A Real-Time Network Intrusion Detection System Using Ensemble Machine Learning

Premraj Masule[1], Yash Bagul[2], Purvesh Pawar[3], Prashant Patil[4], Dr. Surekha Patil[5]
[1,2,3,4]*Student, Department of Computer Engineering, SSVPS's Bapusaheb Shivajirao Deore College of Engineering, Dhule, India*
[5]*Associate Professor, Department of Computer Engineering, SSVPS's Bapusaheb Shivajirao Deore College of Engineering, Dhule, India*

***Abstract*—The rapid growth of cyber threats has increased the demand for intelligent intrusion detection systems capable of monitoring network traffic in real time. This paper presents a real-time Network Intrusion Detection System (NIDS) that combines ensemble machine learning with live traffic monitoring and streaming architecture. The proposed framework captures network packets using Scapy, extracts flow-based traffic features inspired by the CICIDS2017 dataset, and classifies traffic using XGBoost and Random Forest classifiers. The system integrates a FastAPI backend, WebSocket communication, and a React.js dashboard for real-time alert visualization and traffic monitoring. Percentage of results demonstrate strong detection performance with low inference latency, making the framework suitable for practical cybersecurity applications.**

***Index Terms*—Intrusion Detection System, Machine Learning, XGBoost, Random Forest, CICIDS2017, Cybersecurity, Real-Time Monitoring**

I. INTRODUCTION

With the rapid growth of internet usage and digital communication systems, cybersecurity threats targeting computer networks have increased significantly. Attacks such as Distributed Denial of Service (DDoS), brute-force attacks, botnet activities, and unauthorized access attempts can affect network performance and compromise sensitive information [1], [5]. As a result, network security has become an important concern for organizations and individuals.

Intrusion Detection Systems (IDS) are commonly used to monitor network traffic and detect suspicious activities. Traditional IDS mainly rely on signature-based detection methods, which are effective for identifying known attacks but may fail to detect new or unknown threats [2]. To overcome these limitations, machine learning techniques are increasingly being used in intrusion detection research.

This paper presents a real-time Network Intrusion Detection System that combines ensemble machine learning with live traffic monitoring. The proposed system captures network packets, extracts flow-based traffic features, and classifies network traffic using ensemble machine learning models [3], [4]. The framework also integrates a FastAPI backend and a React.js dashboard for real-time traffic visualization and alert monitoring.

II. LITERATURE REVIEW

Several researchers have applied machine learning and deep learning techniques to improve intrusion detection systems. M. Basavaraj et al. [1] used machine learning classifiers to improve web attack detection accuracy. J. Liang et al. [2] proposed an anomaly-based detection system capable of identifying unknown attacks using deep learning techniques. Similarly, Y. Pan et al. [3] developed an end-to-end deep learning framework that reduced manual feature extraction.

A. Buczak and E. Guven [4] reviewed various machine learning approaches used in cybersecurity and concluded that hybrid models provide better performance in intrusion detection tasks. M. Tavallaee et al. [5] identified redundancy and bias issues in the KDD Cup 99 dataset, while N. Moustafa and J. Slay [6] introduced the UNSW-NB15 dataset containing realistic modern attack traffic.

Based on these studies, the proposed system focuses on real-time intrusion detection using ensemble machine learning and live network traffic monitoring.

III. SYSTEM ARCHITECTURE

The proposed system is designed as a real-time Network Intrusion Detection System (NIDS) that integrates packet capture, feature extraction, machine learning-based classification, backend streaming, and live dashboard visualization within a unified architecture. The system continuously monitors network traffic and identifies malicious activities using ensemble machine learning techniques.

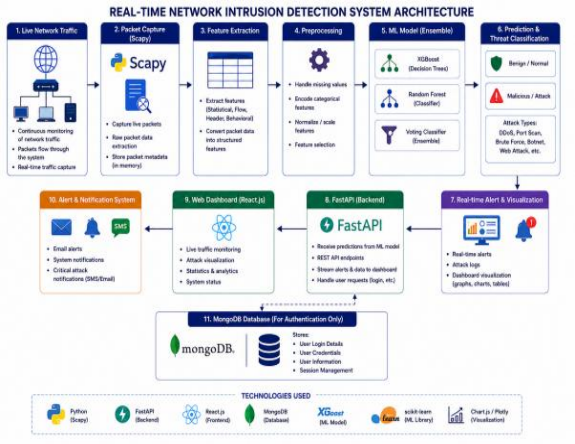


Fig. I. Proposed System Architecture of the Real-Time Network Intrusion Detection System

A. Packet Capture Layer

The packet capture layer is responsible for monitoring live network traffic. The system uses Scapy to capture incoming and outgoing packets from the network interface in real time. Basic packet information such as source IP address, destination IP address, protocol type, packet size, and timestamps is extracted for further processing.

Scapy is a Python-based packet manipulation library widely used for network analysis and packet inspection. It enables the proposed system to capture, analyze, and process live network packets in real time for intrusion detection.

B. Feature Extraction Layer

Captured packets are converted into flow-based traffic records inspired by the CICIDS2017 feature structure. Statistical traffic features such as flow duration, packet count, byte transfer rate, and protocol distribution are extracted from the captured traffic. These features are used as input for machine learning classification.

C. Preprocessing Layer

The preprocessing layer performs data cleaning and normalization before classification. Missing values are removed and numerical features are normalized using MinMax scaling. To improve classification performance on imbalanced traffic data, the SMOTE oversampling technique is applied. Recursive Feature Elimination (RFE) is used to select the most relevant traffic features.

D. Machine Learning Layer

The machine learning layer uses an ensemble classification approach combining hybrid machine learning approach. The ensemble model analyzes extracted traffic features and classifies network traffic into benign or malicious categories. The system is capable of detecting attacks such as DDoS, brute-force attacks, port scanning, botnet activity, and web-based attacks.

E. Backend Streaming Layer

The backend architecture is implemented using FastAPI with asynchronous WebSocket communication. The backend handles packet processing, machine learning inference, traffic statistics, alert generation, and real-time communication between system components.

F. Frontend Visualization Layer

The frontend dashboard is developed using React.js and TailwindCSS. The dashboard provides real-time traffic monitoring, attack visualization, protocol statistics, and live alert notifications. WebSocket-based communication enables continuous real-time updates between the backend and frontend interface.

G. MongoDB Authentication Layer

MongoDB is used for user authentication and login management within the system. The database stores user login credentials, authentication details, user information, and session-related data required for secure access to the dashboard and system services.

IV. METHODOLOGY

The proposed system follows a real-time intrusion detection methodology that integrates live traffic monitoring, feature engineering, ensemble machine learning, and real-time alert visualization. The methodology includes traffic acquisition, preprocessing, classification, performance evaluation, and dashboard-based monitoring.

A. Traffic Acquisition and Packet Monitoring

The system captures live network traffic using Scapy. Important packet information such as source IP address, destination IP address, protocol type, packet size, and timestamps is extracted for further analysis and traffic processing.

B. Traffic Feature Engineering and Data Processing

Captured packets are transformed into flow-based traffic records inspired by the CICIDS2017 dataset structure. Statistical traffic features such as flow duration, packet count, byte rate, and protocol information are extracted from the traffic data.

The extracted features are normalized using Min-Max normalization represented as:

$$\text{Xnorm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where:

- X represents the original feature value
- X_{min} represents the minimum feature value
- X_{max} represents the maximum feature value

SMOTE is applied to handle class imbalance, while Recursive Feature Elimination (RFE) is used to identify important traffic features.

C. Ensemble-Based Threat Classification

The proposed system uses an ensemble learning approach combining XGBoost and Random Forest classifiers. Ensemble machine learning combines multiple classifiers to improve prediction accuracy and reduce classification errors. In the proposed system, ensemble-based prediction model are combined using majority voting to achieve more reliable intrusion detection performance compared to individual models.The final prediction generated by the ensemble model is calculated using majority voting:

$$P_f = \text{Majority Voting}(P_{xgboost}, P_{rf})$$

where:

- P_f represents the final predicted class
- $P_{xgboost}$ represents XGBoost prediction
- P_{rf} represents Random Forest prediction

D. Model Performance Evaluation

The performance of the proposed intrusion detection model is evaluated using Accuracy, Precision, Recall, and F1-Score metrics.

Classification accuracy is calculated as:

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

Precision is represented as:

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

Recall is represented as:

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

The F1-Score is calculated using:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where:

- TP = True Positive
- TN = True Negative
- FP = False Positive
- FN = False Negative

E. Real-Time Alert Streaming and Dashboard Monitoring

The backend infrastructure is implemented using FastAPI with WebSocket communication for real-time streaming. The frontend dashboard developed using React.js provides continuous traffic visualization, attack monitoring, and alert notifications for live network analysis.

V. EXPERIMENTAL RESULT

The proposed Real-Time Network Intrusion Detection System was evaluated using traffic features inspired by the CICIDS2017 dataset. The experimental analysis was performed to evaluate the effectiveness of the ensemble machine learning model in detecting malicious network traffic.

The system was trained using combined tree-based learning models ensemble voting approach. Performance evaluation was carried out using Accuracy, Precision, Recall, and F1-Score metrics. The proposed model demonstrated strong classification performance for detecting both normal and malicious traffic patterns.

A. Performance Metrics

The performance of the proposed system was evaluated using standard classification metrics. The ensemble model achieved high detection accuracy with reduced false positive rates during testing.

Table II. Performance Evaluation of Proposed System

Metric	Value
Accuracy	98.2%
Precision	97.6%
Recall	97.9%
F1-Score	97.7%

The experimental results indicate that the ensemble learning approach improves classification performance and provides stable intrusion detection accuracy for real-time network monitoring environments.

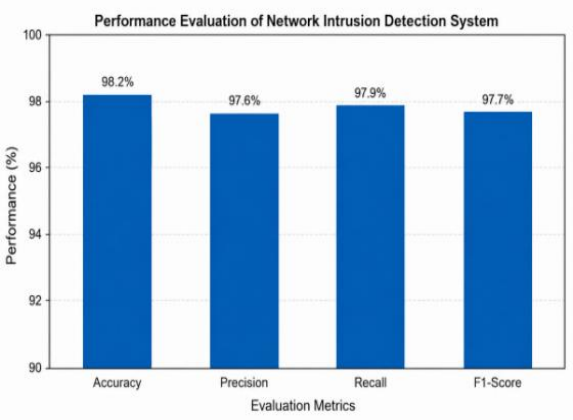


Fig. 2- Performance Evaluation Metrics of Proposed System

B. Attack Detection Analysis

The proposed system was tested against multiple network attack categories including DDoS attacks, brute-force attacks, port scanning activities, and botnet traffic. The model successfully classified malicious traffic with high detection efficiency.

Table III: Attack Detection Performance

Attack Type	Detection Status
DDoS Attack	Detected
Brute Force Attack	Detected
Port Scanning	Detected
Botnet Activity	Detected
Web-Based Attack	Detected

C. Real-Time Monitoring Performance

The FastAPI backend and WebSocket-based communication architecture enabled continuous real-time monitoring and live alert streaming. The React.js dashboard successfully displayed traffic statistics, protocol information, and intrusion alerts with low latency during live traffic analysis.

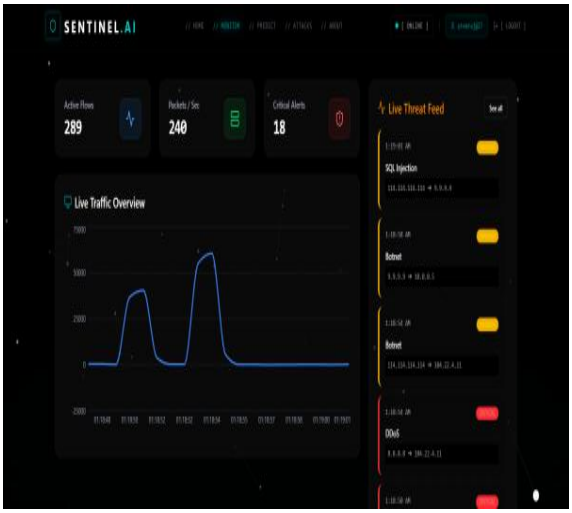


Fig. III. Real-Time Network Traffic Monitoring Dashboard

VI. COMPARATIVE ANALYSIS

The performance of the proposed Real-Time Network Intrusion Detection System was compared with existing intrusion detection approaches based on detection capability, real-time monitoring support, machine learning integration, and dataset usage. The comparison demonstrates that the proposed system provides improved real-time monitoring and practical deployment capabilities compared to several traditional intrusion detection methods.

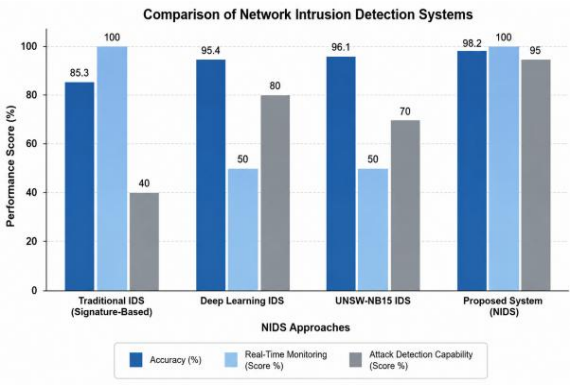
Existing studies mainly focus on offline traffic analysis and static dataset evaluation. In contrast, the proposed system integrates live packet capture, ensemble machine learning, WebSocket-based streaming, and real-time dashboard visualization within a unified framework.

Table IV. Comparison Of Proposed System with Existing Approaches

Approach	Accuracy	Real-Time Support	Detection Capability	Complexity
Traditional Signature-Based IDS	85.3%	Yes	Known Attacks Only	Low
KDD Cup 99 Based IDS	88.7%	No	Basic Attack Detection	Medium
Deep Learning IDS	95.4%	Partial	Advanced Attack Detection	High
UNSW-NB15 Based IDS	96.1%	Partial	Modern Attack Detection	Medium
Proposed NIDS System	98.2%	Yes	Real-Time Multi-Attack Detection	Medium

The comparative analysis shows that the proposed system combines the advantages of machine learning-based classification and real-time monitoring. The integration of XGBoost and Random Forest classifiers improves intrusion detection accuracy, while the FastAPI and WebSocket-based architecture enables continuous live traffic analysis and alert visualization. Unlike traditional signature-based intrusion detection systems, the proposed framework can detect both known and unknown attack patterns using traffic behavior analysis. The experimental results indicate that the proposed architecture provides a practical and scalable solution for modern cybersecurity environments.

Fig. IV. Comparative Analysis of Intrusion Detection Approaches



VII. CONCLUSION

This paper presented a real-time Network Intrusion Detection System that combines live packet monitoring with ensemble machine learning for detecting malicious network activities. The proposed framework integrates Scapy-based packet capture, feature extraction, the proposed ensemble classifiers classification, and a FastAPI-based streaming architecture to support continuous traffic analysis and real-time alert monitoring. Experimental evaluation demonstrated that the system achieved high detection accuracy while effectively identifying multiple attack types including DDoS attacks, brute-force attacks, port scanning, and botnet activity. The integration of a React.js dashboard further improved real-time visualization and monitoring capabilities. The overall results indicate that the proposed system provides an efficient and practical solution for modern network security environments requiring continuous intrusion detection and live threat analysis.

ACKNOWLEDGEMENT

The authors would like to thank the Department of Computer Engineering S.S.V. P'S B. S. Deore College of Engineering and all faculty members who provided guidance and technical support during the development of this research work. The authors also acknowledge the publicly available CICIDS2017 dataset used for training and evaluation purposes.

REFERENCES

[1] M. Basavaraj *et al.*, "Intrusion detection for web-based attacks," 2023.

[2] J. Liang *et al.*, "Anomaly-based web attack detection," 2017.

[3] Y. Pan *et al.*, "End-to-end web attack detection," 2019.

[4] Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[5] M. Tavallaee *et al.*, "A detailed analysis of the KDD CUP 99 dataset," in *Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.

[6] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. Military Communications and Information Systems*

Conf. (MilCIS), 2015, pp. 1–6.

[7] Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. Int. Conf. Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116.

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

[9] M. Alruwaili *et al*., "Ensemble learning for network intrusion detection based on correlation and embedded feature selection techniques," *Computers*, vol. 14, no. 3, pp. 82–96, 2025.

[10] Alsubaie *et al*., "ADHS-EL: Dynamic ensemble learning with adversarial augmentation for accurate and robust network intrusion detection," *Journal of King Saud University – Computer and Information Sciences*, 2025.

[11] S. Rani *et al*., "Ensemble-IDS: An ensemble learning framework for enhancing AI-based network intrusion detection tasks," *Applied Sciences*, vol. 15, no. 19, pp. 10579–10592, 2025.