An Integrated Cyber Threat Intelligence and Intrusion Detection System Using Machine Learning with Real-Time Monitoring and Response

Mr. A. Siva[1], Mr. S. Derikson[2]
[1]*Assistant Professor, Dept. of CSE, Jayaraj Annapackiam CSI College of Engineering, Nazareth, India*
[2]*PG Scholar, Dept. of CSE, Jayaraj Annapackiam, CSI College of Engineering, Nazareth, India*

***Abstract*—The growing sophistication of cyberattacks against modern network infrastructures has exposed critical limitations in traditional signature-based and rule-based intrusion detection systems (IDS). This paper presents an integrated Cyber Threat Intelligence and Intrusion Detection System (CTI-IDS) that combines real-time network packet capture, machine learning-based traffic classification, and a comprehensive suite of cybersecurity response and reporting modules. The proposed system leverages NSL-KDD-style network traffic features and evaluates four machine learning algorithms — Random Forest, XGBoost, Voting Classifier, and Isolation Forest — for classifying network traffic into five categories: Normal, Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). Isolation Forest achieves the highest accuracy of 79.86% and an F1-score of 79.92%. Beyond classification, the system integrates automated alert generation, IP blocking, YARA-based file scanning, IP reputation checking, an AI-powered cybersecurity chatbot, and automated PDF report generation — all accessible through a Flask-based real-time web dashboard. Results confirm the viability of machine learning in real-time intrusion detection and demonstrate the practical benefits of integrating threat intelligence with automated network defense.**

***Index Terms*—Intrusion Detection System; Cyber Threat Intelligence; Machine Learning; Network Security; Random Forest; XGBoost; Isolation Forest; Real-Time Monitoring; Flask Dashboard; YARA File Scanning.**

I. INTRODUCTION

The rapid proliferation of digital infrastructure — spanning cloud computing, the Internet of Things (IoT), mobile networks, and enterprise systems — has dramatically expanded the attack surface available to malicious actors. Cybercriminals exploit vulnerabilities in network environments to conduct unauthorized access, data exfiltration, denial-of-service attacks, and privilege escalation, causing significant financial and operational harm to organizations worldwide. Conventional security mechanisms, including firewalls, antivirus software, and signature-based intrusion detection systems, have proven insufficient against modern threats that employ obfuscation, polymorphism, zero-day exploits, and adaptive evasion strategies.

Machine learning (ML) has emerged as a transformative approach to intrusion detection, enabling systems to learn behavioral patterns from historical data and classify anomalous traffic without relying on static rule sets. ML-based IDS offer superior adaptability to novel attack vectors, scalability to high-traffic environments, and the capacity to detect complex, multi-stage attacks that evade traditional detection methods.

Despite substantial academic progress in ML-based intrusion detection, a significant gap persists between research prototypes focused on classification accuracy and practical, deployable cybersecurity platforms. Most existing systems are validated against benchmark datasets in offline settings and lack real-time monitoring, threat response, user-facing interfaces, or integrated threat intelligence.

This paper presents a CTI-IDS that bridges this gap. The key contributions are:

- A comprehensive, modular CTI-IDS architecture unifying real-time detection with threat response and reporting.

- Comparative evaluation of four ML models — Random Forest, XGBoost, Voting Classifier, and Isolation Forest — on NSL-KDD-style traffic data.
- Integration of threat intelligence modules: YARA-based file scanning, IP reputation checking, and an AI-powered cybersecurity chatbot.
- A Flask-based real-time dashboard for centralized monitoring, alert management, IP blocking, and automated PDF report generation.

II. RELATED WORK

Research in intelligent intrusion detection has progressed significantly, transitioning from rule-based methods toward adaptive, data-driven approaches. Althunayyan et al. [1] surveyed learning-based IDS for in-vehicle networks, highlighting behavioral detection superiority over static methods. Yang et al. [2] reviewed deep reinforcement learning for IDS, demonstrating adaptive decision-making potential in dynamic environments where attack patterns evolve continuously.

Alghamdi and Keshta [3] examined federated learning with blockchain for smart home IDS, enabling model training without sharing sensitive user data. Maimó et al. [4] investigated deep learning-based anomaly detection for 5G networks, emphasizing scalable, low-latency detection requirements. Singh and Agarwal [5] proposed hybrid deep transfer learning for electric vehicular networks, demonstrating improved detection through knowledge transfer.

Qiqieh et al. [6] combined swarm-optimized ML for cyber threat detection. Markkandeyan et al. [7] proposed a hybrid deep learning model with optimization algorithms. Ali et al. [8] compared deep learning and ML for IDS, finding classical ML competitive with lower resource requirements on structured data. Ahmed et al. [14] combined signature-based detection with ML and fuzzy clustering. Rai et al. [15] addressed CAN bus IDS using deep learning for vehicular safety.

While these studies make notable algorithmic contributions, few present a unified platform integrating real-time packet monitoring, multi-model classification, alert management, IP blocking, file scanning, IP reputation analysis, chatbot assistance, and automated reporting. The proposed CTI-IDS addresses these gaps.

III. SYSTEM ARCHITECTURE AND DESIGN

A. Overview

The proposed CTI-IDS adopts a modular, layered architecture comprising thirteen functional modules organized into four logical tiers: (1) data acquisition, (2) processing and classification, (3) response and intelligence, and (4) visualization and reporting. Fig. 1 illustrates the complete system architecture and end-to-end data flow.

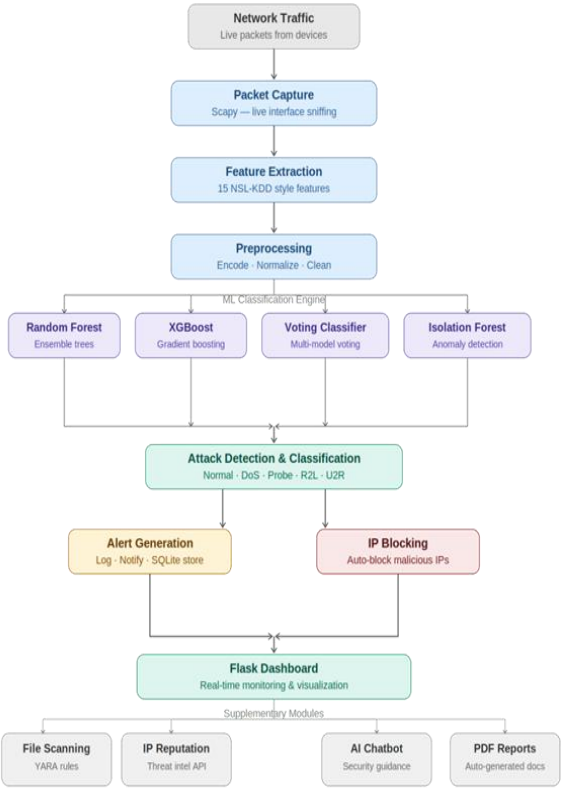


Fig. 1. CTI-IDS System Architecture

B. Block Diagram

Fig. 2 presents the nine-stage linear processing pipeline of the CTI-IDS system, from raw network traffic ingestion through to the final dashboard and report output. Each stage represents a distinct functional module that transforms, analyzes, or responds to network data in sequence.

Fig. 2. CTI-IDS Block Diagram

C. Network Packet Capture

The data acquisition tier employs Scapy, a Python-based packet manipulation library, to intercept live traffic from active network interfaces. Captured packets yield raw attributes including source and destination IP addresses, protocol type, port numbers, packet size, connection flags, and service identifiers. These attributes form the foundation for subsequent feature extraction and classification.

D. Feature Extraction and Preprocessing

Raw packet data is transformed into a structured feature vector aligned with NSL-KDD intrusion detection conventions. Fifteen features are selected for their discriminative power in differentiating normal from malicious traffic, as detailed in Table I. Fig. 3 illustrates the complete five-step preprocessing workflow applied before model training and real-time prediction.

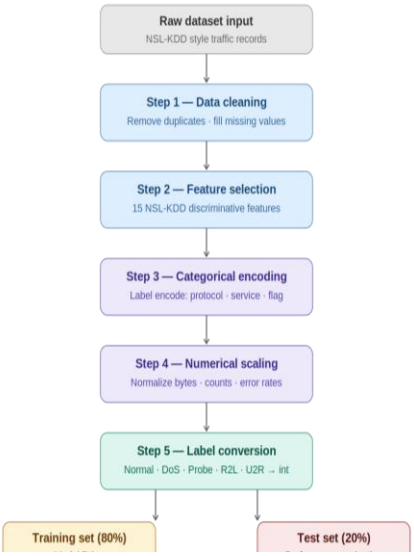


Fig. 3. Preprocessing Workflow

TABLE I. Selected Features for Intrusion Detection

Feature Name	Description
Duration	Time duration of the network connection in seconds
Protocol Type	Communication protocol used: TCP, UDP, or ICMP
Service	Network service accessed during the connection
Flag	Status flag indicating the connection state
Src_Bytes	Number of bytes transmitted from source to destination
Dst_Bytes	Number of bytes transmitted from destination to source
Logged_In	Boolean indicator of successful login activity
Count	Number of connections to the same host in a time window
Srv_Count	Number of connections to the same service in a time window
Serror_Rate	Proportion of connections with SYN errors
Srv_Serror_Rate	SYN error rate for same-service connections
Rerror_Rate	Proportion of rejected connection attempts
Srv_Rerror_Rate	Rejection error rate for same-service connections
Same_Srv_Rate	Proportion of connections using the same service
Dst_Host_Srv_Count	Number of service-level connections to the destination host

E. Machine Learning Classification Engine

The classification tier evaluates four ML algorithms selected for their complementary detection capabilities. Random Forest is an ensemble of decision trees trained via bagging, offering robustness against overfitting on structured tabular data. XGBoost is a gradient boosting framework that sequentially minimizes classification error, demonstrating strong performance under class imbalance. The Voting Classifier aggregates predictions from multiple base models via majority voting, improving reliability by reducing the impact of individual model errors. Isolation Forest is an unsupervised anomaly detection algorithm that isolates outliers by recursively partitioning the feature space, providing complementary detection of unknown and zero-day threats without requiring labeled attack samples.

F. Alert Generation and Threat Response

Upon detecting malicious traffic, the alert module constructs a structured security event record containing the attack category, source and destination IP, timestamp, protocol, confidence score, and top contributing features. Events are persisted to a SQLite database and broadcast in real-time via Flask-SocketIO to the web dashboard. The IP blocking

module evaluates detected source IPs against configurable severity thresholds; IPs meeting auto-block criteria are immediately blocked from further communication, with human-review options retained for borderline cases.

G. Supplementary Threat Intelligence Modules

File Scanning: Uploaded files are analyzed using YARA rules and heuristic pattern matching to detect malware signatures and suspicious executables. IP Reputation Checking: Suspicious IPs are queried against external threat intelligence services to enrich blocking and alerting decisions. AI-Powered Chatbot: A conversational assistant provides users with attack type explanations, mitigation strategies, and cybersecurity best practice guidance. Automated PDF Reporting: The report generation module compiles attack summaries, detection statistics, blocked IP lists, and visual analytics into structured PDF documents for auditing and compliance.

IV. METHODOLOGY

A. Dataset

The system is trained and validated using features derived from the NSL-KDD dataset, a widely adopted benchmark in intrusion detection research. NSL-KDD addresses key limitations of KDD Cup 1999 including duplicate records and severe class imbalance. The dataset contains labeled network connection records spanning five traffic categories, as summarized in Table II.

TABLE II. Attack Categories and Sample Distribution

Category	Description	Approx. Samples
Normal	Legitimate network traffic without malicious behavior	~67,000
DoS	Denial-of-Service — resource exhaustion attacks	~46,000
Probe	Network scanning and reconnaissance activity	~11,500
R2L	Remote-to-Local unauthorized access attempts	~1,100
U2R	User-to-Root privilege escalation attacks	~50

B. Model Training and Evaluation

The dataset is partitioned into training (80%) and test (20%) sets using stratified sampling to preserve class proportions across all five categories. Random Forest is configured with 100 estimators; XGBoost employs a learning rate of 0.1 with maximum tree depth of 6; the Voting Classifier uses soft voting over Random Forest and XGBoost; Isolation Forest is trained with a contamination factor matching the proportion of attack samples in the dataset.

Model performance is evaluated using accuracy, precision, recall, F1-score, and confusion matrix analysis. Accuracy measures overall correct classification; precision quantifies false alarm minimization; recall measures attack detection completeness; F1-score provides a harmonic balance between precision and recall, particularly relevant for the severely imbalanced U2R and R2L categories.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Model Performance Comparison

Table III presents the comparative performance of all four models on the held-out test partition. All models achieve accuracy above 74%, confirming that the selected NSL-KDD features provide sufficient discriminative information for network intrusion detection. Isolation Forest achieves the highest overall performance with accuracy of 79.86% and F1-score of 79.92%, demonstrating effective anomaly isolation of attack-class traffic distributions.

TABLE III. Model Performance Comparison

Model	Accuracy (%)	F1-Score (%)
Random Forest	74.59	69.93
XGBoost	77.88	73.24
Voting Classifier	77.20	72.35
Isolation Forest	79.86	79.92

B. Analysis of Results

XGBoost and the Voting Classifier demonstrate closely matched performance, with XGBoost slightly outperforming due to its sequential error-correction mechanism. Random Forest's lower F1-score (69.93%) reflects greater sensitivity to class imbalance in the U2R and R2L categories, which contain very few samples relative to DoS and Normal traffic. The

gap between training accuracy (approximately 82–86%) and validation accuracy (74–80%) across all supervised models indicates moderate overfitting, suggesting that additional regularization, hyperparameter tuning, or class-balancing techniques such as SMOTE could improve generalization.

Isolation Forest's high F1-score highlights the value of anomaly-based detection. Its superior performance suggests the attack classes exhibit sufficiently distinct distributional characteristics from normal traffic that anomaly isolation is an effective detection strategy. However, unsupervised methods lack the ability to classify specific attack types, necessitating integration with supervised classifiers for actionable threat categorization — a design principle reflected in the platform's multi-model architecture.

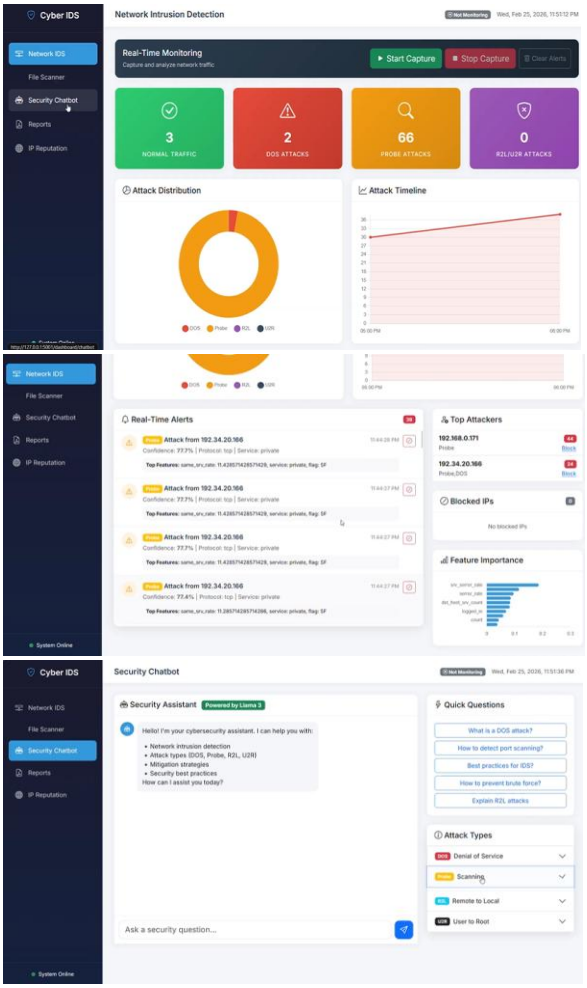
C. Comparison with Existing Systems

Table IV compares the proposed CTI-IDS against conventional IDS paradigms across key operational dimensions. The proposed system surpasses both conventional IDS and most research prototypes, offering an integrated, operationally viable platform suitable for academic research, SME deployment, and security education environments.

TABLE IV. Proposed System vs. Existing IDS Approaches

Feature	Existing IDS	Proposed CTI-IDS
Detection Method	Signature / rule-based	Multi-model ML classification
Real-Time Monitoring	Limited	Live packet capture via Scapy
Attack Classification	Known attacks only	Normal, DoS, Probe, R2L, U2R
IP Blocking	Manual / limited	Automated with thresholds
File Scanning	Rarely integrated	YARA-based file scanning
IP Reputation	Separate tool	Integrated into dashboard
User Assistance	Not available	AI-powered chatbot
Report Generation	Manual documentation	Automated PDF generation

Results



VI. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented a Cyber Threat Intelligence and Intrusion Detection System that integrates real-time network packet capture, machine learning-based attack classification, and a comprehensive suite of cybersecurity response modules within a unified Flask-based web platform. Four machine learning models were trained and evaluated on NSL-KDD-style features. Isolation Forest achieved the highest accuracy (79.86%) and F1-score (79.92%), while XGBoost and the Voting Classifier demonstrated competitive supervised classification performance.

The system's modular design — encompassing file scanning, IP reputation analysis, automated IP blocking, AI chatbot assistance, and PDF report generation — addresses a critical gap in existing literature by providing an end-to-end operational

cybersecurity platform rather than an isolated algorithmic contribution. The integrated block diagram, system architecture, and preprocessing workflow presented in this paper demonstrate a practical and reproducible framework for real-time intrusion detection.

B. Limitations

The NSL-KDD dataset does not fully capture the complexity of contemporary network environments including encrypted traffic, cloud-native attacks, IoT botnets, and advanced persistent threats. Model accuracy in the 74–80% range falls below production security requirements. YARA-based file scanning is constrained by known signature coverage and may fail to detect novel malware variants. The IP blocking mechanism requires refinement to prevent inadvertent blocking of legitimate users sharing IP addresses through proxies or NAT.

C. Future Work

Future enhancements include: (1) retraining on contemporary datasets such as CIC-IDS2017, UNSW-NB15, and BotIoT to improve generalization to modern attack patterns; (2) integrating LSTM and Transformer-based deep learning models for sequential traffic analysis; (3) incorporating Explainable AI (XAI) techniques for interpretable model predictions; (4) implementing role-based access control, multi-factor authentication, and audit logging for production hardening; (5) extending the platform to distributed, cloud-native deployment with multi-sensor network coverage; and (6) enhancing the IP blocking module with risk scoring, severity grading, and administrator approval workflows.

REFERENCES

[1] M. Althunayyan, A. Javed, and O. Rana, "A Survey of Learning-Based Intrusion Detection Systems for In-Vehicle Networks," *Computer Networks*, p. 112031, 2026.

[2] W. Yang, A. Acuto, Y. Zhou, and D. Wojtczak, "A Survey for Deep Reinforcement Learning Based Network Intrusion Detection," Applied AI Letters, vol. 7, no. 2, p. e70026, 2026.

[3] A. Alghamdi and I. Keshta, "Blockchain Consensus Mechanisms and Enhancement Techniques for Federated Learning-Based IDS in IoT Smart Homes," Journal of Reliable and Secure Computing, vol. 2, no. 1, pp. 1–26, 2026.

[4] L. F. Maimó et al., "Dynamic Management of a Deep Learning-Based Anomaly Detection System for 5G Networks," arXiv preprint arXiv:2601.15177, 2026.

[5] N. Singh and R. Agarwal, "Hybrid Net: Enhanced DTL-Based IDS for Electric Vehicular Network," Peer-to-Peer Networking and Applications, vol. 19, no. 1, p. 1, 2026.

[6] I. Qiqieh et al., "An Intelligent Cyber Threat Detection: A Swarm-Optimized Machine Learning Approach," Alexandria Engineering Journal, vol. 115, pp. 553–563, 2025.

[7] S. Markkandeyan et al., "Novel Hybrid Deep Learning Based Cyber Security Threat Detection Model with Optimization Algorithm," Cyber Security and Applications, vol. 3, Art. no. 100075, 2025.

[8] M. L. Ali et al., "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study," Applied Sciences, vol. 15, no. 4, p. 1903, 2025.

[9] N. Alkhafaji, T. Viana, and A. Al-Sherbaz, "Integrated Genetic Algorithm and Deep Learning Approach for Effective Cyber-Attack Detection in IIoT Environments," Arabian Journal for Science and Engineering, vol. 50, no. 15, pp. 12071–12095, 2025.

[10] A. K. S. Ali et al., "Intelligent Intrusion Detection and Data Protection Using AI and ML Techniques," Spectrum of Engineering Sciences, pp. 818–828, 2025.

[11] S. Jamshidi et al., "Application of Deep Reinforcement Learning for Intrusion Detection in IoT: A Systematic Review," Internet of Things, vol. 31, Art. no. 101531, 2025.

[12] K. S. Yerneni et al., "Towards Proactive Cloud Security: A Survey on ML and DL-Based Intrusion Detection Systems," JCETAI, 2025.

[13] T. S. Oyinloye, M. O. Arowolo, and R. Prasad, "Enhancing Cyber Threat Detection with an Improved ANN Model," Data Science and Management, vol. 8, no. 1, pp. 107–115, 2025.

[14] U. Ahmed et al., "Signature-Based Intrusion Detection Using ML and Deep Learning Approaches Empowered with Fuzzy Clustering," Scientific Reports, vol. 15, no. 1, Art. no. 1726, 2025.

[15] R. Rai et al., “Securing the CAN Bus Using Deep Learning for Intrusion Detection in Vehicles,” Scientific Reports, vol. 15, no. 1, Art. no. 13820, 2025.