selected features, which are typically selected by experts and are often prone to error. In addition, several issues arise when applying ML to IDS, including high dimensionality, high levels of noise, and varying traffic types generated by modern Internet of Things (IoT) systems. These challenges indicate that although ML-based techniques have improved over traditional anomaly-based IDS, they are still plagued by many of the same issues. Deep Learning (DL)- based anomaly detection systems have recently attracted attention due to the challenges posed by traditional ML-based techniques. DL-based systems can automatically discover hierarchical feature representations from raw data, thereby reducing the amount of human involvement required to develop an IDS. Research has shown that DL-based systems can discover both spatial relationships and temporal dependencies in network traffic data, enabling them to achieve high detection rates. Convolutional Neural Networks (CNNs) are particularly well-suited to learning spatial structures in network traffic matrices.

In contrast, Long Short-Term Memory (LSTM) networks are well-suited to learning sequential relationships between traffic flows. However, DL-based systems are also plagued by several challenges, including high computational overhead, the potential for overfitting, and the need for large amounts of high-quality training data. These challenges indicate that although Deep Learning is better than traditional ML-based systems, it is still not practical for widespread deployment. Most research papers present a consistent criticism of anomaly-based IDS: their unreliable performance across different datasets and deployment settings. Experimental results from multiple studies demonstrate that anomaly detectors generally do not achieve. A primary reason for this difference is the high variability of real network traffic, leading to a lack of representation of real traffic in curated datasets. As a result, anomaly-based IDS trained on laboratory datasets does not generalize well to operational networks. Anomaly-based IDS can also be affected by concept drift. Concept drift refers to the gradual evolution of the definition of "normal" behavior over time. Researchers have attempted to mitigate the impact of concept drift using adaptive learning techniques. Still, these add complexity to the system and increase the risk of model degradation if not properly controlled. This indicates that anomaly-based IDS is very powerful but also quite unstable, supporting the notion that they are most effective when combined with other stable detection mechanisms [4].

Another common criticism of anomaly-based IDS is that it can be vulnerable to adversarial evasion attacks. Since anomaly-based IDS relies on learning statistical or behavioral norms, adversaries can manipulate traffic to mimic legitimate behavior. Adversarial evasion attacks can also affect deep learning-based anomaly detectors, leading them to be misled by small perturbations added to traffic features. This threat undermines the idea that anomaly-based IDS is inherently more adaptive and safer than signature-based IDS. Many authors suggest combining anomaly detection with signature verification or ensemble classifiers to strengthen the overall system's resistance to these attacks. This further supports the trend towards combining anomaly-based IDS with other detection techniques.

In summary, anomaly-based IDS is considered necessary for detecting unknown attacks but is inherently unstable due to its high false alarm rates, sensitivity to noise, and vulnerability to adversarial attacks. Deep learning-based systems can greatly improve the performance of anomaly-based IDS but also incur high computational costs and require large amounts of high-quality training data, limiting their practical applications, particularly in IoT systems. The dominant research position suggests that the best way to utilize anomaly-based IDS is within hybrid architectures that combine the stability of signature-based systems with the adaptability of anomaly-based systems. Therefore, anomaly-based IDS is considered a powerful yet insufficient solution, and its limitations are motivating the increasing research activity in hybrid deep learning-based signature-augmented IDS.

2.4. Machine Learning Algorithms in Intrusion Detection

Machine learning has changed how researchers study Intrusion Detection Systems (IDS) as machine learning allows researchers to create automatic learning processes that can look for complex patterns of behavior within a stream of network traffic, rather than needing to manually write a set of rules that define what is considered "normal" and what is considered "attack." Signature-based systems rely on predefined attack patterns, whereas machine learning-based IDSs can generalize from training data to detect both known and unknown attacks. In this regard, Machine Learning provides significant value in security domains where attackers continually adapt their behavior and where there is no time to develop new attack patterns to stay up to date with emerging threats. Ahmed et al. [1] found that machine learning models greatly enhanced the detection of known and unknown attacks as compared to traditional signature-based systems; however, they also reported that the effectiveness of machine learning models varies greatly based upon the quality of the features selected, the type of algorithm used, and the quality of the training data. Therefore, while machine learning may achieve higher detection rates than signature-based systems, its use for intrusion detection requires careful selection of all technical components.

Many types of machine learning techniques have been developed for detecting intrusions. Among them, supervised learning has been one of the most heavily researched due to its ability to classify attacks into different classes. Several machine learning algorithms that have performed well at distinguishing between malicious and normal traffic include Random Forests, Support Vector Machines (SVMs), and ensemble methods such as XGBoost. Each of these algorithms takes in features derived from network traffic, such as packet and flow statistics and protocol behaviors, to build a classification model. Almotairi et al. [4] found that ensemble models combining the outputs of multiple machine learning models achieve higher detection rates than individual models in IoT environments. Research has also indicated that Random Forest algorithms achieve good results on noisy, unbalanced datasets commonly encountered in network security applications, and that SVMs provide good decision boundaries when the number of training examples is low. However, machine learning models ultimately depend on the quality of the feature engineering performed before building a classification model. Poor-quality feature engineering will result in lower detection rates, regardless of how sophisticated the machine learning model is [21].

Furthermore, this reliance on feature engineering quality is a significant limitation that requires domain expertise to select appropriate features for an IDS. Feature extraction and dimensionality reduction are two important preprocessing steps that can affect the overall performance of a machine learning IDS. Ahmed et al. [1] reported that selecting relevant features before classification can improve both accuracy and reduce computation time. There are several techniques for extracting the most informative features from network traffic and reducing the dimensionality of the input data, such as PCA, Correlation-Based Feature Selection, and Information Gain Metrics. Additionally, using fuzzy clustering to group similar features together can improve signature generalization and reduce false positives. While feature selection can improve the performance of an IDS, it adds another layer of complexity to the system; thus, the elimination of irrelevant features must be thoroughly validated to ensure that relevant attack indicators are not discarded. Almotairi et al. [4] emphasized that the feature selection method used in an IDS must be validated across various attack scenarios to maintain consistent detection. Ensemble learning methods have proven very effective in IDS research by combining multiple classifiers to improve robustness and accuracy. The combination of Random Forests with Gradient Boosting algorithms, such as XGBoost, has achieved state-of-the-art detection rates in numerous studies. Ensemble learning leverages the strengths of each classifier while minimizing its weaknesses through voting or weighting.

Research has shown that ensemble methods perform particularly well on imbalanced datasets, where attack samples make up only a small fraction of the total traffic. Alasad et al. [2] showed that CNN-XGBoost combinations improved robustness, particularly for detecting botnet-related anomalies. However, ensemble systems add complexity to the IDS and increase the time required to train it compared to a single classifier. As a result, ensemble systems are best suited to environments where accuracy is worth the added cost of computing power. Semi-supervised and Unsupervised learning techniques address the problem of limited labeled training data in IDS research. Bhavani and Mangla [7] demonstrated that Semi-Supervised learning techniques that combine clustering with Supervised Classification can detect attacks with little to no labeled training data. Clustering algorithms such as K-Means and DBSCAN group similar features together, enabling anomaly detection without requiring extensive labeling effort. However, unsupervised learning techniques generally yield higher false-positive rates than supervised learning techniques, since they lack labeled attack examples to guide their learning. Research indicates that Hybrid Architectures that utilize both Semi-Supervised and Supervised learning techniques strike a balance between pure supervised and unsupervised techniques, providing a reasonable compromise for real-world deployments where collecting large amounts of labeled data is impractical or too expensive. Machine Learning IDS optimized for IoT networks must contend with several constraints unique to the IoT environment, including limited processing resources, heterogeneous device capabilities, and variable traffic patterns.

Idris et al. [12] investigated the development of lightweight machine learning models that could run in IoT networks by reducing feature dimensionality while maintaining acceptable detection accuracy. Techniques such as feature hashing, decision tree

pruning, and linear classifier variants enable the deployment of machine learning models on resource-constrained edge devices. However, even with simplified machine learning models, complex computation offloading is required to gateway devices or the cloud infrastructure, which, in turn, raises concerns about latency, privacy, and communication reliability. Therefore, while machine learning provides a conceptually powerful approach to intrusion detection in IoT environments, practical implementations of machine learning IDS require distributed architectures that balance edge and cloud processing capabilities. A major concern in the literature regarding the use of machine learning for intrusion detection is the limited ability of these models to generalize when presented with traffic patterns that differ from those used during training.

Khraisat and Alazab [19] reported that many widely adopted IDS benchmark datasets suffer from class imbalance, outdated attack signatures, and limited diversity in the types of traffic included, resulting in high validation accuracy but low real-world network accuracy. Research has reported significant accuracy degradation when models trained on datasets such as nsl-kdd and KDD Cup 99 encounter modern attack variants or real-world traffic that is noisier and more complex than the training data. Researchers have utilized methods such as data augmentation, synthetic minority oversampling, and transfer learning to improve the model's generalization; however, these methods only partially address fundamental limitations in the training datasets. Therefore, the persistent problem of generalizing across diverse traffic patterns underscores the critical need for contemporary, diverse training datasets that accurately reflect current threat landscapes and network behaviors. In addition to the detection capability of a Machine Learning IDS, another Consideration is the interpretability of the machine learning model. While algorithms like decision trees and random forests provide relatively interpretable decision paths, more complex ensemble methods and nonlinear classifiers operate as black boxes, with reasoning processes remaining opaque. Khan et al. [16] emphasized that the lack of transparency poses challenges in security domains, where understanding why specific alerts were triggered is important for trust and compliance. Security analysts require insight into which features most influence classification decisions to validate alerts, diagnose false positives, and refine detection rules. Some research has investigated integrating explainable AI techniques with machine learning-based IDSs to improve interpretability; however, this often introduces additional computational overhead [14]. Therefore, the literature indicates that while machine learning provides excellent detection capabilities, its interpretability limitations complicate deployment in high-assurance or audit-sensitive environments [18].

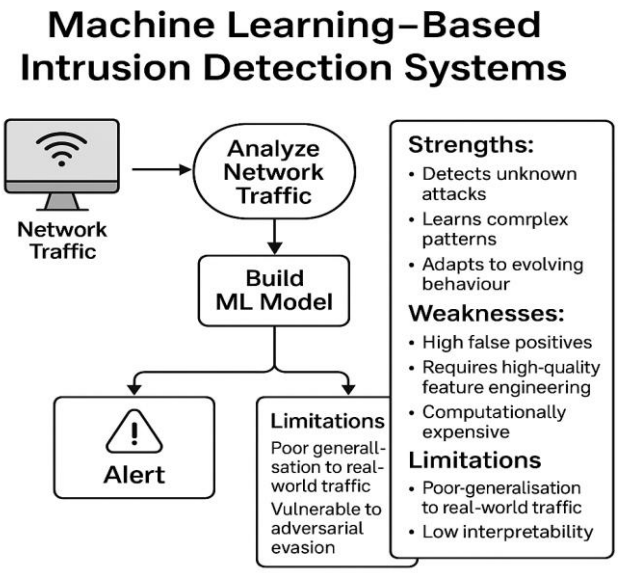

Figure 2: Strengths, Weaknesses, and Limitations of machine learning-based detection

Although machine learning is difficult to implement in many IDS systems, machine learning outperforms signature-only based systems in terms of intrusion detection when there are subtle differences between an intrusion and what has been previously seen. Ahmed et al. [1] showed that machine learning can successfully detect reconnaissance, protocol violations, and distributed attacks, which signature-based systems commonly miss [17]. The advantage of using machine learning for intrusion detection is that it uses statistical modeling of network traffic features, rather than simply looking for a pattern match. Machine learning will perform much better than traditional methods when provided with high-quality input data and sound training practices. However, machine learning will not always perform well and requires extensive tuning, testing, and validation to reach the levels of performance shown by previous researchers [20]. In fact, this variability represents one of the largest challenges in intrusion detection research; strong performance is possible but not guaranteed, regardless of the specific deployment context, as shown in Figure 2.

2.5. Hybrid Machine Learning and Signature-Based Intrusion Detection Systems

The Hybrid IDS is considered a promising direction for IDS research, as the community has realized that no single technique, either signature-based or anomaly-based, can provide a complete solution to the problem of intrusion detection. Signature-based IDS provides precise detection of the well-known attacks; however, it cannot detect unknown attacks. On the other hand, anomaly-based IDSs or machine learning models can detect unknown behaviors, but they lack precision. To leverage the strengths of both approaches, a new generation of IDS is emerging that combines deterministic signature matching with adaptive learning algorithms. In addition to the theoretical aspects, many recent studies have applied hybrid machine-learning models to enhance detection sensitivity while retaining the precision of signature-based IDS. For example, Ahmed et al. [1] have proposed a signature-based IDS enhanced with machine learning and deep learning approaches combined with fuzzy clustering. They demonstrated that selecting relevant features via fuzzy clustering before applying classification algorithms significantly improved accuracy and reduced computational costs.

Even though their architecture included both signature-matching and learning-based detection methods, they emphasized that machine learning components extend signature capabilities by recognizing patterns around known signatures, thereby enabling the detection of mutated attacks that partially match existing rules. Nevertheless, their study indicated that supervised learning frameworks require large, labeled datasets, which may not be available for all types of attacks. This limitation indicates that hybrid IDS must strike a balance between data requirements and model complexity to be applicable across diverse domains. Hybrid machine learning approaches have been used to improve the ability of signature-based IDS to detect modified attack variants. The application of intelligent clustering improves the generality of signatures. It reduces false positives when combined with machine learning classifiers, including Random Forest, SVM, LSTM, and ANN, indicating that machine learning models complement signature mechanisms by recognizing extended patterns around known signatures, thereby enabling the detection of modified attacks that partially resemble existing rules. While architecture provides excellent results, ensemble integration poses challenges for model transparency and Interpretability. Interpretability observation highlights an ongoing trade-off in hybrid IDS design: improving detection capabilities often comes at the cost of explainability, a crucial requirement for operational environments. Additionally, lightweight hybrid IDS for IoT is gaining significant attention due to the constrained nature of edge devices. Kakolu et al. [24] emphasize that resource-constrained IoT devices require intrusion detection mechanisms specifically tailored to meet their security effectiveness needs while balancing computational efficiency. Research demonstrated that distributing detection tasks across network layers, with signature filtering or lightweight anomaly detection at the edge and more complex machine learning analysis running on gateway or cloud infrastructure, is a feasible implementation of hybrid IDS principles. Some hybrid IDS have also combined machine learning with lightweight classifiers or explainable components to achieve a balance between accuracy and Interpretability.

Nawaal et al. [22] introduced a hybrid, lightweight deep learning algorithm that enhances early-stage detection by simplifying the architecture while maintaining strong performance. Their results demonstrate that hybrid models consistently outperform single-architecture IDS across most attack categories, illustrating the benefits of architectural diversity in threat detection. Nonetheless, the authors noted that hybrid models are challenging to fine-tune and thus require extensive hyperparameter optimization. This challenge emphasizes that hybrid IDS research must strike a balance between complexity and practicality to ensure that performance gains are not achieved at the expense of maintainability. While Hybrid IDS is primarily focused on integrating signature-based detection and machine learning, many researchers are also evaluating Hybrid IDS systems that combine both Nawaal et al. [22], for example, showed that combining machine learning classification with traditional signature-based detection achieved the precision of signature-based detection while retaining the adaptability of statistical learning.

They found that adding machine learning classification maintained a very low false-positive rate for known attacks while also significantly improving the ability to identify new variations of known attacks. This study supports the claim above that combining deterministic and probabilistic detection methods yields an optimal detection strategy that achieves precision without sacrificing coverage. Researchers also emphasize that, for Hybrid IDS to be considered reliable outside a laboratory setting, it must be tested in realistic ways using realistic datasets and deployment scenarios. In addition to machine learning combined with signature-based detection, researchers have also studied hybrid IDS architectures that combine machine learning with protocol-specific detection mechanisms. Schrötter et al. [23], for example, compared the performance of a neural network-based intrusion detection system with that of a signature-based system in IoT networks. They concluded that a Hybrid IDS system combining a neural network and signature-based detection achieved better detection results than either method alone.

Specifically, they stated that signature-based detection achieved excellent results for detecting previously identified attack patterns, with very few false positives. Machine learning-based detection identified previously unknown attacks and/or attack variants [11]. While this complementary relationship validated the basic premise of Hybrid IDS, i.e., that different detection mechanisms can be used to detect different types of threats, it also clearly illustrated the importance of understanding the strengths and weaknesses of each method. Finally, researchers have studied the performance of Hybrid IDS across various network environments and attack scenarios. Bhavsar et al. [8], for example, developed an anomaly-based intrusion detection

system for IoT applications that combines statistical profiling with machine learning classification. Their evaluation over multiple attack types demonstrated that Hybrid IDS systems produce consistent detection results across diverse threat categories. In contrast, single-method systems tend to produce inconsistent detection rates, favoring specific attack families. Thus, the ability of Hybrid IDS systems to produce consistent results across various threat categories provides a significant practical advantage, as operational security systems must reliably detect a wide range of threats rather than being optimized for a narrow range [5].

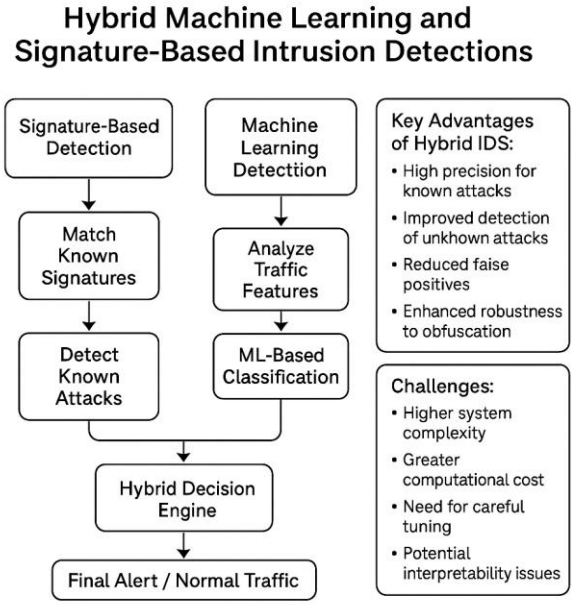

Figure 3: Strengths and limitations of the hybrid IDs

A few researchers have also studied integrating fuzzy logic with machine learning to enhance the adaptability of Hybrid IDS. Ahmed et al. [1] demonstrated that fuzzy clustering enhances feature grouping and signature generalization, allowing machine learning classifiers to better handle ambiguity in traffic classification and borderline cases. The fuzzy logic decision boundary is soft; it accommodates the inherent ambiguity in network traffic classification, thereby reducing false positives while maintaining sensitivity to true threats. This work shows that Hybrid IDS systems can use additional computational paradigms (deterministic signatures, statistical learning, and fuzzy reasoning) to develop more robust detection systems. The practical deployment considerations for Hybrid IDS systems are attracting increasing interest as research transitions from theoretical models to operational systems. Immastephy and Punitha [13] conducted a systematic review, emphasizing that Hybrid IDS systems must consider both detection accuracy and deployment feasibility, including installation complexity, long-term maintenance requirements, and compatibility with other security systems. Their review highlighted that successful Hybrid IDS deployments require careful architectural planning to ensure that system components interact effectively and do not cause unnecessary delays due to processing bottlenecks or synchronization errors, as seen in Figure 3.

2.6. Identified Gaps in Literature

Although intrusion detection research has expanded significantly in recent years, several gaps remain, particularly in hybrid systems that combine signature-based detection with deep learning architectures. Many current studies focus solely on anomaly-based deep learning models for IoT environments, as seen in the work of. Aljumah [3], Bhavsar et al. [8], and Hossain [10]. These studies demonstrate strong detection capabilities for unknown attacks but offer limited consideration of how deep learning could complement signature-based systems. The absence of a unified framework that integrates deterministic signature analysis with adaptive CNN-based learning means that signature accuracy and anomaly detection capability often remain isolated within separate systems rather than functioning as mutually reinforcing components. This gap highlights the need for a combined approach that preserves the strengths of both detection strategies. Another significant gap concerns the limited attention to the generalisability of hybrid IDS across heterogeneous environments. Research by Azar et al. [6] and Khan et al. [16] demonstrates that hybrid IDS models can achieve high accuracy on specific datasets. However, these models are often trained and validated on limited or synthetically augmented datasets such as NSL-KDD, CICIDS2017, or Bot-IoT. These datasets do not fully reflect the complexity and diversity of real-world traffic patterns within modern IoT networks. As a result, even high-performing models may struggle to generalize when deployed in practical environments where traffic behavior varies

widely across device types, protocols, and applications. This indicates a need for hybrid IDS research that explicitly evaluates cross-environment generalisability using diverse and realistic traffic data. A further gap concerns the integration of signature-based techniques within deep learning pipelines. Studies such as Ahmed et al. [1] and Nawaal et al. [22] explore enhancements to signature-based IDS using machine learning and clustering. Yet, very few investigate how CNNs can be used to refine signature matching or to identify extended behavioral patterns associated with known signatures. Most hybrid systems in the literature, for example, those by Alasad et al. [2], combine CNN feature extraction with machine learning classifiers but do not incorporate signature rules as a core component of the detection process. This leaves a conceptual and technical gap regarding how signature logic can be systematically harmonized with deep learning outputs.

The gap is particularly important because signature-based systems remain the primary line of defense in many enterprise environments and therefore require deeper integration within modern architectures. There is also a significant gap in lightweight hybrid IDS frameworks specifically optimized for environments with constrained computational resources. While studies by Kamal and Mashaly [15] and Yang et al. [25] propose lightweight hybrid models, most deep learning-based hybrid IDS research assumes access to powerful computational infrastructures such as cloud servers or GPU-enabled fog nodes. This assumption limits the feasibility of deploying hybrid IDS on edge devices, where IoT applications often require on-device preprocessing for privacy, efficiency, or latency. The limited exploration of resource-aware hybrid architecture highlights a need for research focused on designing hybrid IDS that maintain high accuracy while operating within strict computational budgets. This gap is directly relevant to networks where both immediacy and efficiency are critical for security monitoring. The complexity of hybrid intrusion detection system design introduces another research gap concerning Interpretability. Although models such as SecureNet achieve high accuracy, they offer limited transparency into how classification decisions are made.

Deep learning components, especially CNN and LSTM layers, operate largely as opaque black boxes, while signature-based systems provide highly transparent decision pathways. Despite this contrast, existing hybrid IDS research rarely addresses how to reconcile interpretability with the opacity of deep learning outputs. This results in hybrid systems that deliver strong performance but may be unsuitable for environments that require detailed auditing, forensic investigation, or compliance reporting. A clearer focus on explainable hybrid IDS architectures would therefore help bridge the gap between performance and operational transparency. A further gap concerns the scarcity of research that evaluates hybrid IDS performance in adversarial settings. Although Farhan and Jasim [9] and Schrötter et al. [23] demonstrate that deep learning IDS are vulnerable to adversarial manipulation, very few studies explore how hybrid systems behave when confronted with adversarial traffic designed to evade signature matching. Without adversarial evaluation, the resilience of hybrid IDS remains uncertain, especially given that attackers increasingly employ techniques such as payload obfuscation, protocol manipulation, and traffic mimicry. This indicates a critical need for hybrid IDS research that deliberately assesses robustness against adversarial threats and incorporates defensive strategies into the model design.

3. Methodology

This presents the practical implementation of the proposed Hybrid Intrusion Detection system using the Agile Iterative Development Methodology. This methodology was chosen because it enables incremental implementation of the system, enabling continuous testing, refinement, and performance evaluation without disrupting the development lifecycle (Figure 4).

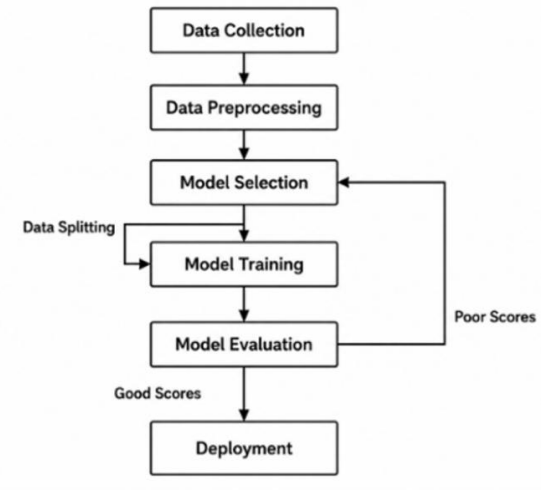

Figure 4: The flowchart of the developed machine learning model

This paper aims to design, implement, and evaluate a Hybrid IDS for IoT Networks that integrates signature-based detection with machine learning-based anomaly detection and deploys the system as a functional streamlit web application capable of performing real-time and batch intrusion analysis. This section outlines the system requirements for developing the hybrid intrusion detection system. It describes the software and hardware specifications, functional requirements, and non-functional requirements to ensure efficient system performance and reproducible results. The choice of Python as the language for our implementation is due to its widespread usage in science today. As a user-friendly, free, object-oriented programming language, Python supports other programming paradigms, including structured, functional, and procedural programming.

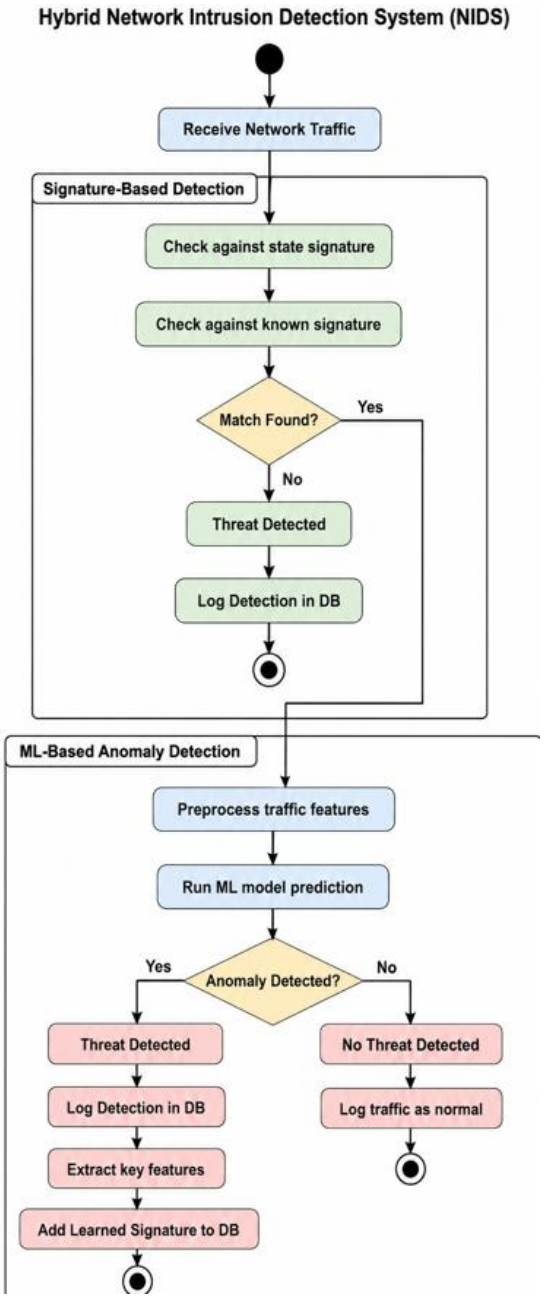

Figure 5: The hybrid detection system dataflow diagram

The layers of the proposed hybrid intrusion detection system include the data layer, preprocessing layer, analysis layer, modeling layer, detection layer, and application layer, which work together to perform different tasks as part of the overall intrusion detection process. Raw network traffic is collected at the Data Layer from a public intrusion-detection dataset or packet-capture recordings. The Preprocessing Layer cleans, transforms, encodes, and normalizes data collected at the Data Layer, preparing it for further processing by other layers. The Modeling Layer trains the Machine Learning algorithm on the dataset created by the Preprocessing Layer, which has been cleaned, transformed, and normalized; the model then learns to

distinguish between normal and abnormal traffic patterns. At the Detection Layer, the results from the trained Machine Learning classifier are combined with Signature Rules defined before the modeling process, creating a hybrid decision-making engine that enhances both the accuracy and robustness of detection mechanisms. Finally, the Application Layer provides the actual environment in which the model is deployed. This includes a Streamlit-based Web Interface that enables end users to interact with the model in real time, predict network behavior, and monitor network activity. This layered architecture design, therefore, allows independent testing, maintenance, and improvement of each layer as needed; it also improves the system's scalability and reliability, as shown in Figure 5. This graphic depicts the workflow of a Hybrid Network Intrusion Detection System (NIDS) that employs signature-based detection and machine learning-based anomaly detection. The incoming network traffic is checked against known attack signatures. To find known threats and log them in the database. When no signature match is detected, traffic is analyzed using a machine learning model to detect abnormal behavior. Anomalies are logged, important features are extracted, and new signatures are added to the database to improve future threat detection capabilities.

3.1. Data Collection

The dataset used for this study was obtained from Kaggle [26], also known as the KDD Cup 1999 intrusion detection dataset, which is one of the most widely recognized benchmark datasets in the field of network security and intrusion detection research. The dataset was originally developed for the KDD Cup competition and is based on a simulated military network environment that generated a large volume of labeled network traffic records. The availability of labeled traffic and the wide variety of attack types make the KDD99 dataset particularly suitable for developing and evaluating the proposed hybrid signature-based network intrusion detection system.

3.2. Data Preprocessing

Since the KDD99 dataset contains both numerical and categorical attributes, several preprocessing steps are required to ensure the data is suitable for machine learning algorithms. Initially, the dataset is checked for missing values, inconsistencies, and duplicate entries. KDD99 includes repeated connection records due to simulation conditions; duplicate rows are identified and removed to minimize bias during training. The categorical features are converted to numerical values using appropriate encoding techniques. This conversion is essential because most machine learning algorithms operate solely on numeric data. After encoding, all numerical features are normalized to a standard range of 0 to 1. This prevents features with large numeric ranges from dominating the learning process.

3.3. Data Splitting

For the paper, the dataset is split into a 70:30 training: test ratio. The training set is used to train the classification model, and the test set is used to evaluate the model's performance. The data is split to avoid overfitting, a common concern in model building.

3.4. Machine learning algorithm Implementation

The implementation of machine learning algorithms for network intrusion detection was accomplished using the Scikit-learn (sklearn) library. This comprehensive Python framework provides robust implementations of a range of machine learning techniques. Each algorithm was systematically constructed, trained, and evaluated using a comprehensive set of performance metrics, including accuracy, sensitivity (recall), specificity, F1-score, precision, and confusion matrix analysis. These evaluation metrics provide a holistic view of each algorithm's performance in detecting network intrusions, with particular emphasis on minimizing false negatives and false positives.

3.5. Model Deployment

After the training and evaluation processes, the best-performing model is selected for deployment. The trained model is saved using a model serialization technique such as joblib or pickle, allowing it to be reused without retraining. The model is then integrated into an application environment where it can receive live network feature inputs and output predictions in real time. Security considerations are taken into account during deployment. These include input validation, secure storage of the model file, and protection against unauthorized access. This stage ensures that the hybrid intrusion detection system is not only accurate but also secure and reliable in a real-world setting.

3.6. Web Application Development using Streamlit

To provide a user-friendly interface, a web application is developed with Streamlit. This application allows users to manually enter network traffic features or upload a dataset for batch analysis. The Streamlit interface includes visual elements, such as

buttons, sliders, and charts, to make interactions more intuitive. Once the user provides the required input, the application processes it with the deployed model and displays the result as either normal traffic or an intrusion. Additional statistics, such as prediction confidence and classification summary, can also be displayed.

3.7. Exploratory Data Analysis

An exploratory analysis of the KDD99 dataset is conducted to better understand the distribution of network traffic and the prevalence of different attack categories. Statistical analysis of the features provides insight into their distributions and variability, helping to identify redundant or low-importance attributes. Correlation analysis identifies highly correlated features that may introduce redundancy or multicollinearity into the model. Removing or limiting such features helps to improve learning efficiency without significantly reducing predictive power. The exploratory stage also helps identify the most influential features related to attack behavior, such as the number of connections within a given time window, the number of failed login attempts, and the frequency of service access. These findings support more informed decisions in feature selection and model optimization (Figure 6).

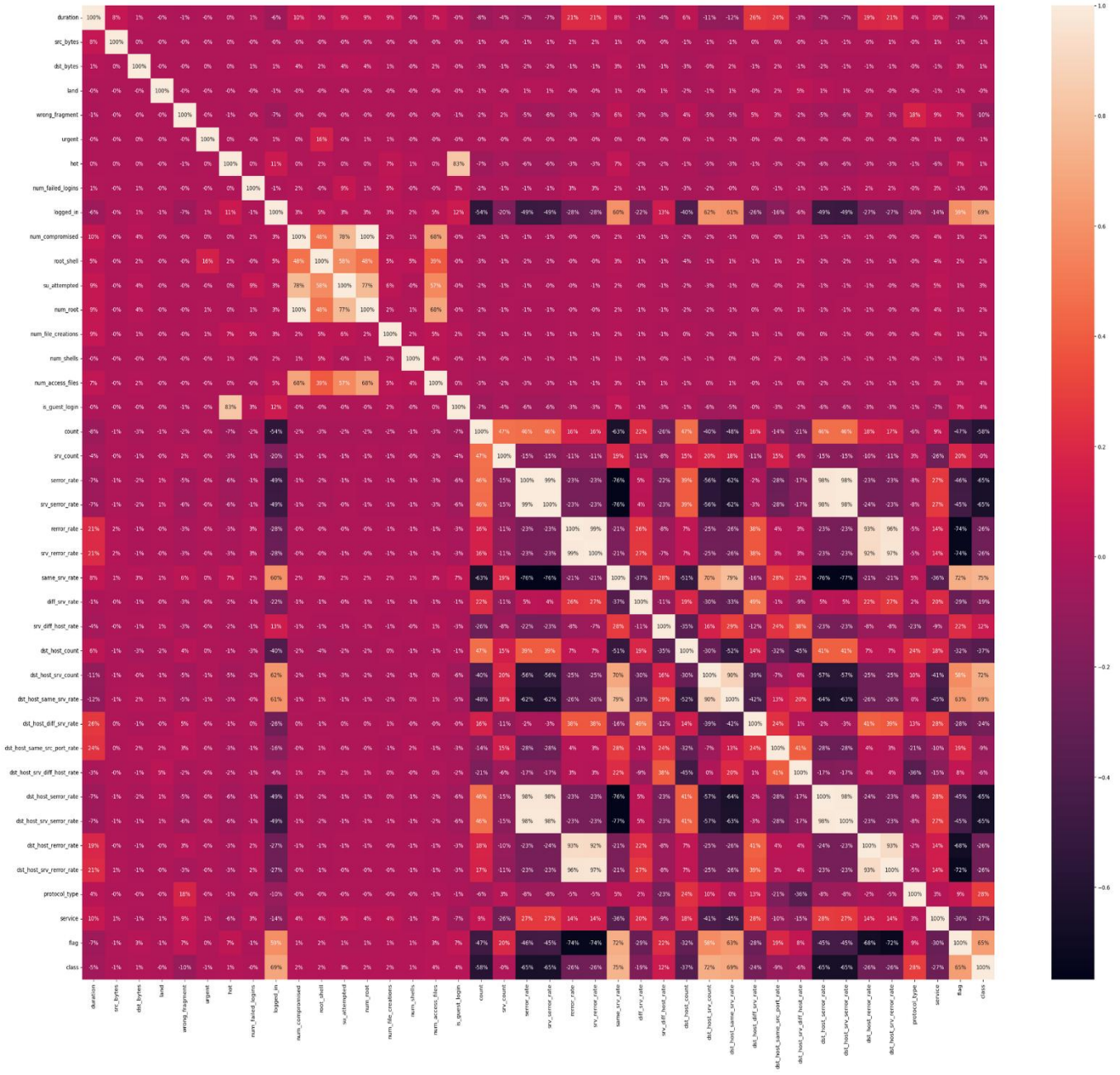

Figure 6: Correlation plot of the KDD99 dataset

3.8. Web Application and Hybrid Detection Modules

The Streamlit web application integrates a deterministic signature engine with a Random Forest anomaly detector to form a transparent Hybrid IDS. The design foregrounds Interpretability, Interpretation, and a lightweight persistence mechanism. Real-time and batch analysis capabilities, together with visual dashboards, facilitate both exploratory research and demonstration of the self-learning hybrid concept. The architecture is therefore well aligned with the dissertation objective of exploring a practical hybrid signature-based system augmented by CNN-classified or Random Forest-classified anomalies trained on KDD99.

3.8.1. Overview of the Web Application

The web application provides a user-facing interface for the hybrid Intrusion Detection System. It is implemented using Streamlit and integrates the signature-based detection engine with a machine learning anomaly detector. The application supports single-connection real-time analysis and batch CSV processing, and it persistently records detection results and learned signatures in an embedded SQLite database. Visualization components built with Plotly provide immediate user feedback through summary metrics and charts. The application, therefore, serves both as an interactive research tool for experimentation and as a lightweight demonstration platform for proof-of-concept evaluations.

3.8.2. Front-end Design and User Interaction

The user interface is organized into five principal tabs: Real-Time Analysis, Batch Analysis, Detection Log, Learned Signatures, and About. A persistent sidebar displays configuration information, key statistics, and metrics from the signature database. The Real-Time Analysis tab presents a form composed of the 26 expected KDD/NSL-KDD features. Sample records are provided for quick testing, so users can populate the form with representative normal or attack traffic and submit a single record for analysis. The Batch Analysis tab accepts a CSV file and iterates through each row, analyzing records in sequence and returning a downloadable CSV of results. The Detection Log tab displays stored detection events and offers visualizations of traffic distribution, attack categories, and a detection timeline. The Learned Signatures tab lists signatures that the system automatically created after ML detections and allows export. The About tab documents the architecture, the dataset used, and the technical stack.

3.8.3. Signature-Based Detection Module

The signature-based detection module is the first line of defense inside the hybrid system. It is implemented in a class that loads a set of signatures, each encoded as a dictionary of feature constraints. Each signature corresponds to a known attack pattern from the KDD99 taxonomy, such as Denial-of-Service, Probe, Remote-to-Local, or User-to-Root attacks. The matching logic iterates through signature rules and counts the number of rule matches for each signature. Numeric rules are evaluated using a greater-than-or-equal comparison, and categorical rules are evaluated using equality. A configurable match threshold is applied; in the current implementation, a signature fires when at least 70% of the rules match. The module also retrieves learned signatures from the database and applies the same matching procedure to these user or system-derived patterns. When a signature match occurs, the module returns the detection result, including the category, attack type, and a computed confidence value representing the proportion of matched rules.

3.8.4. Machine Learning Anomaly Detection Module

The machine learning component serves as a second line of defense to detect unknown, or zero-day attacks not covered by the base signature set. The ML Anomaly Detector class loads a serialized Random Forest model from the file model.pkl. The detector expects a fixed feature ordering that matches the model training stage. Preprocessing in the web application is minimal and deterministic: a one-row pandas Data Frame containing the 26 required features is constructed and supplied to the model. A non-normal label is interpreted as an attack. The current implementation returns a nominal confidence placeholder for ML detections; this placeholder can be replaced by model probabilities if required.

3.8.5. Hybrid Orchestration and Self-Learning

Hybrid orchestration is handled by a controller class that manages the two separate detection components. The analysis process initially initiates the signature-based detector. Upon receiving a base or learned signature, the process stops, and the incident is recorded in the log file. However, if no match is identified, the controller invokes the ML detector. Once the ML detector identifies an anomaly and the "auto learn" option has been selected, the controller captures a compact signature pattern from selected features, then inserts it into the SQLite learned_signatures table. To prevent duplicate signatures, the SQLite database's insert mechanism ensures that repeated signatures are incremented rather than duplicated. All detection incidents are recorded

in the detection_log table, along with the date/time stamp, the method used to detect it, the category, the type of attack, the detection confidence level, and whether a signature was added to the signature database. Therefore, the system's self-learning capability enables it to expand its signature database over time as new ML-detectable anomalies recur.

3.8.6. Database and Persistence Layer

The SQLite database management system is used to implement the persistent storage layer. The database includes a single schema that consists of a detection_log table for recording all detection-related events, with a timestamp (date/time), and a learned_signatures table for storing JSON-encoded signature patterns and metadata. The detection log table is used for both auditing and visualization. In contrast, the learned signatures table serves as persistent memory for the application, allowing it to recover previously established signatures upon restart. The database abstraction is encapsulated in a class that manages database initialization, inserting new records into tables, querying tables for specific information, and performing atomic updates.

3.8.7. Visualization and User Feedback

The Streamlit application uses the Plotly library to create interactive visualizations that help users quickly understand how the system behaves. A pie chart shows the proportions of normal and flagged traffic. A bar chart illustrates the various attack categories in the detected events. A simple scatter plot presents a timeline view of the detection sequence and severity. The detailed detection logs are displayed as a table, allowing users to inspect each event and the exported results. The Learn Signatures Tab allows users to view both the human-readable display of the patterns and a JSON export function so that analysts can save or use the generated signatures for future use.

3.8.8. Integration with KDD99 Model and Feature Mapping

The Streamlit application expects to receive feature vector data that has been processed using the same preprocessing and feature ordering as in the model training with the KDD99 dataset. The MLAnomalyDetector class exposes an ordered list of features that the user must map to the column names used when training the Random Forest classifier. As model predictions must remain consistent with the training data, this rigid mapping is enforced. The application maps the categorical feature types (e.g., protocol type, service) to their corresponding string values for the signature engine; however, it assumes that the training pipeline has encoded the categorical feature values as required by the ML model.

3.8.9. Activity Diagram

The activity diagram below describes the full scope of how a Hybrid Intrusion Detection System (Hybrid IDS) functions, from initial data intake and pre-processing through all subsequent steps, where two parallel methods of detecting intrusion are employed: Signature-Based Detection and Anomaly Detection employing Machine Learning algorithms. In the Signature-Based Path, each data packet received by the Hybrid IDS is processed and matched against known (predefined) or newly learned signatures (i.e., patterns of behavior).

Once a signature pattern in a packet matches one of the signatures stored in the system's databases, the Hybrid IDS will automatically flag a potential threat, assign an attack category and type, generate a confidence score, and log the detection event. If no signature is found for the incoming packets, they are then sent to the ML-based Anomaly Detection Pathway. This pathway employs a Machine Learning algorithm that predicts the likelihood that the observed data packet behavior is anomalous. When a prediction indicates that the packet's behavior is anomalous, the Hybrid IDS assigns a "malicious" designation and an attack label based on the machine learning model's output. The Hybrid IDS also generates a confidence score (as in the signature-based path).

When Auto-Learn is enabled, it extracts key features from the packet, creates a new learned signature (pattern), stores this information in its databases, and updates the detection methodology used for future packet analysis. When a prediction indicates that the packet's behavior is not anomalous, the Hybrid IDS assigns a "normal" designation and a "no threat" status to the packet and logs the determination. The diagram demonstrates the function of a hybridized adaptive detection mechanism in which signature-based rules and machine learning models work together to achieve greater overall accuracy while continually evolving the Hybrid IDS's threat intelligence (Figure 7).

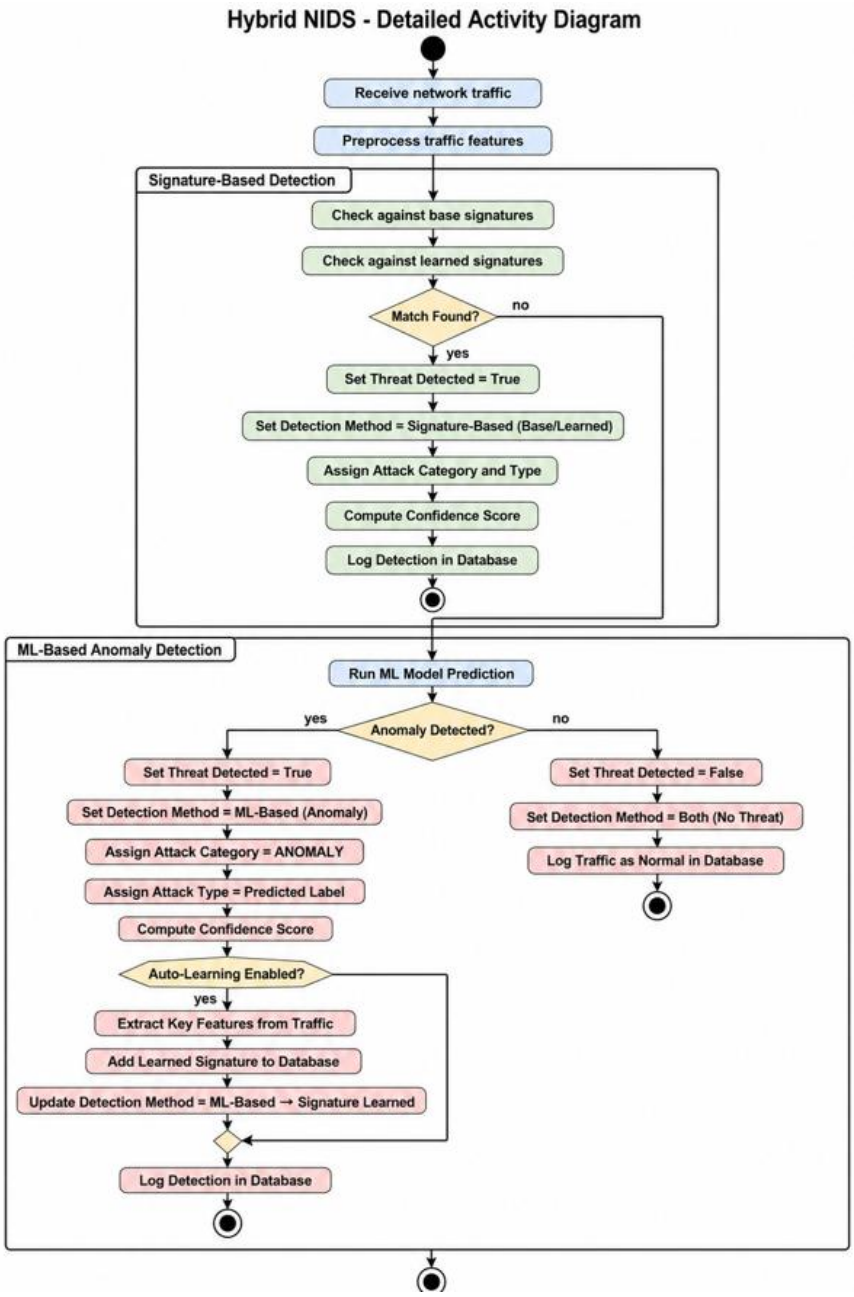

Figure 7: Activity diagram of the system

3.9. Ethical and Legal Considerations

The research described here was completed utilizing publicly accessible benchmarking datasets, including the KDD Cup 1999 dataset. No human subjects were used in this research, and no individual's personally identifiable or sensitive information was collected or analyzed. Therefore, the research itself does not pose a direct risk to individuals' privacy, confidentiality, or their rights to personal data protection. Additionally, access to the web application has been strictly limited to an internal experimental environment to prevent unauthorized third-party access. The secure management of stored model files and detection log files was ensured to minimize risks such as tampering, data corruption, or misuse. The research complies with the ethical standards and guidelines for computational research and adheres to all applicable data protection principles.

3.10. Limitations of Methodology

There are several limitations associated with the methodology utilized in this study. The most significant limitation is the reliance on the KDD99 dataset, which was artificially generated to simulate a network infrastructure environment and may not

be representative of modern networks or emerging types of cyberattacks. Therefore, the machine learning model developed is likely to perform less well when applied to real-world network traffic. Further, since the system was tested in a simulated environment (i.e., offline), rather than through live packet captures, the real-time operational effectiveness of the hybrid IDs could not be assessed. Additionally, because the machine learning approach relies on labeled training data, the quality and representativeness of that data can affect the effectiveness of the detection model. Lastly, the initial implementation of the system focused on a limited set of features and attack classes, potentially limiting its ability to detect additional threat types.

3.11. Justification of Methodology

The methodology selected for this research was justified based on the need to develop both a functional detection model and a deployable detection platform. The use of the Agile Iterative development process provides a systematic method of developing, testing, and refining a useful artifact. This paper describes the design and implementation of a Hybrid Intrusion Detection System (Hybrid IDS). This system combines signature-based intrusion detection for known attacks with machine learning-based detection for unknown attacks. Furthermore, this system has a self-learning capability, so that patterns identified by the ML component are converted into new signatures, thereby improving detection capabilities in subsequent cycles. The system also includes a user interface (built with Streamlit) that enables users to analyze network traffic in real time and in batch. Users can interactively view network traffic visualization displays, reports, and analysis related to all detection events and learned signatures, which are stored in an SQLite database for persistent storage to enable historical analysis and statistics.

4. Results and Analysis

The purpose of the paper is to present the implementation results of the HIDS proposed for IoT networks. This paper provides an in-depth analysis of all the Machine Learning models evaluated, including their performance indicators and relative performance. To determine which model would be the best choice for deploying as a HIDS on the chosen IoT traffic dataset, multiple Supervised Learning algorithms were trained, analyzed, and compared. The classification report provides an overview of a classification model's operation and a comprehensive evaluation of its efficiency. A classification report is one of the primary tools for evaluating how well a classification model predicts classes or categories. The classification report typically includes the most important criteria, providing insight into the model's predictive accuracy:

- **Precision**: The percentage of positive predictions that are accurate; i.e., the percentage of real anomaly instances identified by the model relative to the total number of instances identified as anomalies. High precision is a significant factor in reducing false positives and preventing unnecessary alerts in network intrusion detection.
- **Recall (Sensitivity)**: The proportion of actual anomalies a model correctly identifies. High recall is an essential feature of intrusion detection since failing to detect an attack could be catastrophic.
- **F1 Score**: The F1 score is the harmonic mean of precision and recall, and therefore, it is a balanced metric of performance.
- **Support**: There were 3505 anomaly examples and 4053 normal examples in the dataset; thus, the dataset contained adequate numbers of both classes. Additionally, given the high-performance metrics of the models across such varied class distributions, it may be concluded that the models handled unbalanced datasets effectively without bias toward either class.
- **Accuracy**: The overall accuracy of the models further supports their performance, with Decision Tree and Random Forest achieving 99.7%, KNN achieving 99.1%, Logistic Regression achieving 95.4%, and Gaussian Naive Bayes achieving 87.8%. The overall accuracy rates support the notion that the hybrid approach can effectively distinguish between normal and anomalous traffic.

4.1. Logistic Regression Classifier

Compared with other models, the Logistic Regression classifier demonstrated moderate but acceptable performance; it could not model the complex, non-linear relationships inherent in IoT traffic flow. Thus, it will not serve as the preferred detection method for the Hybrid Intrusion Detection System, as seen in Table 1.

Table 1: Classification report for logistic regression classifier

Model [0] Logistic Regression				
Class / Metric	**Precision**	**Recall**	**F1-Score**	**Support**
Anomaly	0.95900	0.94094	0.94988	3505
Normal	0.94975	0.96521	0.95742	4053
Accuracy			0.95396	7558

Macro avg.	0.95437	0.95308	0.95365	7558
Weighted avg.	0.95404	0.95396	0.95392	7558

4.2. Decision Tree Classifier

Decision Trees demonstrated strong performance relative to the simpler linear models tested. Decision Trees can capture nonlinear relationships in the dataset. While decision trees can stand alone, they often fit the training data too closely. The performance of decision trees may fluctuate with new, unseen IoT traffic unless stabilized using ensemble methods, as shown in Table 2.

Table 2: Classification report for the decision tree classifier

Model [1] Decision Tree Classifier				
Class / Metric	**Precision**	**Recall**	**F1-Score**	**Support**
Anomaly	0.99771	0.99601	0.99686	3505
Normal	0.99655	0.99803	0.99729	4053
Accuracy			0.99709	7558
Macro avg.	0.99713	0.99702	0.99707	7558
Weighted avg.	0.99709	0.99709	0.99709	7558

4.3. Random Forest Classifier

Random forest provides the best results among algorithms for Hybrid Intrusion Detection Systems because random forests minimize overfitting by using both bagging and tree averaging to reduce variability in the model; random forests can handle noisy data with high-dimensionality of IoT feature sets exceptionally, Random forests perform better than other models on all of the key performance metrics and are more robust for practical use in a production environment as seen in the Table 3.

Table 3: Classification report for random forest classifier

Model [2] Random Forest Classifier				
Class / Metric	**Precision**	**Recall**	**F1-Score**	**Support**
Anomaly	0.99715	0.99658	0.99686	3505
Normal	0.99704	0.99753	0.99729	4053
Accuracy			0.99709	7558
Macro avg.	0.99709	0.99705	0.99707	7558
Weighted avg.	0.99709	0.99709	0.99709	7558

4.4. K-Nearest Neighbours (KNN)

KNN demonstrated strong performance and was among the top performers in the testing phase.

Table 4: Classification report for K-nearest neighbors (KNN) classifier

Model [3] KNeighbors Classifier				
Class / Metric	**Precision**	**Recall**	**F1-Score**	**Support**
Anomaly	0.99226	0.98745	0.98985	3505
Normal	0.98919	0.99334	0.99126	4053
Accuracy			0.99061	7558
Macro avg.	0.99072	0.99039	0.99055	7558
Weighted avg.	0.99061	0.99061	0.99060	7558

Although KNN performed well, its computational cost in large-scale or real-time IoT environments is prohibitive because it requires computing distances to every sample during inference. This greatly reduces KNN's scalability compared to Random Forest, as shown in Table 4.

4.5. Gaussian Naive Bayes

Gaussian Naive Bayes performed the worst among the models tested. The independent-feature assumptions of Naive Bayes do not align well with real-world IoT network data, which often contains many interrelated features. Therefore, the model will not perform adequately for critical security applications, as seen in Table 5.

Table 5: Classification report for the Gaussian Naive Bayes classifier

Model [4] Gaussian Naive Bayes Model				
Class / Metric	**Precision**	**Recall**	**F1-Score**	**Support**
Anomaly	0.86101	0.88017	0.87049	3505
Normal	0.89434	0.87713	0.88565	4053
Accuracy			0.87854	7558
Macro avg.	0.87767	0.87865	0.87807	7558
Weighted avg.	0.87888	0.87854	0.87862	7558

4.6. Confusion Matrix

Confusion matrices are generally used to evaluate the performance of all types of classification models. They provide a detailed view of how accurately each classification model predicts the correct class label relative to the class label it should have assigned. A confusion matrix shows the total number of true positives and false positives, and the total number of true negatives and false negatives, thereby providing a better understanding of a model's classification behavior than simply calculating its overall accuracy. As part of this research, confusion matrices were used to compare the performance of five different machine learning models.

4.6.1. Logistic Regression Model

The confusion matrix for the Logistic Regression Model indicated that there were many instances where the model had correctly identified the class label, illustrating good initial performance; however, there were still a considerable number of misclassified cases, illustrating some limitation in the model's ability to capture more complex nonlinear patterns, as seen in Figure 8.

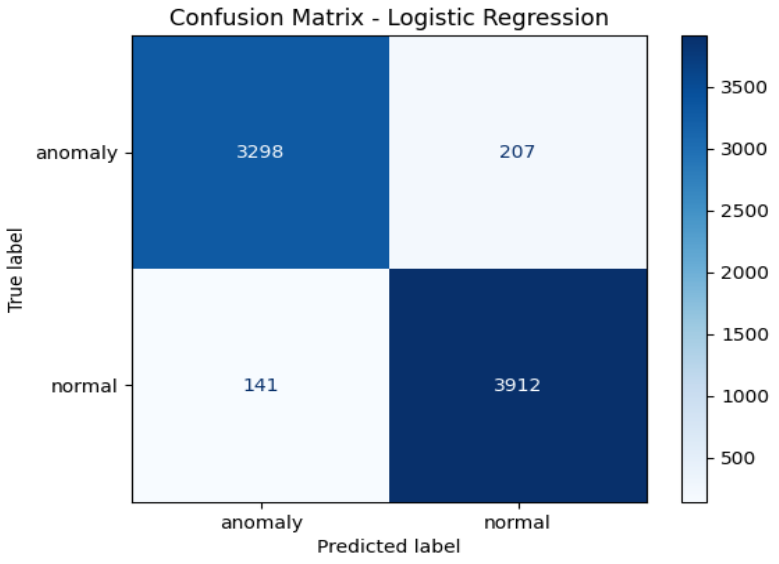

Figure 8: Confusion matrix for logistic regression classifier

4.6.2. Decision Tree Classifier

The confusion matrix for the Decision Tree classifier showed it performed even better than the Logistic Regression classifier at correctly classifying instances, resulting in fewer false positives and false negatives, demonstrating the Decision Tree's ability to learn decision boundaries from the training data (Figure 9).

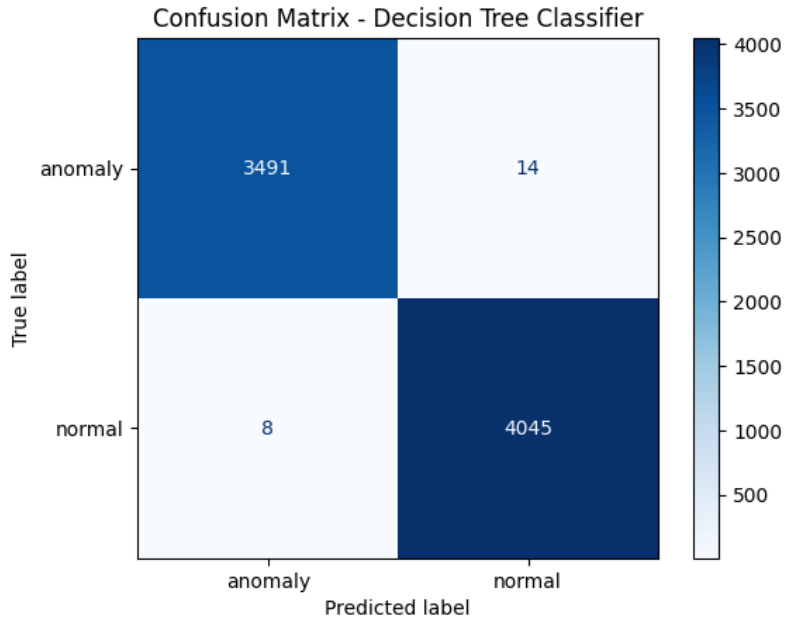

Figure 9: Confusion matrix for the decision tree classifier

4.6.3. Random Forest Classifier

The confusion matrix for the Random Forest classifier showed it achieved the highest performance among the models evaluated, with almost no misclassifications, clearly highlighting the advantages of ensemble learning in improving generalizability and robustness for intrusion detection tasks (Figure 10).

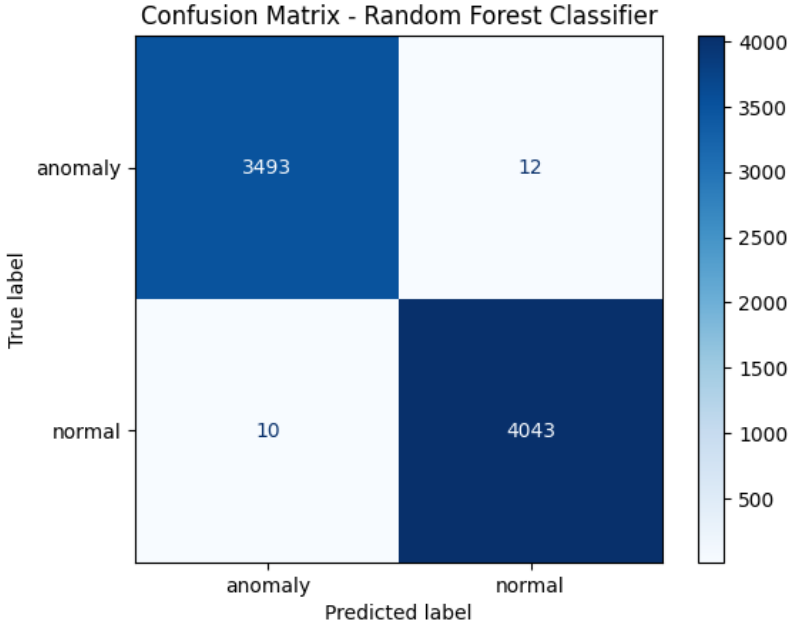

Figure 10: Confusion matrix for random forest classifier

4.6.4. K-Nearest Neighbour Classifier

The K-Nearest Neighbor classifier also demonstrated good performance, albeit at a slight cost of a slightly higher misclassification error relative to ensemble methods, suggesting sensitivity to data distributions and feature scaling, as seen in Figure 11.

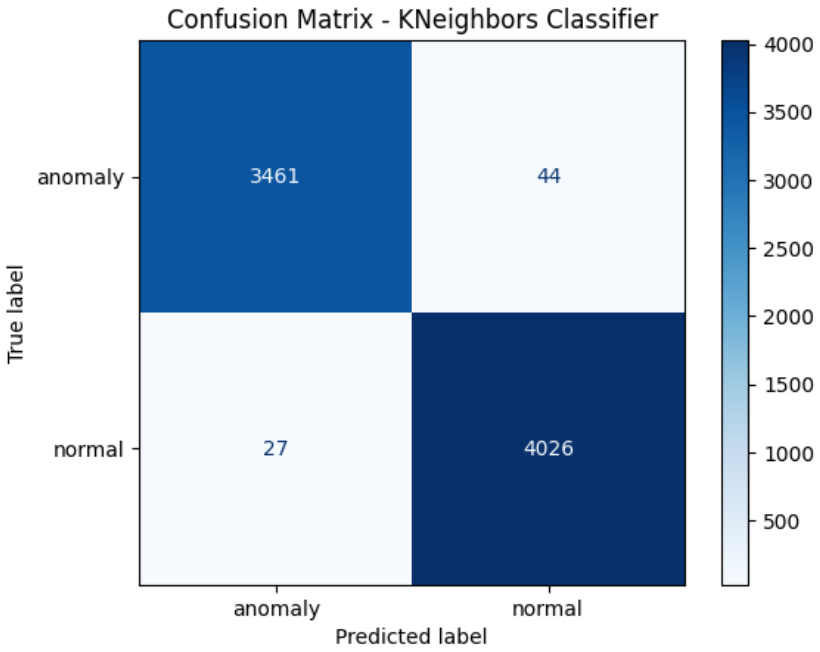

Figure 11: Confusion matrix for the K-nearest neighbors' classifier

4.6.5. Gaussian Naive Bayes model

The confusion matrix for the Gaussian Naive Bayes model showed significantly lower accuracy in identifying anomalous network traffic, with a high number of misclassifications. This poor performance can be attributed to the Naive Bayes model's strong assumption that features are independent, which does not reflect the complex relationships in network traffic data, as shown in Figure 12.

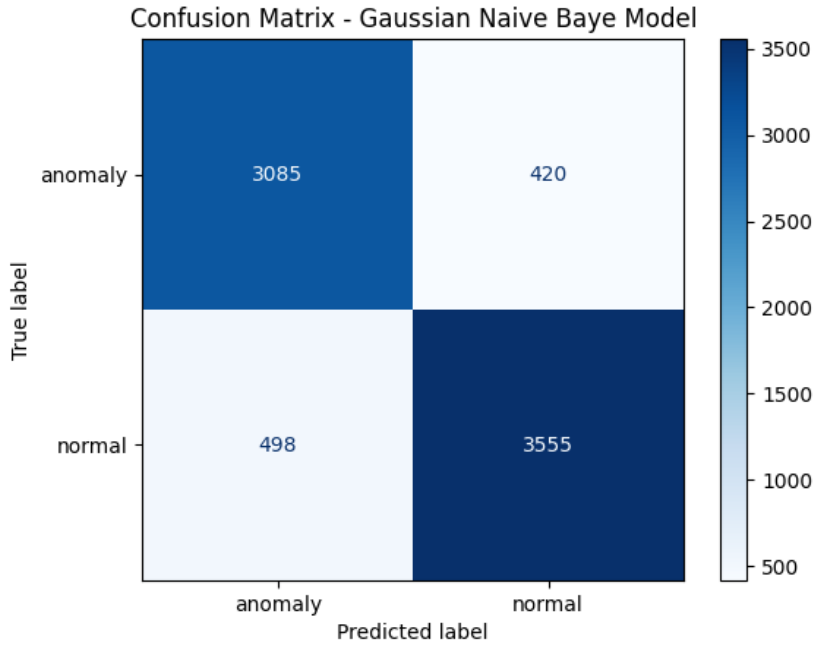

Figure 12: Confusion matrix for Gaussian naive bayes classifier

Overall, the analysis of confusion matrices clearly shows that ensemble-based approaches, especially Random Forests, outperform the others in anomaly detection.

4.7. Web Application Implementation

To enable users to access the proposed Hybrid Intrusion Detection System interactively, a web-based application was created using the Streamlit framework, allowing them to enter network traffic parameters or upload datasets for analysis, as shown in Figure 13.

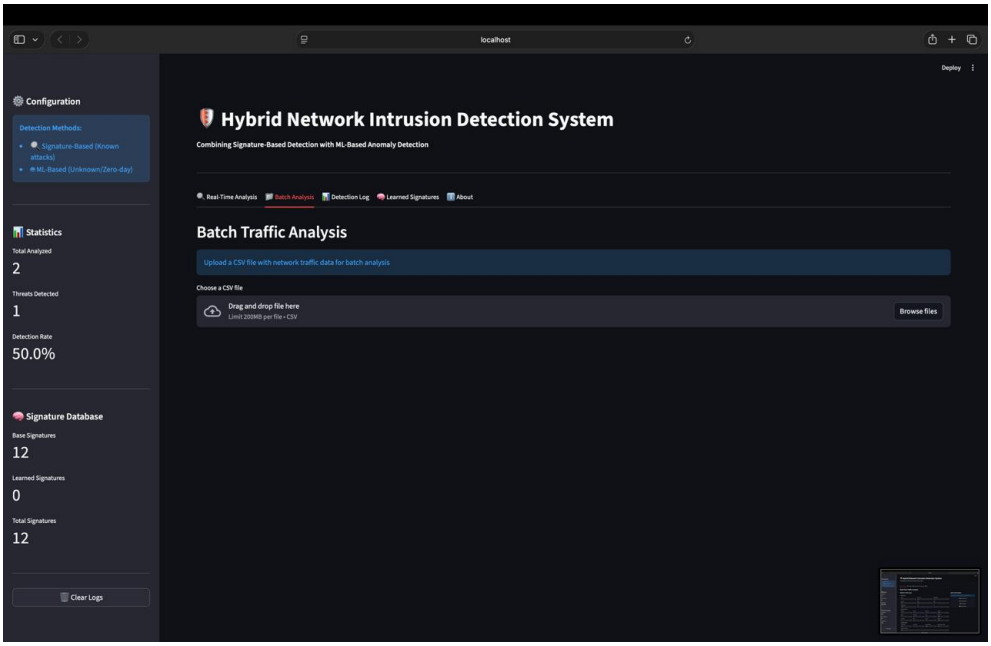

Figure 13: Operational streamlit framework for the hybrid intrusion detection system

Using the Streamlit framework allows users to easily create a web-based interface that facilitates communication between the user interface and underlying machine learning models, while enabling the easy integration of data entry, model inference, and visualization components. Additionally, Streamlit's routing system allows users to define specific paths and application logic for processing user requests, efficiently executing prediction tasks, and providing real-time feedback. In addition to performing prediction tasks, the web application includes visualization dashboards that present detection results, traffic distribution, and attack categories, enhancing the system's interpretability for security analysts, as shown in Figure 14.

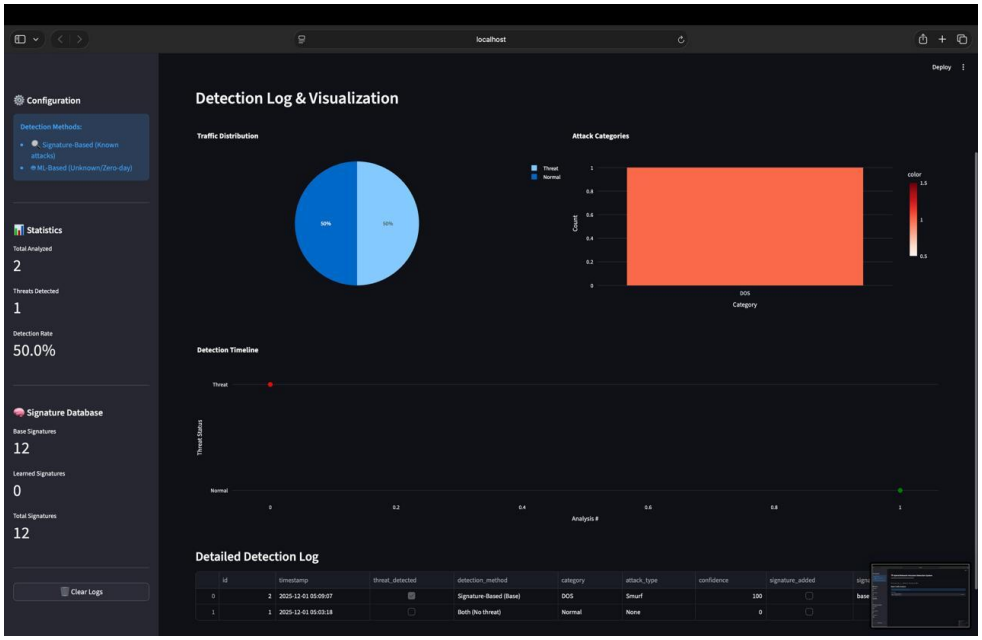

Figure 14: Data representation from the operational hybrid IDS

The proposed Hybrid Intrusion Detection System was successfully integrated into a user-friendly web application built using the Streamlit framework to make the system accessible, usable, and applicable for use by security professionals who require the ability to analyze their network traffic in real time, as well as the ability to run batch evaluations of their network traffic without requiring significant technical expertise. By combining the strengths of signature-based detection with those of machine-learning-based anomaly detection, the system can identify both previously unknown and known attacks. In addition to the ease of use provided by the Streamlit framework, its modularity enables seamless upgrades and additions in the future. As such, the successful deployment of the system demonstrates the potential to integrate advanced machine learning techniques into a user interface accessible to security professionals working on real-world network security problems.

5. Discussion

This research demonstrated that an integrated solution of signature-based detection with machine learning (ML) based Anomaly Detection increases the capacity of an intrusion detection system (IDS) to detect both known and unknown attacks. The Hybrid IDS system integrated a deterministic rule-based signature-based detection engine with a Random Forest Classifier. Further, it included a self-learning mechanism to convert identified anomalies into reusable signatures. Results demonstrate that the Hybrid IDS system has good detection reliability while also providing usability via a web-based interface.

5.1. Explanation of Findings and Their Meaning

Findings demonstrate that the Hybrid Architecture enhances an IDS's overall detection coverage by leveraging the complementary strengths of its components. The signature-based detection component effectively identifies previously known attack patterns with high accuracy. In contrast, the AD component extends an IDS's detection capability to identify anomalous behaviors not captured in the signature repository. Furthermore, incorporating a self-learning mechanism enhances the IDS's adaptability. Anomalies detected by the ML component are converted into signatures with key features extracted and added to the signature repository. This process allows the IDS to evolve adaptively, thereby lessening the reliance on manual signature updates and enhancing its long-term efficacy.

5.2. Comparison to Previous Literature

Past research has clearly established that standalone signature-based detection systems achieve high accuracy in identifying known threats; however, they fail to protect against zero-day attacks. On the other hand, standalone AD systems typically provide poor detection due to high false positive rates. As expected, the results of this research validate previous findings; however, they extend current knowledge by demonstrating that the combination of signature-based detection and AD within a unified framework, coupled with self-learning, mitigates these limitations. This extension of current knowledge addresses a significant gap in literature: many Hybrid IDS models lack practical implementations that support real-time user interaction and persistent learning.

5.3. Theoretical and Practical Implications

From a theoretical standpoint, the results illustrate that the most effective method for protecting against intrusion attacks is through multi-layered security approaches rather than relying on individual techniques. The Hybrid Architecture illustrates a system-level approach to detecting intrusions, where detection efficiency arises from synergistic effects among components rather than from individual algorithmic superiority. Practically speaking, the web-based deployment of the Hybrid IDS demonstrates that advanced detection mechanisms can be delivered through accessible, user-friendly interfaces. The inclusion of real-time monitoring, batch processing, persistent logging, and visual analytics greatly enhances the usability of the Hybrid IDS for network administrators and security analysts.

5.4. Limitations

This section outlines the major limitations of the Hybrid Network Intrusion Detection System, specifically regarding Data, Methodology, Scope, and Resources. The Hybrid NIDS was developed and tested using the KDD99 dataset, which is simulated, outdated, contains duplicate and imbalanced records, and thus does not accurately reflect the modern network environment or the types of attacks currently being conducted. Additionally, the supervised learning approach used to develop the Hybrid NIDS requires labeled data and offline evaluation, limiting its evaluation under real-time performance conditions. Furthermore, the design science approach used to develop Hybrid NIDS focused on artifact development rather than causal-oriented or longitudinal analysis. Additionally, the Hybrid NIDS was evaluated on a single dataset containing predefined attack categories and did not account for additional environmental factors, such as encrypted traffic or adaptive attackers. Finally, potential bias from feature selection and model tuning, and limited time and computational resources for large-scale experimentation and live traffic integration, limit the applicability of the Hybrid NIDS in larger-scale deployments.

5.5. Challenges

This section describes the primary challenges faced in designing, implementing, and evaluating the Hybrid Network Intrusion Detection System. Significant methodological challenges arose during the preprocessing of the KDD99 dataset, including handling duplicate records, class imbalance, complex categorical feature encodings, and maintaining consistent ordering between the training and deployed feature sets. Theoretical challenges existed in determining how to dynamically allocate weights to both signature-based detection and AD components of the Hybrid IDS. Lastly, practical challenges arose due to time and resource limitations for large-scale experimentation, live packet capture integration, and simultaneous development of the web interface. Interpreting the results was challenging, as distinguishing subtle anomalies from noise was difficult. At the same time, ethical concerns arose about ensuring that all stored detection logs contained only synthetic traffic features. All these challenges were addressed through iterative preprocessing of the KDD99 dataset, careful mapping of feature sets, and systematic validation of the Hybrid IDS. This research also made contributions to the field by providing a fully functional Hybrid NIDS that includes machine learning, signature-based detection, self-learning capabilities, and a web-based deployment platform. In addition, the Hybrid NIDS introduced the first practical implementation of an Adaptive IDS that can not only detect intrusion attempts but also update its knowledge about what constitutes an intrusion attempt over time. The Hybrid NIDS, therefore, represents a new contribution beyond the theoretical hybrid detection models commonly found in literature.

6. Conclusion

This research paper developed and evaluated a Hybrid Network Intrusion Detection System (NIDSS) using a Machine Learning Anomaly Detection Engine in conjunction with a Signature-Based Network Intrusion Detection System (NIDSS). The objective of the study was to improve upon traditional IDS solutions. Based on the findings, the Hybrid Network Intrusion Detection System (NIDSS) appears capable of identifying both known and unknown threats, including zero-day attacks. The machine learning anomaly detection engine will detect attacks that have not yet been discovered. At the same time, the SB-NIDSS will identify attacks that have already been discovered using the signature engine. The Hybrid NIDSS includes an internal mechanism that "learns" from anomalous activity and incorporates it into its signature engine, continuously updating the system's knowledge base. Additionally, the Hybrid NIDSS web-based platform enables real-time data monitoring, ease of use, and continuous data logging. Based on the findings, it has been determined that the Hybrid Architecture offers enhanced detection capabilities and flexibility, while providing the same level of use as conventional intrusion detection systems (IDSs) through a web-based interface. In addition, the findings indicate that the Hybrid Architecture offers superior detection coverage in comparison to conventional intrusion detection systems. Based on the outcomes of this research, this paper offers actionable, prioritized recommendations for future investigations.

Acknowledgment: The author would like to express sincere gratitude to Northumbria University for providing the academic environment, resources, and support necessary for the successful completion of this research.

Data Availability Statement: The data associated with this study are available from the corresponding author and may be provided upon reasonable request.

Funding Statement: This research and the preparation of this manuscript were completed without receiving any form of financial support or funding.

Conflicts of Interest Statement: The author declares that no conflicts of interest exist in relation to the publication of this research article.

Ethics and Consent Statement: The research was conducted in accordance with ethical standards, and informed consent was obtained from all participants.

References

1. U. Ahmed, M. Nazir, A. Sarwar, T. Ali, E. H. M. Aggoune, T. Shahzad, and M. A. Khan, “Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering,” *Scientific Reports,* vol. 15, no. 1, pp. 1–33, 2025.
2. G. Alasad, F. Zawaideh, F. H. Zawaideh, G. Al-Asad, G. Swaneh, S. Batainah, and H. Bakkar, “Intrusion detection system for (IoT) networks using convolutional neural network (CNN) and XGBoost algorithm,” *Journal of Theoretical and Applied Information Technology*, vol. 102, no. 4, pp. 1750–1759, 2024.
3. A. Aljumah, “IoT-based intrusion detection system using convolution neural networks,” *PeerJ Computer Science*, vol. 7, no. 9, pp. 1–19, 2021.

4. A. Almotairi, S. Atawneh, O. A. Khashan, and N. M. Khafajah, "Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models," *Systems Science and Control Engineering*, vol. 12, no. 1, pp. 1–18, 2024.
5. E. Alotaibi, R. B. Sulaiman, and M. Almaiah, "Assessment of cybersecurity threats and defense mechanisms in wireless sensor networks," *Journal of Cyber Security and Risk Auditing,* vol. 2025, no. 1, pp. 47–59, 2025.
6. A. T. Azar, E. Shehab, A. M. Mattar, I. A. Hameed, and S. A. Elsaid, "Deep learning-based hybrid intrusion detection systems to protect satellite networks," *Journal of Network and Systems Management,* vol. 31, no. 9, pp. 1–31, 2023.
7. D. Bhavani and N. Mangla, "A novel network intrusion detection system based on semi-supervised approach for IoT," *International Journal of Advanced Computer Science and Applications (IJACSA),* vol. 14, no. 4, pp. 207–216, 2023.
8. M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," *Discover Internet of Things,* vol. 3, no. 5, pp. 1–23, 2023.
9. B. I. Farhan and A. D. Jasim, "Survey of intrusion detection using deep learning in the Internet of Things," *Iraqi Journal for Computer Science and Mathematics,* vol. 3, no. 1, pp. 83–93, 2022.
10. M. A. Hossain, "Deep learning-based intrusion detection for IoT networks: A scalable and efficient approach," *EURASIP Journal on Information Security*, vol. 2025, no. 9, pp. 1–23, 2025.
11. O. A. Hussain, Z. Chen, and H. Zhu, "Secure Net: A hybrid CNN-LSTM-based intrusion detection system for securing IoT networks," in *Proc. 4th Int. Conf. Comput., Artif. Intell. Control Eng. (CAICE),* Hefei, China, 2025.
12. M. Idris, S. M. Yasin, and G. H. Audi, "Systematic review of intrusion detection system based on machine learning techniques for Internet of Things," *Nigerian Journal of Computing, Engineering and Technology (NIJOCET),* vol. 2, no. 2, pp. 1–16, 2023.
13. A. J. A. Immastephy and K. Punitha, "A systematic review on network intrusion detection system based on machine learning and deep learning approach," *E3S Web of Conferences,* vol. 540, no. 6, pp. 1–15, 2024.
14. M. Janati and F. Messaoudi, "Intrusion detection system-based network behavior analysis: A systemic literature review," *International Journal of Advanced Computer Science and Applications (IJACSA),* vol. 16, no. 3, pp. 793–802, 2025.
15. H. Kamal and M. Mashaly, "Robust intrusion detection system using an improved hybrid deep learning model for binary and multi-class classification in IoT networks," *Technologies*, vol. 13, no. 3, pp. 1–71, 2025.
16. A. R. Khan, M. Kashif, R. H. Jhaveri, R. Raut, T. Saba, and S. A. Bahaj, "Deep learning for intrusion detection and security of Internet of Things (IoT): Current analysis, challenges, and possible solutions," *Security and Communication Networks,* vol. 2022, no. 1, pp. 1–13, 2022.
17. N. W. Khan, M. S. Alshehri, M. A. Khan, S. Almakdi, N. Moradpoor, A. Alazeb, S. Ullah, N. Naz, and J. Ahmad, "A hybrid deep learning-based intrusion detection system for IoT networks," *Mathematical Biosciences and Engineering*, vol. 20, no. 8, pp. 13491–13520, 2023.
18. A. Kruti, U. Butt, and R. B. Sulaiman, "A review of SolarWinds attack on Orion platform using persistent threat agents and techniques for gaining unauthorized access," *arXiv preprint arXiv:2308.10294*, 2023. [Accessed by 12/03/2025].
19. A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the Internet of Things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 3, pp. 1–27, 2021.
20. D. A. Laila, M. Aljawarneh, Q. Al-Na'amneh, and R. B. Sulaiman, "Optimizing intrusion detection systems through benchmarking of ensemble classifiers on diverse network attacks," *STAP Journal of Security Risk Management,* vol. 2025, no. 1, pp. 71–84, 2025.
21. R. B. Sulaiman and A. Khraisat, "Metaheuristic-driven feature selection with SVM and KNN for robust DDoS attack detection: A comparative study," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 4, pp. 182–203, 2025.
22. B. Nawaal, U. Haider, I. U. Khan, and M. Fayaz, "Signature-based intrusion detection system for IoT," in Cyber Security for Next-Generation Computing Technologies, *CRC Press*, Florida, United States of America, 2024.
23. M. Schrötter, A. Niemann, and B. Schnor, "A comparison of neural-network-based intrusion detection against signature-based detection in IoT networks, *information,* vol. 15, no. 3, pp. 1–26, 2024.
24. S. Kakolu, M. A. Faheem, and M. Aslam, "AI-enabled intrusion detection systems in IoT networks: Advancing defense mechanisms for resource-constrained devices," *International Journal of Science and Research Archive,* vol. 9, no. 1, pp. 752–769, 2023.
25. Y. M. Yang, K. C. Chang, and J. N. Luo, "Hybrid neural network-based intrusion detection system: Leveraging LightGBM and MobileNetV2 for IoT security," *Symmetry*, vol. 17, no. 3, pp. 1–19, 2025.
26. T. Jamal, "KDD99 dataset," Kaggle, 2021. [Accessed by 23/03/2025].