Cyber Threat Detection and Prevention using AI-ML Model (Hybrid Approach)

Mangalgouri P Kademani[1][0009−0000−3214−5042], Yuvaraj P Rathod[2][0009−0004−2635−9975], Nisha B Kubasad[3][0009−0001−1361−3805], Ramaraddi G Maraddi[4][0009−0003−1440−2350], and Ashok Chikaraddi[5]

School of Computer Science & Engg, KLE Technological University Hubballi, India
01fe23bcs422@kletech.ac.in, 01fe23bcs423@kletech.ac.in,
01fe23bci401@kletech.ac.in, 01fe23bcs416@kletech.ac.in,
chikaraddi@kletech.ac.in

Abstract. This work presents an integrated approach to cyber threat detection and prevention using advanced machine learning and deep learning techniques, combined with a real-time monitoring dashboard. The proposed hybrid architecture makes use of the CICIDS2017 dataset, which contains 682578 rows and 78 columns and covers a wide range of modern attack types. To identify unusual activity, an ensemble of Isolation Forest and Autoencoder models is employed; the Autoencoder reconstructs typical patterns to identify deviations, while the Isolation Forest focuses on outlier detection in benign traffic. With a 74% accuracy rate, this ensemble outperformed individual models. After detection, XGBoost is used to classify detected assaults into many classes, achieving high precision and recall in multiple classes, particularly for DDoS and PortScan types. A CNN-LSTM model, which has a 99% classification accuracy, is trained on sequential network data to anticipate possible attack scenarios for proactive security. The models were included into a cyber threat monitoring dashboard that enables network managers to make decisions, visualize forecasts in real-time. The system's performance has been evaluated using measures such as F1-score, recall, accuracy, and precision. The outcomes show how well the hybrid ensemble learning technique works to improve cyber threat identification and prevention in current network environments when combined with deep learning and an interactive dashboard.

Keywords: Cybersecurity · Ensemble Learning · Isolation Forest · Autoencoder · XGBoost · CNN-LSTM · Threat Detection · Network Attacks.

1 Introduction

In the modern digital era, the frequency and sophistication of cyber-attacks are escalating rapidly, posing a significant threat to critical infrastructures, enterprises, and individual users. Real-time threat detection and prevention has

emerged as a key component of cybersecurity systems [13]. Since attacks are always changing and becoming more covert, many harmful actions continue to avoid defenses even with the implementation of conventional security measures like firewalls and Intrusion Detection Systems (IDS). This calls for the development of threat detection systems that are smarter, more adaptable, and more accurate.

The high-dimensional, unbalanced, and diverse character of network traffic data makes cyber threat detection difficult since malicious activity can be masked within a huge amount of truthful data. Conventional signature-based systems frequently miss new or zero-day assaults. Because of their ability to automatically identify patterns and identify anomalies without the need for predefined rules, machine learning (ML) and deep learning (DL) techniques have become strong alternatives to this problem [2].

In order to detect and prevent cyber threats, this work proposes a hybrid AI-ML method that combines supervised and unsupervised learning approaches [8]. The CICIDS2017 dataset, which contains 682038 rows and 78 columns and comprises a variety of modern attack types and benign traffic, is a thorough and realistic baseline for intrusion detection that is used in the system's design and evaluation [6].

The CICIDS2017 dataset is used in a multi-stage pipeline for cyber threat detection in the proposed Fig. 1. Data collection and preprocessing are the first steps, during which unprocessed network traffic is cleaned, standardized, and prepared for analysis. After that, feature engineering is used to enhance model performance by reducing dimensionality and identifying key attributes.

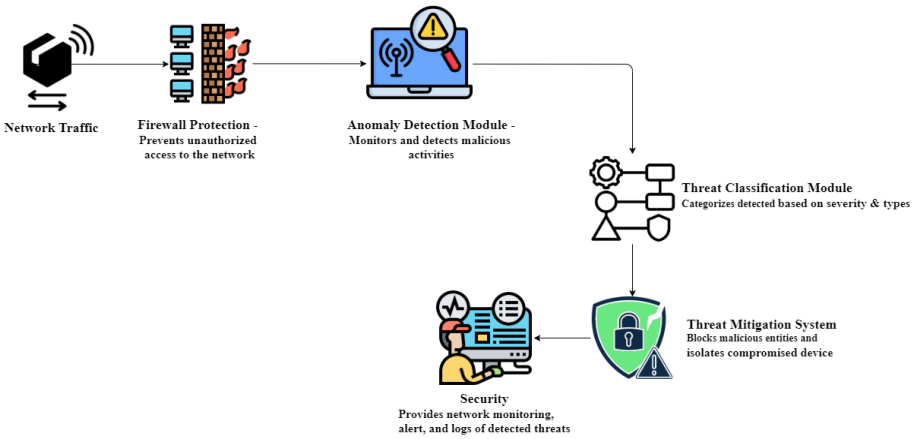

Fig. 1. Proposed hybrid AI-ML architecture for cyber threat detection and prevention

Both supervised and unsupervised learning methodologies are included into the hybrid model architecture. An autoencoder learns typical traffic patterns and

highlights deviations, whereas an isolation forest uses data isolation to identify anomalies. Convolutional neural networks (CNNs) and long short-term memory (LSTM) identify complex sequence-based attacks, while XGBoost, a gradient boosting model, efficiently manages structured data [14]. Real-time threat monitoring is made possible by the visualization of results via a cyber threat monitering dashboard.

This work is divided into five sections: Section 2 describes the algorithms for machine learning that are currently available and an overview of the machine learning algorithms utilized in the system, such as CNN-LSTM, XGBoost, Autoencoder, and Isolation Forest, along with information on how well they detect different kinds of cyberattacks. The entire data pipeline, including feature engineering methods, preparing the CICIDS2017 dataset, and integrating various models into a soft voting ensemble framework, is described in Section 3. The experimental findings are shown in Section 4, where key metrics including accuracy, precision, recall, and F1-score are used to compare the performance of the individual models. Section 5 concludes with a discussion of the findings, possible limitations, and future research, including improvements to real-time deployment techniques for better cyber threat identification.

2 Background Study

Cyber threat detection has become a critical domain in modern network security, with an increasing reliance on ML to handle the scale and complexity of current attack surfaces [16]. Traditional models, such Naive Bayes and Logistic Regression, were among the first methods used in IDS because of their interpretability and ease of use [12]. In situations where there were well established statistical correlations between the output label (malicious or benign) and the input variables (such as packet size, duration, or port usage) or linearly separable features, these models worked effectively. However, in today's dynamic cyber environment, their usefulness is severely limited due to their incapacity to simulate complex and non-linear feature interactions [19].

In order to overcome these limitations, more sophisticated tree-based ensemble techniques like Random Forests and XGBoost have been used extensively [3]. XGBoost in particular is well-known for its capacity to effectively handle missing values, handle massive amounts of structured data, and use gradient boosting to identify intricate patterns [4]. Despite their great accuracy, these models frequently have a significant computational cost, which makes them less suitable for real-time or resource-limited circumstances where low latency is crucial.

At the same time, DL methods have become more and more well-liked due to their capacity to identify significant patterns in high-dimensional, unstructured data [7]. CNN and LSTM networks are two examples of architectures that have proven especially useful for detecting stealthy or advanced persistent threats (APTs), detecting subtle variations in flow behavior, and modeling the spatiotemporal characteristics of network traffic [10]. Despite their strength, these

models are difficult to understand and require a lot of processing power, which prevents security operations centers (SOCs) from using them more widely [9].

In order to get beyond the individual drawbacks of standalone models, researchers are increasingly using ensemble learning methods. Through techniques like bagging, boosting, stacking, or weighted averaging, ensemble learning combines the predictive skills of several models to enhance generalization, lower variance, and lessen false positives [15]. To prevent redundancy and guarantee complementary performance increases, it is necessary to carefully choose a variety of base models and adjust combination techniques while building an effective ensemble [17].

Beyond basic ensemble approaches, hybrid intrusion detection frameworks have also been explored. Tree ensembles, autoencoders, and deep networks are examples of heterogeneous learners that are combined in stacking-based IDS models. This approach frequently results in improved accuracy at the expense of slower and more complex systems. Federated learning techniques have also been used for IDS in distributed or Internet of Things (IoT) environments. These techniques enable models to jointly learn threat patterns without centralizing sensitive data, improving privacy but requiring strong aggregation procedures.

To address these issues, this work suggests a hybrid strategy that combines four different models: Autoencoder, CNN-LSTM, XGBoost, and Isolation Forest to take use of both supervised classification and anomaly detection in a single system [11]. Every model makes a distinct contribution where Isolation Forest is useful for detecting zero-day and uncommon attacks since it measures how quickly data can be segregated in decision trees to identify outliers [5]. An unsupervised neural network called Autoencoder learns compact representations of benign (normal) information and uses reconstruction error to identify anomalies [18]. With organized structured information, XGBoost is an excellent multi-class classification tool that provides quick and precise threat classification [1]. For the purpose of detecting intricate infiltration patterns across time, CNN-LSTM integrates temporal sequence modeling and spatial pattern extraction [20].

where each model produces probabilistic results that are weighted according to how well it performs on validation data. This guarantees robustness against a variety of attack types in addition to increasing detection accuracy and recall. The system's practical usability is further increased by the incorporation of a Flask-based web dashboard, which enables real-time interaction, prediction visualization, and user-friendly monitoring.

3 Proposed Methodology

The work proposes a hybrid AI-ML-based intrusion detection and prevention system that uses models for sequential pattern recognition the Fig. 2 represents the end-to-end workflow of the proposed work where the CICIDS2017 dataset is first preprocessed before being divided into training and testing groups. Combining the Isolation Forest and Autoencoder models, anomaly detection identifies suspicious networks. Traffic is further examined using the XGBoost classifier to classify the type of attack if it is determined to be suspicious. By taking use of

periodic patterns in the traffic, the CNN-LSTM model is used in combination to mitigate threats. Every prediction is recorded in a MongoDB database, encompassing both harmful and benign activity. Following the visualization of these logs, administrators can review connected devices, monitor system IPs, identify malicious IPs, examine attack patterns, and implement mitigation techniques like IP blocking using a Flask-based real-time online dashboard. The solution guarantees thorough threat identification and response, allowing for log-based analysis and real-time security monitoring.

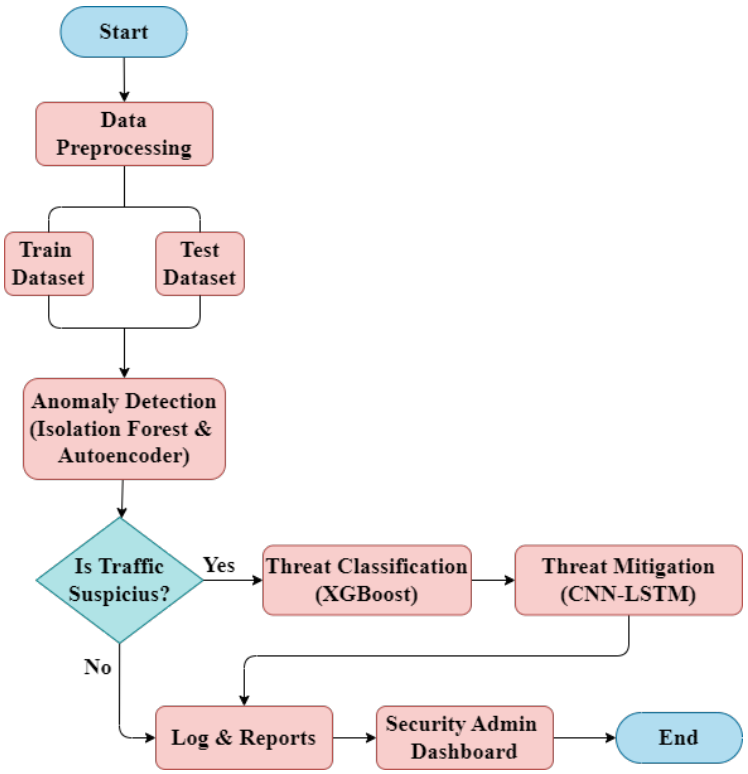

Fig. 2. Proposed system for Cyber Threat Detection and Prevention Models

3.1 Data Collection

The CICIDS2017 dataset, which has 682,578 rows and 79 columns following preprocessing, is used by the system. It includes labeled examples of different attack methods (e.g., DDoS, PortScan, Web Attacks, Brute Force), as well as benign flows and realistic network traffic.

3.2 Data Preprocessing

The following steps are involved in preprocessing:

Column cleaning Dropping non-informative features (e.g., IPs, timestamps).

Removal of null and inf Replacing values for NaN, inf, and -inf.

Label Encoding For supervised learning, labels are encoded using LabelEncoder.

Feature Scaling Standardizing features using StandardScaler.

Binary Labeling Creating a binary_label (0 = benign, 1 = attack) for use in binary classifiers.

3.3 Dataset Splitting

To maintain the distribution of attack types, stratified sampling is used to divide the processed dataset into 80% training and 20% testing. CNN-LSTM generates flow data sequences with a window size of 6 in order to record temporal behavior.

3.4 Model Training

Isolation Forest Isolation Forest only receives training on safe traffic. Tree partitions are used to isolate abnormalities. The model assigns an anomaly score, and a threshold is used to classify the sample as in equation 1:

- If the anomaly score exceeds the threshold, it is considered anomalous $\rightarrow$ $y_{IF} = 1$
- Otherwise, it is normal $\rightarrow$ $y_{IF} = 0$

$$y_{IF} = \{\, 1\, , if sample is anomalous 0, otherwise \tag{1}$$

Autoencoder The benign flows are reconstructed using a neural autoencoder. Abnormalities are recognized using Mean Squared Error (MSE).

Loss Function: The equation 2 measures the average squared difference between the original input x_i and its reconstruction $\hat{x}_i$, indicating reconstruction error and n represents the total number of data samples.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{2}$$

Dynamic Thresholding: Sets the anomaly detection threshold as the 90th percentile (P90) of MSE values from benign samples in equation 3.

$$Threshold = P_{90}(MSE_{benign}) \tag{3}$$

Samples with MSE threshold are classified as anomalies: In equation 4 y_{AE} is the predicted label from the Autoencoder; where $I[\cdot]$ is the indicator function returning 1 if the MSE exceeds the threshold (e.g., 90th percentile of benign MSE), else 0.

$$y_{AE} = I[MSE > Threshold] \tag{4}$$

Ensemble Anomaly Detection Predictions from Isolation Forest and Autoencoder are combined via logical OR in equation 5:

$$y_{Anomaly-Ensemble} = y_{IF} \vee y_{AE} \tag{5}$$

This makes use of both models' advantages to increase recall.

3.5 XGBoost

To identify the type of attacks, XGBoost is trained using just attack samples. To fix the class imbalance, class weights have been used.

Objective Function: The total loss L combines the prediction loss $\ell(y_i, \hat{y}_i)$ for each sample by equation 6 and a regularization term $\Omega(T_m)$ to penalize model complexity across M trees.

$$L = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{m=1}^{M} \Omega(T_m) \tag{6}$$

Tree Weight Update: The equation 7 Calculates the optimal weight w_m for tree m using the ratio of first-order (h_i) and second-order (g_i) gradient statistics from the loss function.

$$w_m = -\frac{\sum h_i}{\sum g_i} \tag{7}$$

where g_i and h_i are the gradient and hessian, respectively.

3.6 CNN-LSTM

To identify attacks over time, CNN-LSTM is trained on traffic feature sequences. Patterns in space are extracted using the Conv1D layer.

Activation Function: The Sigmoid activation function maps the input z to a value between 0 and 1 as in equation 8, suitable for binary classification tasks.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (Sigmoid) \tag{8}$$

3.7 Performance Evaluation

All models are evaluated using:Accuracy, Precision, Recall, F1-Score, Macro and Weighted Averages. When compared to individual models, the ensemble (IF + AE) improves recall and F1, XGBoost classifies the attack, and CNN-LSTM further verifies threats recognized over sequences.

3.8 Dashboard Interface

A web-based dashboard was created utilizing the Flask web framework in Python to allow for real-time interaction with the suggested cyber threat detection system. Providing end users, security analysts, or system administrators with extensive capabilities for threat visualization and anomaly monitoring, the dashboard functions as the main user interface.

Network traffics are fetched from the ARP (Address Resolution Protocol). Plotly-based interactive charts are used to dynamically show attack types and anomaly statistics, improving situational awareness. In order to resolve DNS asynchronously using multithreading, the system uses a subprocess to scan the local ARP database and parses device IP, MAC address, hostname, and online status.From prediction logs, malicious IP addresses, associated threat types, and detection timestamps are taken out and displayed on the dashboard. By submitting mitigation requests, users can actively block suspicious IP addresses. JavaScript keeps the dashboard in line with backend activity by enabling updates every 30 seconds. To provide responsive and user-friendly interactions, HTML, CSS, and JavaScript were used in the interface implementation.

3.9 Log Storage

The backend database used to store all real-time monitoring and prediction logs is MongoDB. Real logs are created in batches and added using insert_many(). The timestamp, IP address of the source and destination, attack type, detection signature, protocol, port, payload size, threat source, and geographic metadata are all included in each log document.Live monitoring are supported by MongoDB's ability to facilitate quick and adaptable document retrieval through the use of filters. The web user interface allows logs to be exported in CSV format for offline reporting and analysis.

4 Results

4.1 Model Training and Evaluation

Several machine learning and deep learning models were developed and evaluated using the CICIDS2017 dataset in order to accurately identify and stop cyberthreats. XGBoost, CNN-LSTM, Autoencoder, Isolation Forest, and a proposed ensemble model that combines autoencoder and isolation forest predictions to improve robustness and reliability.

The Autoencoder was trained using only typical traffic data in a semi-supervised fashion. Reconstruction error was employed as a threshold to identify anomalies during inference. By allocating anomaly scores according to feature-space isolation, the Isolation Forest functioned similarly in an unsupervised environment.

The Ensemble model uses a majority voting technique for binary classification to combine the results of the autoencoder and isolation forest models. By utilizing each algorithm's complementary strengths, this method increases general robustness.

A multi-class classification assignment was used to train XGBoost for supervised learning, and it demonstrated amazing precision and recall for all attack types. The CNN-LSTM model was used to capture temporal dependencies in the data by taking advantage of the sequential pattern of network traffic flows.

Table 1. Model Performance Metrics for Cyber Threat Detection

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Isolation Forest	0.53	0.51	0.53	0.51
Autoencoder	0.77	0.77	0.77	0.76
Ensemble(ISF + AE)	0.80	0.78	0.78	0.79
XGBoost	1.00	1.00	1.00	1.00
CNN-LSTM	0.99	0.99	0.99	0.99

According to the results from Table 1, the Ensemble model provides the most balanced and dependable performance, particularly when applied to wider network settings, while XGBoost and CNN-LSTM show nearly perfect classification on the test set. It performs better in F1-score than the independent Autoencoder and Isolation Forest, which makes it a good fit for real-time threat detection systems where robustness and consistency are essential.

Accuracy, precision, recall, and F1-score are used to compare the performance of five models in Fig. 3. While Autoencoder and Isolation Forest perform moderately, their ensemble (IF + AE) slightly increases accuracy. XGBoost and CNN-LSTM perform better in anomaly detection tasks, as seen through their near-perfect results on all measures.

4.2 System Implementation

A Flask-based web application that combines ML and DL trained models is used to construct the proposed hybrid cyber threat detection system. Real-time anomaly detection and result visualization via an easy-to-use user interface.

A real-time network security dashboard with system IP addresses, connected devices, and a list of malicious IPs found is displayed in the Fig. 4. This configuration offers fast visibility into network security risks as well as centralized monitoring.

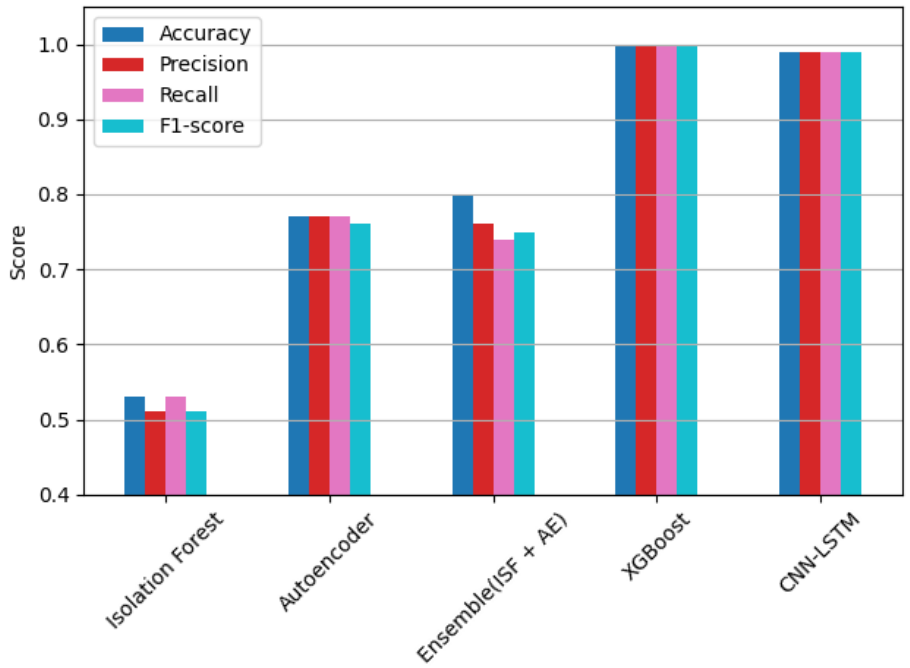

Fig. 3. Performance Metrics of Cyber Threat Detection and Prevention Models

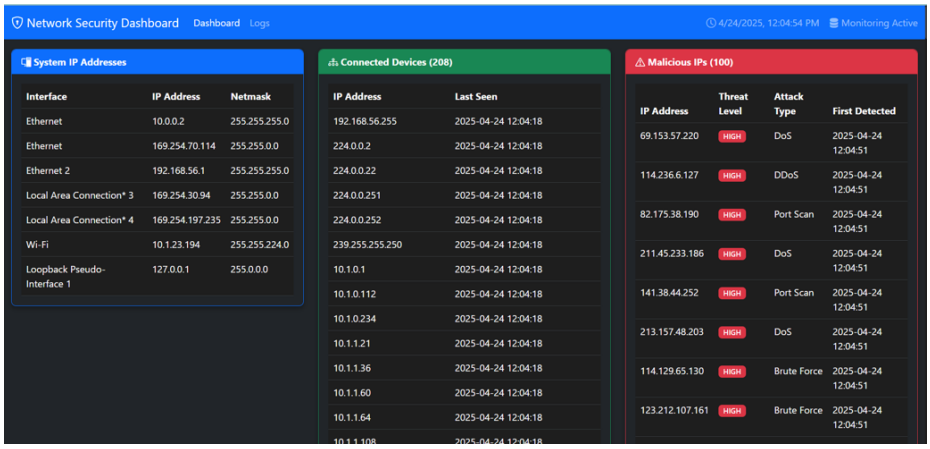

Fig. 4. Network Security Dashboard Overview

The statistical distribution of the different types of cyberattacks that have been identified in the network environment under observation is shown in the pie chart in Fig. 5.

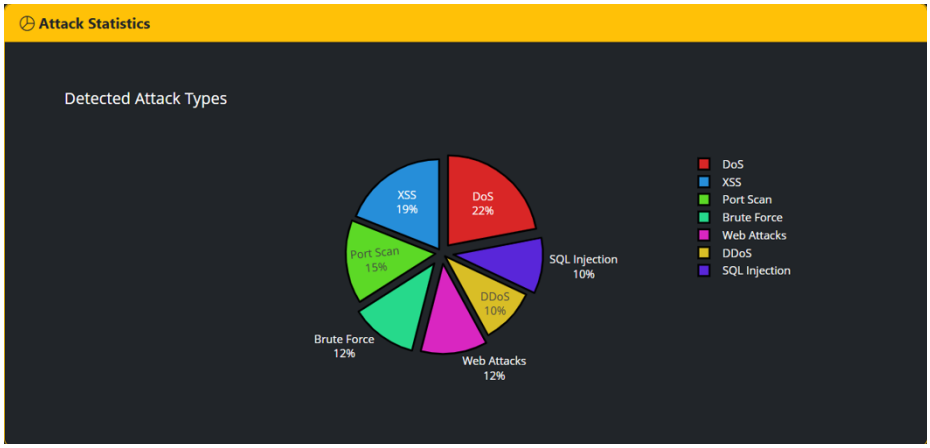

Fig. 5. statistical distribution of Detected Attack Types

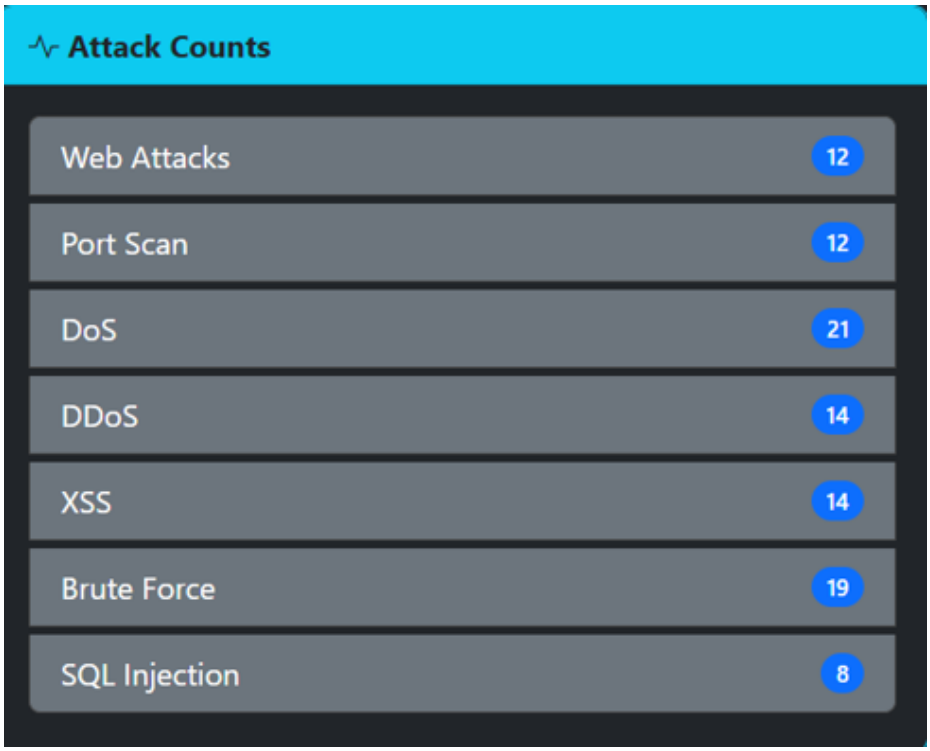

Fig. 6. Recorded Counts of Detected Attack Types

The frequency of the different sorts of attacks that the network monitoring system has detected is displayed in a bar-style list in Fig. 6. In order to prioritize

defense measures in the network security plan, these counts offer a quantitative viewpoint on threat occurrences.

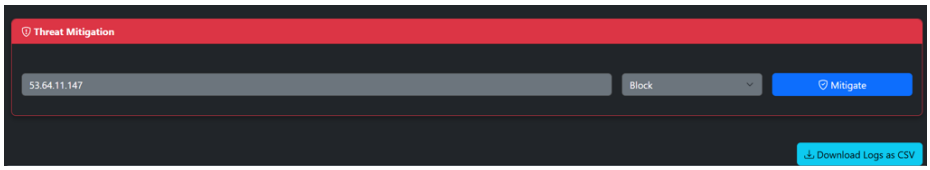

Fig. 7. Threat Mitigation Interface of the Network Security Dashboard

This Fig. 7 shows how the network security dashboard's threat mitigation features work. Administrators can use it to enter a malicious or suspicious IP address and implement a mitigation measure, such blocking the address.

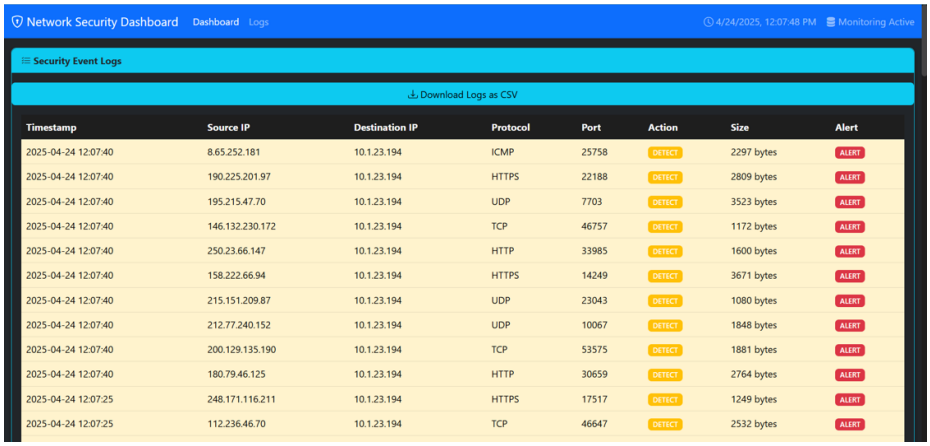

Fig. 8. Real-time Network Security Dashboard displaying active security event logs

A network security dashboard displaying many real-time security events is shown in the Fig. 8. The timestamp, protocol type, port number, source and destination IP addresses, and data size are all recorded in each log entry.

5 Conclusion and Future work

The work aimed to identify the most effective approach for detecting cyber threats using a combination of ML and DL models. Using the CICIDS2017 dataset, CNN-LSTM, XGBoost, Autoencoder, and Isolation Forest were used to identify and categorize threats. The unsupervised models were outperformed

by XGBoost and CNN-LSTM, which performed almost perfectly on every evaluation metric. Autoencoder and Isolation Forest, on the other hand, produced outcomes that were average. When compared to separate unsupervised models, the ensemble model which included Isolation Forest and Autoencoder improved detection performance and produced a more robust and balanced result. These findings show that hybrid methods can greatly enhance cyber threat detection capabilities, particularly when they combine supervised and unsupervised learning. The trained models were also integrated into a centralized dashboard in a Flask-based real-time cyber threat detection and prevention system. This system tracks security actions, shows connected and malicious IPs, visualizes real-time threat occurrences, and offers an interface for proactive threat mitigation, including blocking suspicious IPs and exporting logs for study. This real-world application demonstrates the system's suitability for deployment in network security and monitoring tasks.

Future studies could further improve detection accuracy, scalability, and real-time responsiveness by including external threat intelligence feeds, integrating live packet capture technologies, using advanced ensemble techniques like stacking, and using adaptive learning approaches.

References

1. Al-Essa, M., Appice, A.: Dealing with imbalanced data in multi-class network intrusion detection systems using xgboost. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 5–21. Springer (2021)
2. Cook, J., Rehman, S.U., Khan, M.A.: Security and privacy for low power iot devices on 5g and beyond networks: Challenges and future directions. IEEE Access **11**, 39295–39317 (2023)
3. Demir, S., Sahin, E.K.: An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using adaboost, gradient boosting, and xgboost. Neural Computing and Applications **35**(4), 3173–3190 (2023)
4. Dhaliwal, S.S., Nahid, A.A., Abbas, R.: Effective intrusion detection system using xgboost. Information **9**(7), 149 (2018)
5. Fan, J.: Analyzing the applicability of isolation forest for detecting anomalies in time series data. B.S. thesis, J. Fan (2025)
6. Jeon, B.: Enhancing Intrusion Detection Systems Using Deep Learning Techniques: A Comparative Study on CICIDS 2017 Dataset. Ph.D. thesis (2023)
7. Maxima, A.: Integration and analysis of unstructured data towards database optimization and decision making using deep learning techniques. Ph.D. thesis, Kampala International University (2024)
8. Meng, X.: Advanced ai and ml techniques in cybersecurity: Supervised and unsupervised learning, and neural networks in threat detection and response. Applied and Computational Engineering **82**, 24–28 (2024)
9. Muniz, J., McIntyre, G., AlFardan, N.: Security operations center: Building, operating, and maintaining your SOC. Cisco Press (2015)
10. Mutalib, N.H.A., Sabri, A.Q.M., Wahab, A.W.A., Abdullah, E.R.M.F., AlDahoul, N.: Explainable deep learning approach for advanced persistent threats (apts) detection in cybersecurity: a review. Artificial Intelligence Review **57**(11), 297 (2024)

11. Nalini, M., Yamini, B., Ambhika, C., Siva Subramanian, R.: Enhancing early attack detection: novel hybrid density-based isolation forest for improved anomaly detection. International Journal of Machine Learning and Cybernetics pp. 1–19 (2024)
12. Namdar, J.H., Yonan, J.F.: Revolutionizing iot security in the 5g era with the rise of ai-powered cybersecurity solutions. Babylonian Journal of Internet of Things **2023**, 85–91 (2023)
13. Ofoegbu, K.D.O., Osundare, O.S., Ike, C.S., Fakeyede, O.G., Ige, A.B.: Real-time cybersecurity threat detection using machine learning and big data analytics: A comprehensive approach. Computer Science & IT Research Journal **4**(3) (2024)
14. Rajapaksha, S., Kalutarage, H., Al-Kadri, M.O., Petrovski, A., Madzudzo, G., Cheah, M.: Ai-based intrusion detection systems for in-vehicle networks: A survey. ACM Computing Surveys **55**(11), 1–40 (2023)
15. Rane, N., Choudhary, S.P., Rane, J.: Ensemble deep learning and machine learning: applications, opportunities, challenges, and future directions. Studies in Medical and Health Sciences **1**(2), 18–41 (2024)
16. Rathore, S., Park, J.H., Chang, H.: Deep learning and blockchain-empowered security framework for intelligent 5g-enabled iot. IEEE access **9**, 90075–90083 (2021)
17. Seni, G., Elder, J.: Ensemble methods in data mining: improving accuracy through combining predictions. Morgan & Claypool Publishers (2010)
18. Siddalingappa, R., Kanagaraj, S.: Anomaly detection on medical images using autoencoder and convolutional neural network. International Journal of Advanced Computer Science and Applications (7) (2021)
19. Uzoka, A., Cadet, E., Ojukwu, P.U.: The role of telecommunications in enabling internet of things (iot) connectivity and applications. Comprehensive Research and Reviews in Science and Technology **2**(02), 055–073 (2024)
20. Zhang, W., Zhou, H., Bao, X., Cui, H.: Outlet water temperature prediction of energy pile based on spatial-temporal feature extraction through cnn–lstm hybrid model. Energy **264**, 126190 (2023)