

Enhanced Machine Learning-Based Intrusion Detection System for Zero-Day Attacks

¹ UE-Bee, Kakuubari Promise, ²Sako, D.J.S, ³Nwiabu, N. D

Department of Computer Science,
Rivers State University, Port Harcourt, Nigeria

¹kakuubari.ue-bee@rsu.edu.ng,

²dum.sako@ust.edu.ng, ³nwiabu.nuka@ust.edu.ng

<https://doi.org/10.56201/ijcsmt.vol.12.no10.2026.pg13.25>

Abstract

The recent rise in complex cyber-attacks has led to the need for intrusion detection systems that can detect both known and previously unseen attacks in network traffic with minimal false positives. This paper presents Enhanced Machine Learning Based Intrusion Detection System for Zero-Day Network Attacks as a solution to the problem of zero-day attack detection with minimal false positives. The proposed model uses a hybrid detection architecture that comprises of Signature-Based Detection, Random Forest classifier, Confidence Gate, and MiniBatch K-Means clustering to detect both known and zero-day attacks respectively. The intrusion detection system was designed and implemented using Python and CICIDS2017 dataset was used for training and evaluating the proposed models. The experimental results showed that Random Forest attained an accuracy of 99.59 % with a precision of 99.58%, a recall of 99.59%, and an F1-score of 99.58%. On the other hand, MiniBatch K-Means attained a Silhouette Score of 0.8989%. This demonstrated that that signature-based detection rules coupled with supervised and unsupervised machine learning algorithms could effectively detect known attacks and suspicious activities. The prototype was able to monitor live network traffic and present dashboards with visualizations of security-related events.

Keywords: Intrusion detection system, zero-day attack, Random Forest, MiniBatch K-Means, hybrid detection, Security Operations Center; MITRE ATT&CK

1. Introduction

The adoption of emerging technologies such as cloud computing, the Internet of Things (IoT), Artificial Intelligence (AI), and high-speed Internet connectivity has tremendously increased the volume and complexity of traffic flowing through modern networks. The increasing number of connected devices, applications, and services has also expanded the potential attack surface available to cybercriminals, making continuous monitoring and identification of malicious network activities increasingly challenging (Pinto et al., 2023). The same networks are vulnerable to attacks by cybercriminals, whose methods have increased in both complexity and volume. Organizations have consequently adopted several security mechanisms, including firewalls, antivirus software, and Intrusion Detection Systems (IDSs), to reduce the impact of cyber-attacks and protect network resources from unauthorized activities (Guo et al., 2023). Most existing IDSs use rule-based detection mechanisms such as the traditional signature-based detection to detect

and classify network traffic. Signature detection compares observed traffic against predefined patterns associated with previously identified attacks. Although this approach is useful for detecting known attacks, it depends on the availability of previously defined signatures and is therefore limited when an attack has not previously been identified (Talaie Khoei & Kaabouch, 2023). The Random Forest model optimizes signature-based detection. RF is supervised machine learning ensemble algorithm that builds multiple decision trees on randomized subsets of data and features, combining their outputs (via majority voting) to classify patterns or anomalies (). RF algorithm addresses limitation associated with the rule-based system by identifying deviations from established traffic behaviour (Chen et al., 2024). RF has high accuracy on structured and complex traffic data; less prone to overfitting than single decision trees; handles high-dimensional data well (). However, it requires a large, clean, labeled dataset and struggles to classify brand-new attack types it has never seen during training. Also, not every deviation from normal network behaviour represents a malicious activity. Legitimate network activities can sometimes appear unusual and may consequently be classified as malicious. This can generate excessive false-positive alerts, increase the workload of security analysts, who must investigate numerous alerts that may not represent genuine threats and ultimately reduce the reliability of the detection system. (Guo et al., 2023; Sajid et al., 2024). Mini-Batch K-Means, an unsupervised learning algorithm, is It is extremely fast and memory-efficient and optimized for fast anomaly detection in massive, real-time high-speed streaming network traffic where labels do not exist () making it effective in the detection of zero-day attacks. This study developed an enhanced hybrid IDS that combines Signature-Based Detection, Random Forest classification, a 99.9% Confidence Gate, and MiniBatch K-Means anomaly detection for zero-day attacks. The system extends model evaluation beyond an offline dataset by integrating live packet capture, flow generation, feature processing, SQLite logging, MITRE ATT&CK mapping, and Security Operations Center (SOC) dashboard for live monitoring and visualization were employed during the prototype development.

2. Related Work.

Zhen Dai et al. (2024) proposed a machine learning-based intrusion detection model for detecting zero-day attacks in previously unseen traffic. The study implements a combination of an Autoencoder and a Random Forest classifier for attack categorization using the CIC-MalMem-2022. The proposed model was evaluated using Accuracy, Precision, Recall, F1-score, and Matthew's Correlation Coefficient. The results indicate that the model demonstrates excellent detection performance and is capable of detecting previously unseen attacks, thus proving the efficacy of using both supervised and unsupervised learning in detecting zero-day attacks. However, the study fails to consider the implementation of real-time packet capture, analysis, and operational visibility in a Security Operations Center (SOC) through a dashboard. Thus there is a need to develop an intrusion detection system that is able to process and analyze traffic in real-time while providing operational visibility into the detected attacks. Chen et al. (2024) proposed a hybrid framework for detecting complex cyber-attacks by combining the Network-Based Intrusion Detection System (NIDS) and the Host-Based Intrusion Detection System (HIDS). The study explores the effectiveness of using both host and network-level monitoring and implements machine learning algorithms in the development of the framework. The results indicate that the hybrid framework outperforms the traditional NIDS and HIDS approaches by providing enhanced visibility and protection against sophisticated cyber-attacks. However, the proposed model complicates the detection system by introducing an additional layer of complexity in the detection process due to the simultaneous analysis of network and host-level traffic. The study also fails to

develop an anomaly detection model for detecting zero-day attacks. Thus there is a need to develop an efficient hybrid intrusion detection framework that is able to offer real-time network traffic monitoring while being equipped with an anomaly detection model for detecting zero-day attacks. Guo et al. (2023) present a survey of machine learning-based techniques for zero-day attack detection. The study reviews and evaluates the efficacy of both unsupervised, supervised, and hybrid machine learning techniques for zero-day attack detection. The study concludes that hybrid machine learning techniques offer the highest level of accuracy in detecting zero-day attacks due to the combination of different detection mechanisms. However, the survey highlights some of the limitations of using machine learning in zero-day attack detection, including high false-positive rates, lack of real-time analysis capabilities, and limited success in detecting sophisticated attacks. The study recommends the development of an intelligent hybrid intrusion detection system that is capable of combining different detection mechanisms while maintaining a high level of accuracy in real-time operation. Khan et al. (2023) proposed an ensemble-based framework for intrusion detection by combining the Random Forest, XGBoost, and Neural Network classification algorithms for improved accuracy in detecting both known and unknown attacks. The model was evaluated using standard intrusion detection benchmark datasets and demonstrated enhanced attack detection performance due to the combination of different classification algorithms. However, the study increases the complexity of the intrusion detection framework by implementing different machine learning algorithms simultaneously, thus limiting its applicability in real-time network traffic analysis due to the high computational power required. Thus there is a need to develop a hybrid intrusion detection framework that is efficient and capable of operating in real-time while maintaining a high level of accuracy in attack detection.

3. Proposed System

The system design describes how the proposed system will capture the network packets and flows, process the packets and flows features, detect cyber-attacks, store the detected attacks in a database, and display detection results, detection statistics, and history through the Security Operations Center dashboard. The proposed system architecture is presented in Figure 1. The proposed architecture is sequential and conditional rather than a set of independent parallel classifiers. It consists of the following components: packet capture, data preprocessing, feature extraction, selection and scaling, hybrid detection engine, SQLite database, and a security operation center (SOC) dashboard for real-time detection of known and potential zero-day network attacks. Traffic data are captured, preprocessed and the features extracted, selected, and scaled before entering the hybrid detection engine, the core component of the proposed intrusion detection system. The hybrid engine implements complementary detection responsibilities and uses a hybrid approach to detect normal and intrusive activities in network data by combining the conventional signature-based method and the machine learning-based methods (Random Forest and Minibatch K-Means algorithms) to accurately detects both known and zero-day attacks.



Figure 1. Architectural design of the proposed System.

Signature rules identify known attack patterns by inspecting network packets or system logs and comparing them against a database of known threat signatures. While it is extremely accurate and low overhead for known attacks and has very low false-positive rate, it is completely blind to zero-day attacks or polymorphic variants that lack a pre-existing signature. It requires constant signature updates. Random Forest performs supervised classification of traffic represented by the labelled training data. The Confidence Gate reduces unnecessary anomaly processing by allowing high-confidence benign predictions to terminate as normal traffic. MiniBatch K-Means is reserved for uncertain benign-labelled traffic and provides an unsupervised mechanism for identifying traffic that is substantially different from learned behaviour. It is a scalable variant of the standard K-Means clustering algorithm that uses small, random subsets (mini-batches) of data to iteratively update cluster centroids rather than loading the entire dataset into memory. It groups similar network flows together and flags data points that fall far from any normal cluster centroid as anomalies. As shown in Figure 2, signature matching is performed first. Unmatched traffic is classified by Random Forest. Traffic predicted as benign is evaluated by a 99.9% Confidence Gate; high-confidence benign traffic is accepted as normal, whereas lower-confidence traffic is escalated to MiniBatch K-Means for anomaly analysis. Final decisions are logged and exposed through the Security Operations Center (SOC) monitoring layer/dashboard.

Given that x represents extracted network flow feature vector, $M(x)$ is the signature matching function, $RF(x)$ represents a Random Forest prediction, $A(x)$ is the MiniBatch K-Means anomaly score and θ represents the anomaly threshold used to distinguish normal and anomalous traffic, the Hybrid Detection Engine is represented as:

$$\begin{cases} \text{Known Attack,} & M(x) = 1 \\ \text{Known Attack,} & M(x) = 0 \wedge RF(x) \neq \text{Benign} \\ \text{Normal Traffic} & M(x) = 0 \wedge RF(x) = \text{Benign} \wedge A(x) \leq \theta \\ \text{Potential Zero - Day Attack,} & M(x) = 0 \wedge RF(x) = \text{Benign} \wedge A(x) > \theta \end{cases} \quad (1)$$

Where $M(x) = 1$ indicates that the network flow matches a predefined attack signature.

The final security decision generated by the Hybrid Detection Engine is

$$R \in \{\text{Normal Traffic, Known Attack, Potential Zero-Day Attack}\}. \quad (2)$$

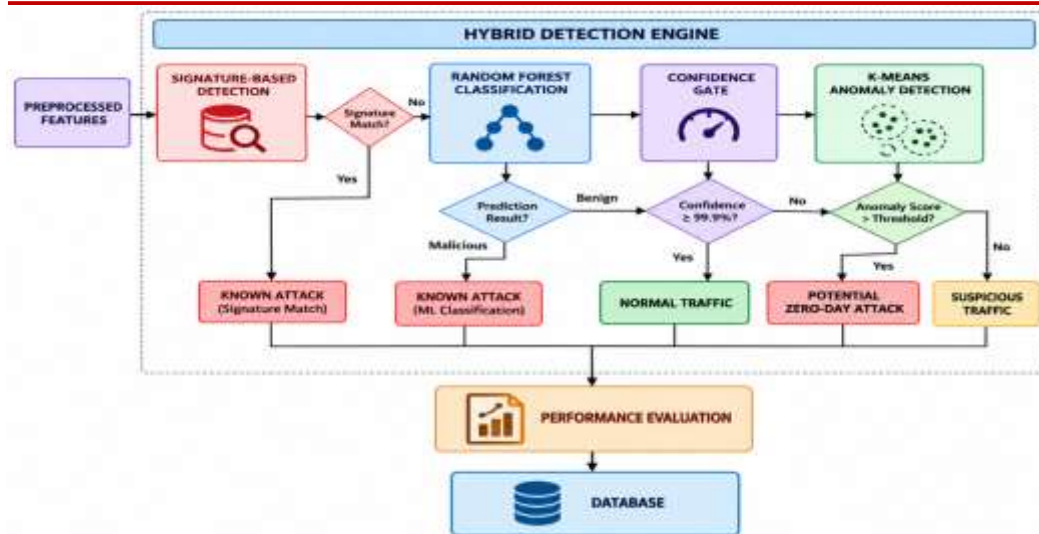


Figure 2. Hybrid detection engine and decision flow of the proposed system.

4. Experiment and Results

4.1 Experimental Setup

The system was implemented in Python 3.10 to ensure compatibility with the chosen machine-learning and web frameworks, and its extensive ecosystem of machine learning libraries, including PyTorch, and Scikit-learn, Pandas, Numpy, TensorFlow and Keras, Streamlit, Matplotlib, etc. The proposed intrusion detection system was evaluated in real-time in a controlled laboratory environment consisting of three (3) virtual machines set up in a local area network (LAN) using the VirtualBox Host-Only network. The Ubuntu Linux machine (IP Address: 192.168.76.7) hosted the proposed Enhanced Hybrid Intrusion Detection System and served as the monitoring server. The Kali Linux machine (IP Address: 192.168.76.3) was used as the intruding host, and the Windows machine (IP Address: 192.168.76.6) was used as the legitimate network user.

3.1 Dataset and Feature Preparation

The Canadian Institute for Cybersecurity Intrusion Detection System 2017 (CICIDS2017) dataset was used for model development and evaluation. The cleaned dataset contained 2,520,798 rows and 79 columns in the implementation workflow with an attack of 15. The work describes 78 network-flow attributes and selects 20 features for model training and live inference. The selected variables represent flow duration, packet counts, byte and packet rates, packet-length statistics, inter-arrival-time statistics, header lengths, average segment size, protocol, destination port, and related forward/backward flow characteristics. The dataset was split into 80%/20% for training and testing. Table 1 is a sample of the CICIDS2017 Dataset. Data preprocessing included cleaning invalid values, encoding categorical information where required, selecting the 20 operational features, and applying feature scaling. The same preprocessing logic and saved scaler were used for live traffic so that real-time feature vectors were compatible with the trained models.

Table 1: Sample of the CICIDS2017 Dataset (Sharafaldin, et al., 2018).

Flow Duration	Total Fwd Packets	Total Bwd Packets	Flow Bytes/s	Packet Length Mean	Flow IAT Mean	Protocol	Destination Port	Label
112345	12	10	2543.61	425.67	1023.45	TCP	80	BENIGN
54321	20	15	4987.20	612.14	543.22	TCP	443	PortScan
87456	18	14	3765.89	501.38	812.77	UDP	53	DDoS
45218	8	6	1824.55	298.44	1345.19	TCP	22	SSH-Patator
98654	25	19	6214.87	710.92	456.81	TCP	21	FTP-Patator

4.3 Performance Evaluation Metrics

Classification performance was evaluated with Accuracy, Precision, Recall, F1-score, False-Negative Rate, and False-Positive Rate. TP denotes malicious traffic correctly identified, TN denotes normal traffic correctly identified, FP denotes normal traffic incorrectly classified as malicious, and FN denotes malicious traffic incorrectly classified as normal.

$$(i) \quad \text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

(3)

$$(ii) \quad \text{Precision} = TP / (TP + FP)$$

(4)

$$(iii) \quad \text{Recall} = TP / (TP + FN)$$

(5)

$$(iv) \quad \text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

(6)

$$(v) \quad \text{FNR} = FN / (FN + TP)$$

(7)

$$(vi) \quad \text{FPR} = FP / (FP + TN) \times 100$$

(8)

$$(vii) \quad \text{Relative FPR Reduction} = (\text{FPR}_{\text{initial}} - \text{FPR}_{\text{refined}}) / \text{FPR}_{\text{initial}} \times 100$$

(9)

MiniBatch K-Means clustering quality was evaluated using the Silhouette Score, Davies-Bouldin Index (DB), and Calinski-Harabasz Index (CH)..

The Silhouette Score, $s(i)$, measures cohesion and separation; values closer to 1 indicate stronger separation. It is calculated as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

(10)

Where $a(i)$ is the cohesion (the average distance from point i to all other points in its own cluster; lower is better) and $b(i)$ is the separation (the average distance from point i to all points in the nearest neighboring cluster; higher is better).

For k clusters, the Davies-Bouldin Index is calculated as:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left(\frac{\bar{d}_i + \bar{d}_j}{d_{i,j}} \right)$$

(11)

where $\bar{d}_i$ is the average distance of points in cluster i to its centroid, $\bar{d}_j$ is the average distance of points in cluster j to its centroid and $d_{i,j}$ is the distance between the centroid of cluster i and cluster j . Lower Davies-Bouldin values indicate better-separated clusters.

Calinski-Harabasz index, CH , is calculated as:

$$CH = \left[\frac{BCSS}{k-1} \right] \div \left[\frac{WCSS}{n-k} \right] \quad (12)$$

where $BCSS$ (Between-Cluster sum of Squares) measures how far apart individual cluster centers are from the global dataset centroid (Higher is better), $WCSS$ (Within-Cluster Sum of Squares) measures how tightly packed or compact the data points are around their respective cluster centroids (Lower is better), k is the number of clusters and n is the total number of data points. Higher Calinski-Harabasz values indicate a stronger ratio of between-cluster to within-cluster dispersion.

4.2 Results and Analysis

We present the results and discuss the performance of the system developed in this study. The performance of the Signature-Based Detection Engine was evaluated using different attack scenarios. The goal of the experiments was to observe how many attacks would be identified by the detection rules and subsequently dropped by the network monitoring application before being processed by the machine learning algorithms. The following attacks were used in the experiments: SYN Scan, Port Scan, SYN Flood, ICMP Flood, UDP Flood, SSH-Patator, FTP-Patator, SQL Injection, Cross-Site Scripting (XSS), and Directory Traversal. The detection performance of the Signature-Based Detection Engine is presented in Table 2.

All the detected attacks were correctly classified as known ones, and their corresponding MITRE ATT&CK tactics were defined. The signature engine detected all ten supported attack scenarios evaluated in the controlled signature test: SYN Scan, Port Scan, SYN Flood, ICMP Flood, UDP Flood, SSH-Patator, FTP-Patator, SQL Injection, Cross-Site Scripting, and Directory Traversal. Their corresponding MITRE ATT&CK tactics were also defined. This result should be interpreted as 100% detection of the supported signature-based test scenarios, not as universal detection of all possible attacks. Random Forest achieved 99.59% accuracy, 99.58% precision, 99.59% recall, and 99.58% F1-score on the reported test set as presented in Table 3. The class-wise performance of the Random Forest classifier is presented in Table 4. The Random Forest algorithm has shown excellent performance for most of the classes, including BENIGN, DDoS, DoS GoldenEye, FTP-Patator, PortScan, and Heartbleed. On the other hand, the classes with minority instances, including Web Attack – SQL Injection and Web Attack – XSS, were poorly predicted, probably because of the lack of training samples. Overall, the weighted average of the Random Forest classifier shows good performance in detecting cyber-attacks.

The MiniBatch K-Means model performance was evaluated to identify the best anomaly detection model for predicting zero-day attacks. Unlike the Random Forest classifier, the MiniBatch K-Means algorithm was applied unsupervised to group normal network traffic observations. The performance of the MiniBatch K-Means was thus measured using the Silhouette Score, Davies–Bouldin Index, and Calinski–Harabasz Index. Table 5 shows the clustering performance of the MiniBatch K-Means model. The Silhouette score was 0.8989, which means there was a significant separation between the clusters formed by the algorithm. This implies that the network traffic behaviour in the intrusion detection system could be divided into groups with dissimilar characteristics. The Davies–Bouldin index was 0.5812, while the Calinski–Harabasz index was 30,201.4585, which also points to a good clustering outcome. Moreover, MiniBatch K-Means required only 0.59 seconds of training time to process the dataset, which makes it a good candidate for real-time data processing in an intrusion detection system. Thus, it would be possible to detect anomalous traffic patterns in time to mitigate possible zero-day attacks.

Together, these results indicate well-separated clusters in the evaluated feature space and support the use of MiniBatch K-Means as the anomaly-analysis stage of the hybrid pipeline. For the initial legitimate-traffic experiment, 4 false-positive alerts were observed among 500 legitimate records. Therefore:

$$\text{FPR}_{\text{initial}} = (4 / 500) \times 100 = 0.80\%$$

After refinement, 2 false-positive alerts were observed among 929 legitimate records:

$$\text{FPR}_{\text{refined}} = (2 / 929) \times 100 \approx 0.22\%$$

The observed relative reduction is: $((0.80 - 0.22) / 0.80) \times 100 = 72.50\%$

This is presented in Table 6.

Table 2. Performance of the Signature-Based Detection Engine

Attack Type	Detection Status	Detection Source	Security Decision	MITRE ATT&CK Tactic
SYN Scan	Detected	Signature	Known Attack	TA0043 – Reconnaissance
Port Scan	Detected	Signature	Known Attack	TA0043 – Reconnaissance
SYN Flood	Detected	Signature	Known Attack	TA0040 – Impact
ICMP Flood	Detected	Signature	Known Attack	TA0040 – Impact
UDP Flood	Detected	Signature	Known Attack	TA0040 – Impact
SSH-Patator	Detected	Signature	Known Attack	TA0006 – Credential Access
FTP-Patator	Detected	Signature	Known Attack	TA0006 – Credential Access
SQL Injection	Detected	Signature	Known Attack	TA0001 – Initial Access
Cross-Site Scripting (XSS)	Detected	Signature	Known Attack	TA0001 – Initial Access
Directory Traversal	Detected	Signature	Known Attack	TA0001 – Initial Access

Table 3. Random Forest Classifier Performance

Performance Metrics	Result
Accuracy	99.59%
Precision	99.58%
Recall	99.59%
F1-Score	99.58%

Table 4: Class-wise Classification Report of the Random Forest Classifier

Traffic Class	Precision	Recall	F1-Score	Support
BENIGN	1.00	1.00	1.00	419,012
Bot	0.81	0.47	0.60	390
DDoS	1.00	1.00	1.00	25,603
DoS Hulk	1.00	0.75	0.86	2,057
DoS GoldenEye	1.00	0.99	0.99	34,569
DoS Slowloris	0.96	1.00	0.98	1,046
DoS SlowHTTPTest	0.99	0.99	0.99	1,077

Traffic Class	Precision	Recall	F1-Score	Support
FTP-Patator	1.00	0.99	0.99	1,186
Heartbleed	1.00	1.00	1.00	2
Infiltration	1.00	1.00	1.00	7
PortScan	0.98	1.00	0.99	18,139
SSH-Patator	1.00	0.92	0.96	644
Web Attack – Brute Force	0.70	0.65	0.67	294
Web Attack – SQL Injection	0.00	0.00	0.00	4
Web Attack – XSS	0.34	0.28	0.31	130

Table 5. MiniBatch K-Means Anomaly Detection Performance

Performance Metrics	Result
Silhouette Score	0.8989
Davies–Bouldin Index	0.5812
Calinski–Harabasz Index	30,201.4585

Table 6. False-Positive Evaluation

Experimental Stage	Legitimate Records	False-Positive Alerts	False-Positive Rate (%)
Initial Experiment	500	4	0.80
Refined Experiment	929	2	0.22

A prototype application was implemented to allow users to monitor live network traffic, detect intrusion events, visualize security-related data, and manage the intrusion detection process. The application incorporated Hybrid Detection Engine and Security Operations Center (SOC) dashboard to enable real-time monitoring, attack visualization, health tracking, and reporting features. Figure 3 shows the Machine-learning pipeline interface of the proposed intrusion detection system. Figure 4 represents Real-Time Security Operations Center Monitoring Dashboard. It displays real-time information including total detections, known attacks, zero-day alerts, critical alerts, live packet and flow counts, Random Forest confidence, and MiniBatch K-Means anomaly distance. This provides visual evidence of the system's real-time monitoring and detection capability.

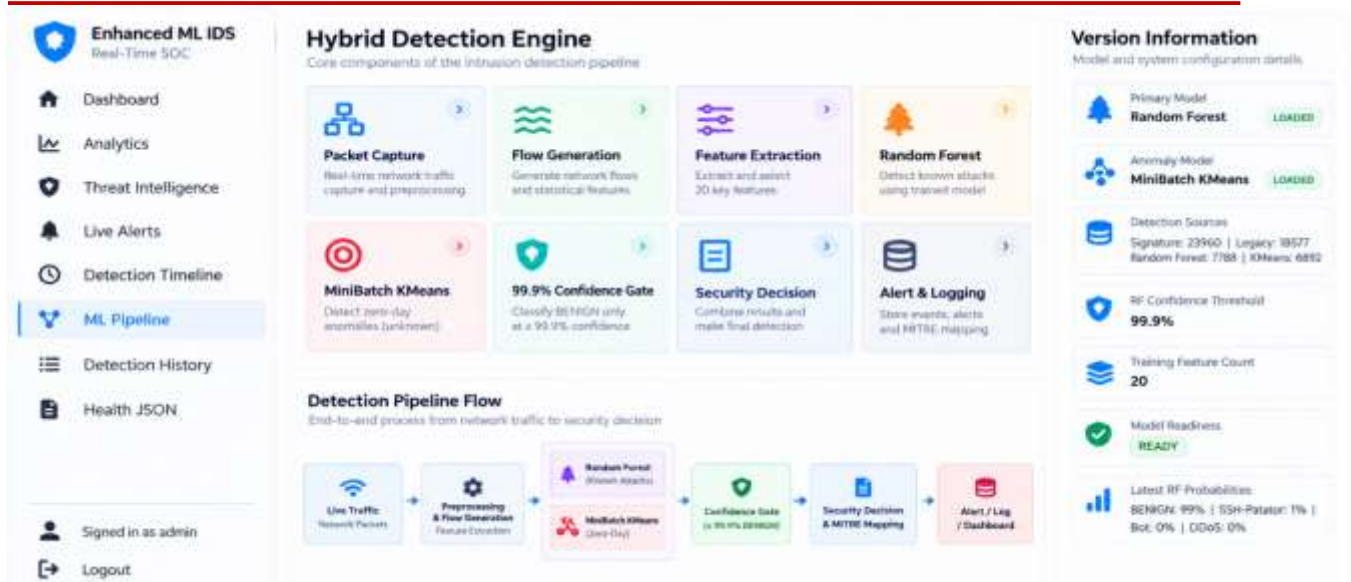


Figure 3. Machine-learning pipeline interface of the proposed intrusion detection system.

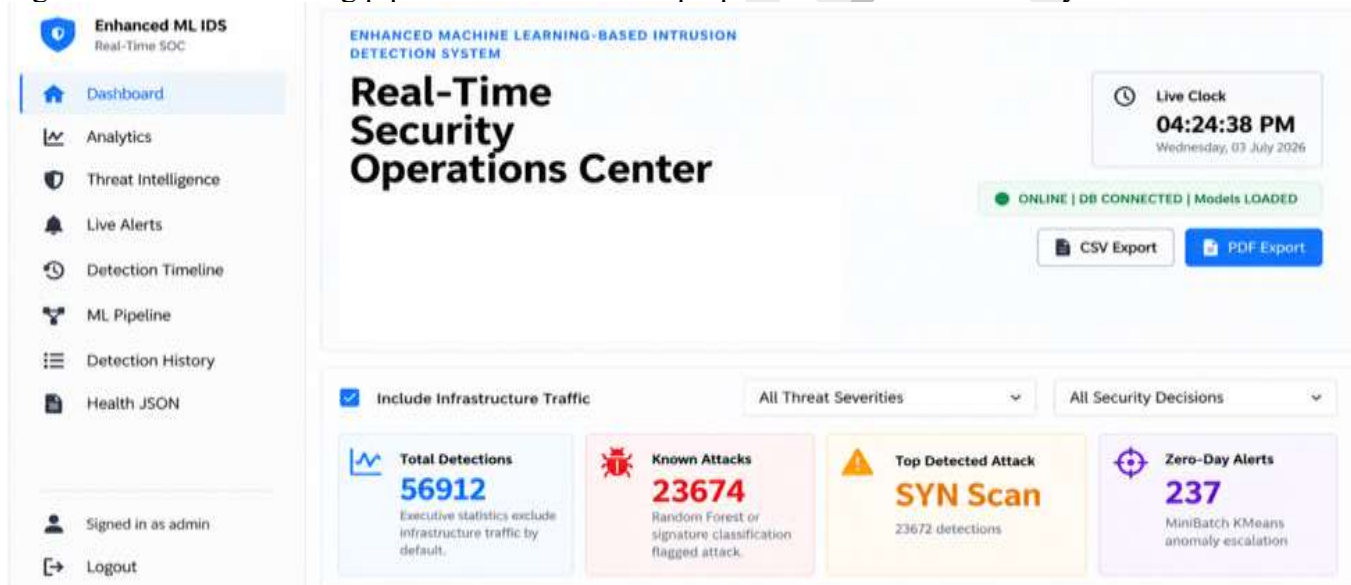


Figure 4: Real-Time Security Operations Center Monitoring Dashboard.

These results show that the supervised and unsupervised components serve different purposes within the hybrid architecture. Random Forest provides highly accurate classification for labelled traffic categories represented in the training data, while MiniBatch K-Means provides a separate anomaly-analysis stage for uncertain traffic. The Confidence Gate prevents every benign Random Forest prediction from being passed to the clustering model, thereby limiting unnecessary anomaly processing. The false-positive experiment is particularly important for operational IDS use. A system that raises excessive alerts can burden analysts even when its overall classification accuracy is high. The refinement from the initial experiment to the later legitimate-traffic test shows that detection rules and decision thresholds materially affect operational alert quality.

The live-network experiment also clarifies the scope of the zero-day claim. CICIDS2017 supplies labelled historical attack traffic for model development, whereas the live experiment demonstrates

the system's ability to process dynamic traffic and escalate anomalous patterns. Accordingly, the most defensible interpretation is that the system provides a mechanism for identifying traffic characteristic of potential zero-day or previously unseen attacks, rather than claiming verified detection of a previously unknown vulnerability exploit.

5. Conclusion

This study designed, implemented, and evaluated an Enhanced Machine Learning-Based Intrusion Detection System that integrates Signature-Based Detection, Random Forest classification, a 99.9% Confidence Gate, and MiniBatch K-Means anomaly detection. The Random Forest classifier achieved 99.59% accuracy, 99.58% precision, 99.59% recall, and 99.58% F1-score, while MiniBatch K-Means achieved a Silhouette Score of 0.8989. The controlled live-network evaluation demonstrated end-to-end packet capture, flow processing, detection, database logging, MITRE ATT&CK mapping, and SOC visualisation. The refined legitimate-traffic experiment recorded a 0.22% false-positive rate compared with 0.80% in the initial experiment, and the study reports a 73.09% reduction in false-positive detections. The contribution is therefore best understood as a practical hybrid IDS framework for known-attack detection and the identification of traffic patterns characteristic of potential zero-day attacks in a real-time monitoring environment. However, the evaluation was conducted in a controlled virtual network and should be extended to larger enterprise, cloud, and heterogeneous environments. The anomaly component identifies unusual traffic patterns; confirmation that an event is a true zero-day exploit requires additional threat intelligence, vulnerability analysis, or post-event investigation. Future work should also consider adaptive or online learning, more advanced anomaly models, automated incident response, explainable AI, larger multi-network datasets, and continued tuning of confidence and anomaly thresholds.

References

- Abdelaziz, M. T., et al. (2025). Enhancing network threat detection with Random Forest for network intrusion detection systems. *Journal of Network and Systems Management*.
- Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1), e4150.
- Ahmed, N., et al. (2025). Hybrid model for novel attack detection using a cluster-based classification approach. *Future Internet*, 17(6), 251.
- Ahmed, U., et al. (2025). Signature-based intrusion detection using machine learning and deep learning techniques. *Scientific Reports*.
- Aldweesh, A., Derhab, A., & Emam, A. Z. (2020). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems*, 189, 105124.
- Ali, T. E., Chong, Y. W., & Manickam, S. (2023). Comparison of machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking. *Applied Sciences*, 13(5), 3033.
- Alkasassbeh, M., & Al-Haj Baddar, S. (2023). Intrusion detection systems: A state-of-the-art taxonomy and survey. *Arabian Journal for Science and Engineering*, 48(8), 10021–10064.
- Almuhanna, R., & Dardouri, S. (2025). A deep learning/machine learning approach for anomaly-based network intrusion detection. *Frontiers in Artificial Intelligence*, 8, 1625891.
- Alsubaei, F., Abuhussein, A., & Shiva, S. (2022). A survey of intrusion detection systems in the Internet of Things. *Journal of Network and Computer Applications*, 198, 103294.
- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., et al. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8, 53.
- Arreche, O., Abdallah, M., & Guntur, T. (2024). XAI-IDS: Toward an explainable artificial intelligence framework for enhancing network intrusion detection systems. *Applied Sciences*, 14(10), 4170.
- Arreche, O., Bibers, I., & Abdallah, M. (2024). A two-level ensemble learning framework for enhancing network intrusion detection systems. *IEEE Access*.
- Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Computer Science*, 89, 117–123.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176.
- Dai, Z., Por, L. Y., Chen, Y.-L., Yang, J., Ku, C. S., Alizadehsani, R., Pławiak, P., & Acharya, U. R. (2024). An intrusion detection model to detect zero-day attacks in unseen data using machine learning. *PLOS ONE*, 19(9), e0308469.
- Danso, P. K., Neto, E. C. P., Dadkhah, S., Zohourian, A., Molyneaux, H., & Ghorbani, A. A. (2022). Ensemble-based intrusion detection for Internet of Things devices. In *Proceedings of the IEEE HONET Conference* (pp. 34–39).
- Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering*, 13(2), 222–232.

- Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2023). Deep learning for cyber security intrusion detection: Approaches, datasets, and future research directions. *Computer Networks*, 225, 109653.
- Gu, J., & Lu, S. (2021). An effective intrusion detection approach using support vector machine with Naïve Bayes feature embedding. *Computers & Security*, 103, 102158.