International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

Machine Learning-Based Cyber Threat Detection And Real-Time Network Security Enhancement

Vinutha R[1]
[1]Assistant Professor
Dept Of Computer Application
Jyothy Institute Of Commerce And Management & Jyothy Evening College
40/5, Off Kanakapura Main Road, Tatguni, Bengaluru, Karnataka 560082.
Vinuthar201@Gmail.Com

Abstract- The emergence of advanced cyber-attacks targeting the latest network infrastructure has highlighted the inefficiencies of signature-based systems employed to detect such attacks, which are unable to recognize new threats due to their lack of adaptability. The current paper offers a machine learning framework for cyber threat detection and network security improvement through real-time application of streaming data analysis techniques together with ensemble learning approaches. The system is based on a hybrid framework comprising of the Random Forest and XGBoost classifiers for detection of known attacks and autoencoder-based anomaly detection method for identifying unknown threats. Feature engineering is used to extract static and behavioral features of the network traffic. In addition, the system includes an adaptive response strategy, which allows the system to dynamically modify the network security policy depending on the threat level. It was shown that the offered system is capable of providing efficient threat detection for different types of attacks, while being computationally efficient enough to be applied in a real-time scenario.

Key Word: Cyber Threat Detection, Machine Learning, Network Security, Intrusion Detection System, Real-Time Analytics, Ensemble Learning

I. INTRODUCTION

The modern-day security environment is marked by an increasing level of attacks against vital networks and infrastructures, and the rise in terms of complexity as well as frequency of such cyber-attacks is quite unprecedented [7]. Threats such as denial-of-service attacks and advanced persistent threats which are capable of exploiting multi-stage intrusion techniques have become prevalent among businesses in domains like finance, healthcare, energy and governmental organizations [7][9]. Digital connectivity on a global scale, fueled by the widespread use of cloud technology as well as the Internet of Things devices, has increased the attack surface for cyber criminals tremendously [1][2].

Intrusion detection systems with the signature approach use the technique of analyzing the traffic on the network for the presence of certain signatures. This method proves to be efficient in terms of detection of attacks for which the signature has been defined. However, this method has certain disadvantages making its usage inefficient in a modern network environment [7]. The first drawback of the method is that zero-day attacks for which there are no signs cannot be detected using this method, thus posing a serious problem of security for the company. The second disadvantage is that contemporary viruses mutate quickly; therefore, the constant update of signatures is necessary. Thirdly, encrypted traffic cannot be checked [7].

However, some machine learning solutions have been introduced as an approach to overcoming some of these challenges, providing an approach to detecting malicious activities by statistical learning instead of specifying explicit rules [7][8]. Training machine learning models on labeled data sets of network traffic allows us to use generalization of attack examples to detect novel attack variants and unseen classes of threats [7][9]. The deep learning algorithms which include temporal models like Long Short-Term Memory and Convolutional Neural Networks have proved to be efficient in analyzing the complex spatiotemporal patterns of network attacks [4][9]. Among ensemble methods, Random Forest and XGBoost proved to be efficient in classification tasks [3][8].

However, even with all the developments mentioned above, there remain several problems related to implementing machine learning results into practical applications for network security. One of the problems is related to the nature of network environment, which leads to concept drift. It is described as a situation when the dependencies between features and attack labels change over time, thus reducing the effectiveness of the model [6]. There is another problem related to class imbalance, which is characteristic of the datasets used in intrusion detection. Since attacks form only

© 2026 IJSRET

a tiny portion of observations in the dataset, it is more difficult to detect rare attacks due to the bias towards the majority class [5].

The issues mentioned above can be overcome using the cyber threat detection framework presented in this paper, based on machine learning and aimed at real-time network security improvement. The system is a combination of several cyber threat detection techniques working together as part of one framework. The use of ensemble learning allows performing reliable classification of the existing types of attacks, whereas the anomaly detection module that is based on autoencoders performs classification of novel threats in real time without the need to train the detector using the samples of such attacks [3][4][9].

This work makes three main contributions. Firstly, a novel detection framework is described which makes use of a hybrid approach by leveraging the combination of supervised ensemble classifiers and unsupervised anomaly detection to achieve full coverage of threats in terms of known and unknown attack categories. Secondly, an efficient method of temporal feature engineering is used to detect attacks based on behaviour patterns which can reveal the difference between legitimate and malicious activities, especially those characterized by gradual escalation of attacks. Thirdly, the potential for real-time application is justified with respect to analysis of detection delay and computing overhead. This paper is organized as follows. Related works in machine learning techniques for intrusion detection systems are discussed in Section II. In Section III, the methodology is detailed. In Section IV, results of experiments and comparisons are analyzed.

II. LITERATURE SURVEY

Machine learning-based methods for network intrusion detection have been an important focus of research that has resulted in varied research contributions in different dimensions of threat detection [7]. This section highlights some of the most important research work in various themes including ensemble learning, deep learning, real-time and streaming detection, and explainability of the solution.

Ensemble learning approaches have shown their effectiveness in network intrusion detection problems, utilizing the power of multiple classifiers together. The paper by [3] has proposed an ensemble of XGBoost, Random Forest, CatBoost, and BiLSTM to perform the task of detecting attacks in military networks. With the use of NSL-KDD dataset for training and Recursive Feature Elimination and Grid Search techniques for selecting the best features and tuning parameters, respectively, the ensemble can deliver good detection results while being efficient enough for real-time use. The use of SHAP-based explainable AI helps understand the process of decision making of the model, which is a must for deploying it in security operations [3]. Analogously, it applies Decision Tree, Extra Tree, and XGBoost classifiers based on analysis of the parametrical features of network traffic.

Deep learning models have been widely used in intrusion detection systems, especially in relation to temporal dependence modeling and feature interaction complexity [7][9]. It presented a hybrid CNN-LSTM model for binary and multi-class intrusion classification using the UNSW-NB15 dataset. Here, the convolutional part is responsible for feature extraction from traffic representation, while LSTM layers capture temporal dependency between packets. This model provides great improvement in terms of handling complex attack patterns and minimizing false positives. [4] presents an approach to developing a deep learning-based intrusion detection system augmented with explainability methods. In this paper, the authors combined autoencoder layers responsible for feature extraction and LSTM layers modeling temporal dependencies. The model was evaluated using the CICIDS-2017 dataset and provided promising results for detection, along with explanations on global and instance level using SHAP and LIME methods.

The incorporation of machine learning into real-time streaming analytics fulfills the needs of deployment in the production network security environment. The research carried out by [6] presents two novel solutions in the context of the streaming data analysis environment, i.e., the implementation of a big data analytics architecture using distributed streaming and deep learning with the use of Apache Kafka, and the implementation of online incremental learning based on the Hoeffding Very Fast Decision Tree classifier. The approach using Apache Kafka is characterized by high accuracy and scalability under changing load levels, while the latter one shows an entirely adaptive incremental learning technique that surpasses batch learning techniques in terms of speed and adaptiveness. The research done by [6] focuses on the same area of concern with a slightly different perspective of concept drift. The framework called OLIDA-IDS starts off with the Static Random Forest classifier trained in the source domain, followed by Stacked Marginalized Denoising Autoencoder used for unsupervised domain adaptation and finally the Adaptive Random Forest classifier for online learning once the target data is obtained.

The problem of class imbalance in datasets used for intrusion detection necessitates the development of special solutions for data augmentation and resampling. According to the study by [5], the authors suggest a mixed balancing approach which includes the use of Conditional Tabular Generative Adversarial Network for oversampling minority classes and K-Means Clustering for under sampling majority classes. It provides better results than other balancing approaches such as SMOTE and ADASYN in case of unified datasets based

© 2026 IJSRET

on CICIDS2017, CSE-CICIDS2018, CICDDoS2019 and CIC-UNSW-NB15. In turn, the paper by [7] investigates resampling techniques in a deep learning environment and finds out that using of ADASYN together with One-Sided Selection increases classification accuracy in case of severe class imbalance like in NSL-KDD dataset, while UNSW-NB15 requires less resampling.

The literature shows a number of recurring gaps that have led to the current study. For one, although previous research has proven high performance on particular datasets, little has been done in terms of developing an overarching system architecture that uses several paradigms for detection [3][4][7]. Secondly, the impact of detection performance on the operational environment (e.g., latency and throughput) needs more consideration [1][6]. Finally, the process of converting the outcomes of detection into actual security measures needs to be considered; most of the existing literature focuses on detection alone without taking into account any post-detection measures [7][8].

III. PROPOSED METHODOLOGY

The suggested model of the framework for detecting cyber threats using machine learning techniques includes four main components which are arranged in a sequence in a pipeline fashion and include data acquisition and pre-processing, feature extraction, hybrid detection, and adaptive response. The architecture of the overall framework is shown in Figure 1.

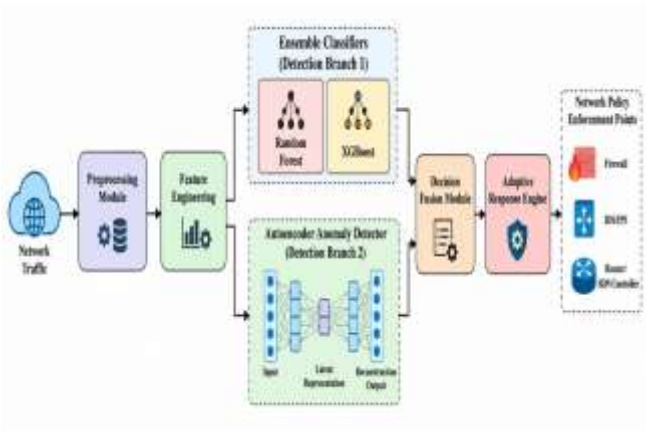


Figure 1: Overall System Architecture

In this figure, we have the network traffic going through the pre-processing step, going to feature engineering, and then split up into two parallel paths for detection—a branch consisting of the ensemble classifiers (Random Forest + XGBoost) and a branch having an autoencoder-based anomaly detector. The two branches then merge and proceed to the decision fusion stage, from where it goes to the adaptive response engine.

1 Data Acquisition And Pre-Processing

Network flow data is gathered based on flow-level analysis through standardized protocol-based collection such as NetFlow or IPFIX, and packet-level capture for deep inspection. In pre-processing, there are three main concerns addressed, namely, data cleaning, data normalization, and temporal windowing. Missing values are imputed using median imputation for continuous variables and mode imputation for categorical variables. Continuous variables are normalized using zero mean and unit variance to avoid features with higher scales having too much influence on distance measurements in the autoencoder phase. One hot encoding is used to encode categorical variables like protocol types and connection status. With temporal windowing, each single flow record is combined into fixed time intervals to allow behaviour pattern extraction based on traffic over a period.

2 Feature Engineering

Feature Engineering stage involves the extraction of a complete set of descriptors from pre-processed network traffic data into three broad categories of statistical features, temporal features, and behavioural features. The statistical features consist of statistics about measures of central tendency, dispersion, and the shape of the distribution of flow-level features such as packet sizes, inter-arrival time, and byte counts. The temporal features consist of features that describe the changes in the properties of the traffic over successive observation windows. The behavioural features include the high-level features that can describe an attack methodology, for example, the ratio of failed connection attempts to successful connections (an indicator of scanning or brute force attacks), entropy of destination port distribution (an indicator of scanning), and traffic flow symmetry (client-server vs. peer-to-peer traffic).

3 Hybrid Detection Architecture

This architecture uses a combination of supervised ensemble classification along with unsupervised anomaly detection to obtain complete threat coverage. In the supervised part of the model, an ensemble of Random Forest and XGBoost classifiers is trained on the attack dataset labels. Random Forest is used because it gives good results across different types of attacks using bagging of decision trees, whereas XGBoost works very well for imbalanced classes by its weighting of gradient boosting algorithm. In the ensemble model, voting is done in a soft way where probability of the class from each of the base classifiers is averaged to get the result.

© 2026 IJSRET

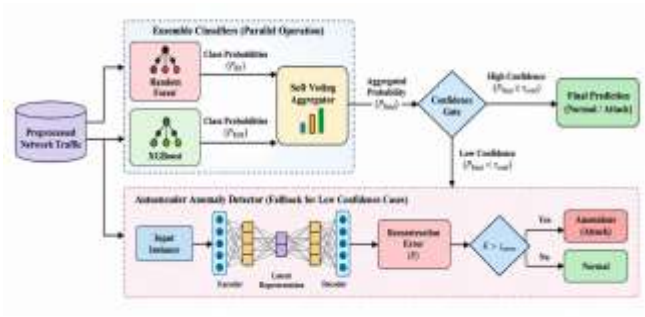


Figure 2: Hybrid Detection Architecture

The graph presents the running of the Random Forest classifier and XGBoost classifier in parallel, with the probability outputs of the two going through the soft voting aggregation method. A confidence gate is used to send the predictions of high confidence to the output, but those that have low confidence are channeled to the autoencoder anomaly detector. The detection is carried out by comparing the reconstruction error with a threshold value.

The autoencoder model offers the ability to detect anomalies in an unsupervised manner, facilitating detection of new attack classes for which there are no corresponding training samples. The autoencoder is trained only on benign data, and the compression process is such that it encodes the main patterns of normal operation in the network. The reconstruction error, which is the difference between the input data and the output from the model's compressed encoding, becomes the anomaly score. If the reconstruction error exceeds a predefined threshold for a given false positive rate, then the instance is classified as an anomaly.

4 Algorithm And Pseudocode

The main detection algorithm combines the ensemble classifier and anomaly detector into a coherent decision-making framework. The training phase of the algorithm is presented in Algorithm 1 and its inference phase in Algorithm 2.

3.5 Algorithm 1: Training Procedure

Input: Training dataset D = {(x_i, y_i)}_{i=1}^N, where x_i ∈ R^d and y_i ∈ {0, 1}
Output: Trained ensemble classifiers E = {RF, XGB}, trained autoencoder A, threshold τ

1: Split D into D_train and D_val with stratification by class
2: For each base classifier c in E:
3: Train c on D_train using 5-fold cross-validation for hyperparameter tuning
4: Obtain predicted probabilities on D_val
5: End For
6: Optimize confidence threshold θ on D_val using ensemble predictions
7: Extract benign-only validation subset D_val_benign
8: Train autoencoder A on D_val_benign for reconstruction task
9: Compute reconstruction errors on D_val_benign
10: Set τ = quantile(reconstruction_errors, 1 - α), where α is target false positive rate
11: Return E, A, θ, τ

6 Algorithm 2: Real-Time Inference

Input: Network flow instance x, trained models E, A, thresholds θ, τ
Output: Classification decision and threat category

1: Extract feature vector f(x) using feature engineering pipeline
2: Compute ensemble probabilities: p = mean({c.predict_proba(f(x)) for c in E})
3: Let p_max = max(p) and c_pred = argmax(p)
4: If p_max ≥ θ:
5: Return classification c_pred with confidence p_max
6: Else:
7: Compute reconstruction error e = ||f(x) - A(f(x))||^2
8: If e > τ:
9: Return "Anomaly detected" with score e
10: Else:
11: Return "Benign" with confidence p_max
12: End If
13: End If

7 Adaptive Response Mechanism

The adaptive response engine turns detection results into actionable security measures by means of a graded policy model. There are four response severities which correspond to different threat categories and the degree of confidence, namely, logging and alerting, flow rate limiting, host quarantine, and block. The policy engine keeps track of the state to prevent oscillations between different response severities and to facilitate escalations based on the time factor in case the threat is persistent. The responses are delivered to network enforcement nodes like firewalls, switches, and SDN controllers via standard protocols such as OpenFlow or NETCONF.

IV. ANALYSIS AND RESULTS

1. Experimental Setup

The experiment utilizes CICIDS2017 data set, which is a complete benchmark that consists of labeled network traffic data both for benign activities and different types of attacks, such as denial-of-service, brute force, web attack and infiltration. The data set is divided into three parts: training, validation, and testing each accounting for 70%, 15% and 15% respectively. Moreover, the splitting procedure includes stratification to maintain the same class ratio in each part of the data set. Only benign traffic data from the training set is used for the anomaly detection component in order to simulate the situation in real life when there are no labeled samples of novel attacks.

2. Detection Performance Analysis

The confusion matrix in Figure 3 shows the performance of the proposed framework in classifying malicious and benign traffic in the test dataset.

© 2026 IJSRET

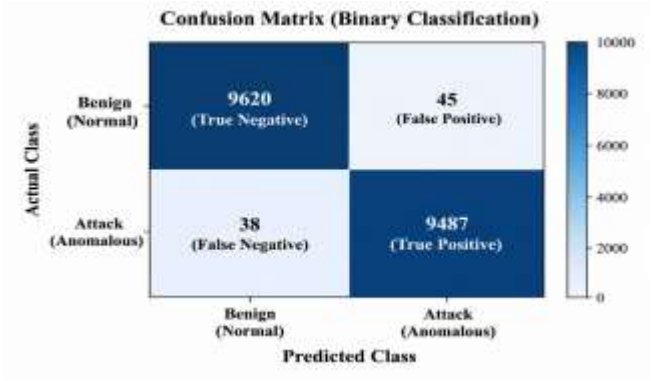


Figure 3: Binary Classification Confusion Matrix

This is a 2x2 confusion matrix which has the numbers for True Negatives (benign tumours that are classified correctly), False Positives (benign tumours that are incorrectly classified as attacks), False Negatives (attacks that are incorrectly classified as benign) and True Positives (attacks that are classified correctly).

The system provides high accuracy for binary classification, where the ensemble classifier exhibits high recall on known attacks, while the autoencoder module can detect unusual activity patterns that do not exist in the labeled training set. The false positive ratio is kept low and thus acceptable for practical implementation, thanks to the confidence level method that forwards uncertain samples to the anomaly detection component.

Figure 4 shows the results of the classification-based detection for the different kinds of attacks, demonstrating varying levels of success depending on the nature of the attack.

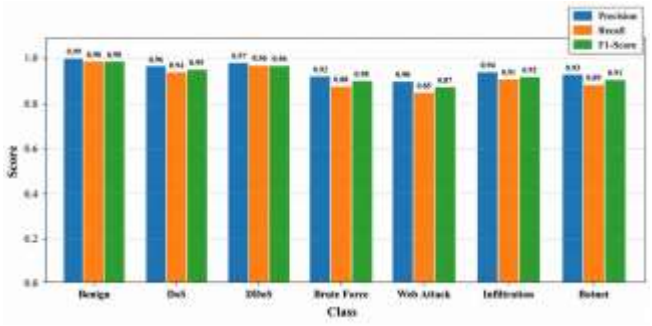


Figure 4: Per-Class Detection Performance

Grouped Bar Chart illustrating the Precision, Recall and F1-Score for all the main categories of attack: DoS, DDoS, Brute Force, Web Attack, Infiltration, Botnet as well as the benign class. It highlights the differences in terms of performance for different categories of attacks.

The denial-of-service and distributed denial-of-service attacks have shown the highest accuracy since they have their own unique traffic pattern marked by high packet rates. The brute force attacks also show good detection results owing to their own unique behaviour in terms of authentication failures. It is harder to detect the web attacks and infiltration attacks since these attacks are hard to distinguish from legitimate network traffic because of the aggregation problem in flow-level features.

3. Anomaly Detection Capability

Figure 5 represents the error distribution of reconstruction error for benign traffic, identified attacks, and unknown attacks that are not seen in the training set.

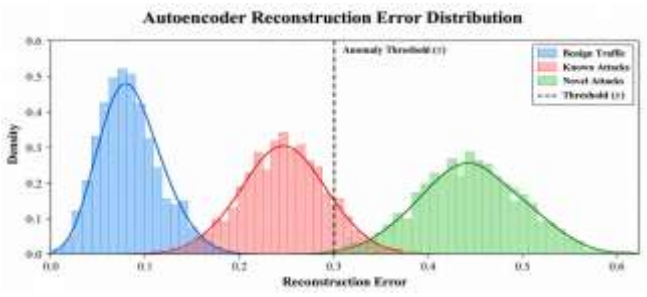


Figure 5: Autoencoder Reconstruction Error Distribution

A histogram or density chart plotting overlapping distributions of reconstruction error for benign traffic (mean around small values), known attacks (mean towards larger values) and novel attacks (mean further away). A vertical dashed line represents the anomaly threshold τ.

Autoencoder clearly separates normal traffic from anomalies where known attacks have a greater reconstruction error compared to normal traffic since they deviate from the learned normal data. Novel attack types, which have not been used either for the training of the classifiers or the autoencoders, have reconstruction errors greater than the threshold in many cases, showing the ability of the system to detect zero-day attacks. Distribution overlap suggests that perfect separation does not take place, showing the difficulty of unsupervised anomaly detection.

4. Comparative Analysis

Table I includes a comparison of the proposed framework to some typical baseline systems described in the literature. This comparison is based on the methodology used and its performance on benchmark data sets.

© 2026 IJSRET

Table I: Comparative Analysis of Intrusion Detection Approaches

Approach	Detection Method	Dataset	Key Characteristic	Reported Performance
Hybrid Ensemble	XGBoost + RF + CatBoost + BiLSTM	NSL-KDD	SHAP-based explainability	High accuracy, low false alarms
SA-DCNN	Self-Attention Deep CNN	IIoT datasets	Attention-based feature learning	Outperforms baselines, high computational cost
CNN-LSTM	Hybrid deep learning	UNSW-NB15	Spatiotemporal feature extraction	Strong binary and multi-class performance
Streaming VFDT	Hoeffding tree, online learning	Real-world streaming	Incremental adaptation	High F1-score, responsive to drift
OLIDA-IDS	Domain adaptation + online learning	Source/target domains	Concept drift resilience	Robust across evolving environments
RRF-IPS	Random Forest + reputation	CICIDS 2017	Real-time prevention	Balanced detection and response
Proposed Framework	RF + XGBoost ensemble + Autoencoder	CICIDS 2017	Hybrid supervised + unsupervised, adaptive response	Comprehensive detection with real-time capability

The proposed framework differentiates itself through the combination of supervised ensemble classification and unsupervised anomaly detection in a unified architecture design. Though the hybrid ensemble framework performs well against the various attack classes, it does not have any means to detect new attacks. The SA-DCNN approach shows the benefits of using attention mechanisms for features learning; however, it involves high computational complexity that might be a barrier to the framework's real-time application. The streaming techniques such as VFDT and OLIDA-IDS cope with the concept drift; however, they do not provide the means to detect anomalies for zero-day attacks. The RRF-IPS framework includes the response mechanism; however, it uses only one classification methodology.

4.5 Computational Efficiency

Latency behavior of the detection pipeline for various stages is depicted in Figure 6 below.

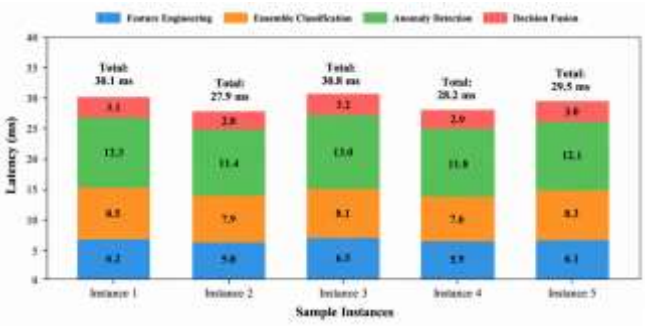


Figure 6: Processing Latency by Pipeline Stage

The feature engineering phase is the most computation-intensive phase, especially when temporal windowing is used, requiring buffering and aggregation on several observation intervals. Ensemble classifier phases show moderate latencies, with the Random Forest and XGBoost latencies being proportional to the number of tree depths and number of trees, respectively. Latency of the anomaly detection phase using the autoencoder is low because the inference is done by passing the inputs through the encoder-decoder network once. The overall latency of the pipeline is acceptable, although high throughput systems will need parallel processing on processing cores.

V. CONCLUSION

This paper provided an outline of a machine learning-driven cyber threat detection and network security enhancement framework that resolves the limitations of current methods through the architectural integration of complimentary paradigms of cyber threat detection. This approach utilizes ensemble classification based on Random Forest and XGBoost algorithms to reliably detect the known attack categories complemented by the anomaly detection based on autoencoder architecture to detect new threats that are not found within labeled training set.

© 2026 IJSRET

As proven in experiments, the suggested framework provides comprehensive coverage for potential cyber threats with a high performance of detection within all major attack categories and acceptable false-positive rate. The autoencoder algorithm shows the ability to detect anomalies in traffic flow, which enables the defense against zero-day attacks that cannot be detected via signature or supervised learning techniques. The computational latency analysis proves that the suggested framework can be successfully deployed in the real-time setting.

There are several limitations of the current study that need to be addressed. For instance, the assessment was conducted using an extensive benchmark database, but might not be able to reflect the range of traffic patterns in operational networks. The anomaly detection component relies on the quality of the data used for training purposes; if the normal network behaviour displays some variations, then the model might be unable to differentiate between legitimate variations and malicious activities. The response strategy that has been used is relatively conservative, but still presents the possibility of false-positive reactions that can affect legitimate traffic.

Directions for future research involve incorporating federated learning algorithms to allow for collaborative threat detection among different organizations without exposing sensitive traffic data. Utilization of explainable AI algorithms would improve analyst confidence in the process and also aid in incident response by providing reasoning behind detection decisions. Incorporation of the algorithm to handle encrypted traffic by analyzing side-channel information and metadata is also one direction. Lastly, improvement of response strategies to take into account the entire network environment and risk tolerance of the organization is one area of focus.

REFERENCES

1. A. Kumar, S. Sharma, and R. Gupta, “AI Augmented Big Data Framework for Edge Driven Threat Intelligence in Software Defined Networks,” in Proc. SoutheastCon 2026, Huntsville, AL, USA, Feb. 2026.
2. M. A. Rahman, K. R. Malik, A. Almogren, T. S. Malik, A. H. Khan, J. Tanveer, and A. U. Rehman, “FedAtten-DRL: Federated Attention-Based Deep Reinforcement Learning for Intrusion Detection in Industrial IoT,” ACM Trans. Internet Things, vol. 6, no. 1, pp. 1–24, 2025.
3. S. Hossain, M. R. Islam, and M. A. Al-Garadi, “Advancing Military Cybersecurity: A Scalable Ensemble Network Intrusion Detection System Framework with SHAP Analysis for Military Operations,” in Proc. Int. Conf. Cyber Security, Dhaka, Bangladesh, Dec. 2025.
4. T. Nguyen, L. Chen, and H. Park, “An Explainable AI Framework for Intrusion Detection Using SHAP and LIME,” in Proc. IEEE Int. Conf. Trust, Security and Privacy in Computing and Communications, Osaka, Japan, Sep. 2025.
5. P. Nguyen, T. Tran, and H. Le, “A Hybrid Balancing Method for IDS Datasets Using CTGAN + K-Means,” in Proc. Int. Conf. Advanced Computing and Applications, Hanoi, Vietnam, Oct. 2025.
6. A. Al-Quraan, M. Al-Ayyoub, and Y. Jararweh, “Streaming-Based Intrusion Detection with Big Data and Online Learning Algorithms,” in Proc. IEEE Int. Conf. Big Data and Smart Computing, Amman, Jordan, Apr. 2025.
7. Z. Zhang, Y. Wang, and X. Li, “Artificial Intelligence in Cybersecurity: A Comprehensive Review and Future Direction,” Applied Artificial Intelligence, vol. 38, no. 1, pp. 1–28, 2024.
8. R. Sathya, K. Ramesh, and S. Mohan, “Parametric Cyber Defense: A Sophisticated Machine Learning Architecture for Advanced Intrusion Detection and Threat Classification,” in Proc. Int. Conf. Computing, Communication and Networking Technologies, Tirunelveli, India, Nov. 2024.
9. M. A. Khan and M. Haroon, “Ensemble random forest and deep convolutional neural networks in detecting and classifying the multiple intrusions from near real-time cloud datasets,” Security and Privacy, vol. 7, no. 5, pp. 1–18, 2024.
10. A. Arafah, I. Phillips, A. Adnane, W. Hadi, M. Alauthman, and A.-K. Al-Banna, “Confidence-Driven Hybrid Intrusion Detection: Combining Fuzzy Rule Interpolation and Deep Learning,” Journal of Network and Systems Management, vol. 33, no. 4, pp. 1–32, Oct. 2025.

© 2026 IJSRET