International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

ShieldAI – Intrusion Prevention System

Manikrao Dhore[1], Suvidha Dhakane[2], Atharva Dhakate[3], Pranav Bhawari[4], Shriram Dindore[5]

Computer Engineering Department, Vishwakarma Institute of Technology, Pune, INDIA

Emails: manikrao.dhore@vit.edu, suvidha.dhakane22@vit.edu,atharva.dhakate22@vit.edu, pranav.bhawari22@vit.edu,shriram.dindore22@vit.edu

Abstract

In today's world of internet, safeguarding network infrastructure against malicious activities is crucial. This project is real-time threat detection that leverages machine learning and AI to detect and classify cyber-attacks based on live packet capturing. Utilizing Scapy for packet capturing and FastApi as a backend, the system extracts 71 features from packet and feeds them into a pre-trained Autoencoder and DNN based model to accurately predict potential threats such as DDoS, Port Scans, Brute Force, SQL Injection, and more. The detection results are streamed to a dynamic React- based frontend providing real-time threat visualization and monitoring. The system handles such attacks by IP blocking for malicious sources. So, this system will enhances proactive defense mechanisms and provide scalable solution for modern cybersecurity needs.

Index Terms—Machine Learning, Deep Learning, Network Security, Scapy, Real-time Packet Analysis, FastApi, Python, React

Introduction

With the exponential growth of network-connected systems, the risk and sophistication of cyber threats have also increased significantly. From personal computers to enterprise-level infrastructures, digital systems are constantly exposed to various attack vectors such as Denial-of-Service (DoS), probing, and packet spoofing. Conventional security mechanisms, particularly signature-based intrusion detection systems (IDS), are often limited by their de- pendency on predefined attack patterns and their inability to detect zero-day threats. These limitations have created a demand for intelligent, adaptive systems capable of identifying novel intrusions in real time. In response to these challenges, we present an advanced real- time network intrusion detection system (NIDS) that combines machine learning, real-time

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

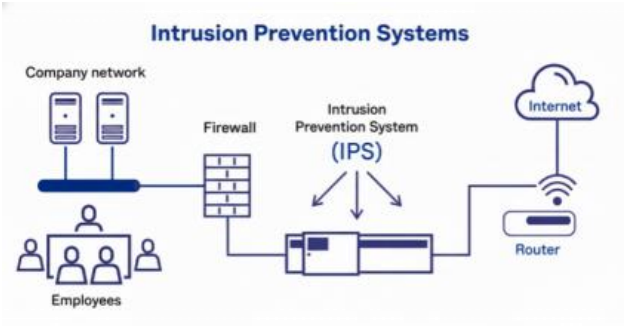

Figure 1: intrusion Prevention system

packet analysis, and live data visualization. This system is designed as a fully local, privacy-conscious solution that continuously monitors network traffic, detects anomalies, and visually represents threat levels without transmitting any sensitive packet data to external servers. At the core of system is a FastAPI backend that leverages Scapy for efficient packet sniffing and a Keras-based Autoencoder for anomaly detection. The Autoencoder is trained on be- nign traffic patterns to recognize deviations that may indicate malicious behavior. Once a packet is captured and its features are extracted, the model evaluates it for potential threats based on reconstruction error thresholds. Detected anomalies are immediately broadcast to the frontend using WebSocket protocols, allowing users to receive alerts and visual feedback with minimal latency.

The frontend, developed using React and Chart.js, features a dynamic dashboard that dis- plays incoming predictions and a real-time threat level chart. It also includes a live-updating packet log table, enabling end users to monitor traffic patterns and anomaly classifications as they occur. The integration of Matplotlib on the backend ensures that statistical sum- maries and trends can be visualized and served efficiently via HTTP or streamed directly to the dashboard. This system emphasizes a modular, extensible architecture that facilitates seamless deployment on local machines, virtual environments, or embedded systems. By keeping all analysis and detection logic on the client side, this system ensures user privacy and enhances operational transparency. Furthermore, the system's lightweight footprint and real-time capabilities make it suitable for both research and practical deployment in home or small enterprise networks. In this paper, we detail the design and implementation of this system, outline the methodology for feature extraction and anomaly detection, and present qualitative results that demonstrate its effectiveness in real-time intrusion detection sce- narios. Our results suggest that machine learning-based IDS solutions can provide a more adaptive and proactive defense layer against evolving cyber threats.

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

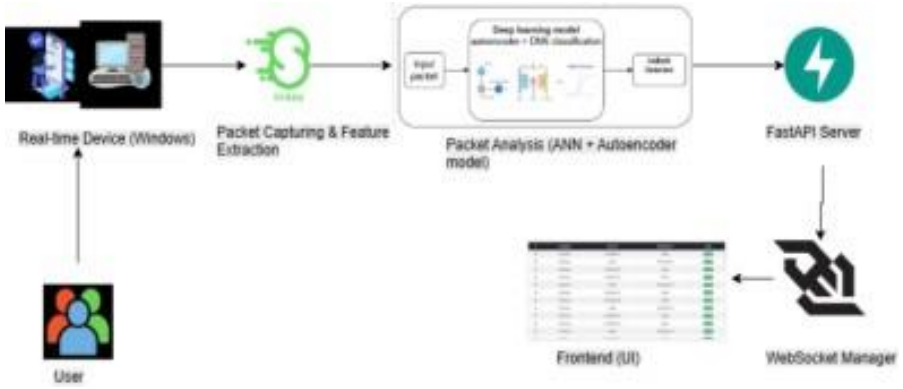

Figure 2: system diagram

Architecture

Literature Survey

Recent advancements in artificial intelligence (AI) and machine learning (ML) have signifi- cantly strengthened intrusion detection systems (IDS), enabling improved detection accuracy and adaptability in cybersecurity. This review analyzes six key studies that employ diverse AI and ML techniques to enhance network intrusion detection and real-time threat mitiga- tion. Study

By [1] Hongpo Zhang et al. utilized a Deep Autoencoder (DAE) for feature selection and a Multilayer Perceptron (MLP) for traffic classification. Their approach enabled efficient real-time intrusion detection, achieving an impressive 98.80

In [2], Usman Musa et al. from Sharda University compared SVM, Decision Trees, and k-NN models through single, hybrid, and ensemble configurations. The study found that hybrid and ensemble approaches improved accuracy but at the expense of increased computational costs and higher false-positive rates. Moreover, reliance on outdated datasets limited real-world applicability, emphasizing the need for updated benchmark data. Study

by [3] Sinem Osken et al. from İstanbul Kültür University investigated the performance of deep neural architectures such as ANN, CNN, RNN, LSTM, and Autoencoders for intrusion detection. The deep learning-based models effectively captured complex attack patterns and achieved 91

by [4], Kim et al. proposed an entropy-based analysis combined with ML algorithms for detecting encryption-related malicious behaviors. Their system achieved 87Yuansheng Dong et al.

A [5] explored a deep learning framework integrating AlexNet and Autoencoders for real-time cyberattack prediction. This hybrid model successfully prevented multiple intrusion

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

attempts with 94.32Finally, Vinayakumar R. et al.

A [6] introduced a comprehensive approach combining Decision Trees, Random Forests, SVMs, DNNs, and SHIA across multiple public datasets. Their ensemble method achieved 95–97

Common trends across these studies reveal that hybrid and deep learning models outperform traditional ML approaches in detecting sophisticated and previously unseen attacks. Nevertheless, challenges such as high computational costs, dataset imbalance, and overfitting persist. Future research should focus on developing lightweight, adaptive, and explainable IDS models that maintain high accuracy while minimizing false positives and computational overhead

Tables

Table 1: Literature Review

Sr.no	Authors	Methodology	Results	Limitations
1	Hongpo Zhang et al.	DAE + MLP	98.8%	Dataset imbalance; High training cost
2	Usman Musa et al.	SVM, DT, kNN	High FP	Expensive; Outdated dataset
3	Sinem Osken et al.	ANN, CNN, RNN	91% precision	Needs large training data
4	Kim et al.	Entropy + ML	87%	Post-encryption only
5	Yuansheng Dong et al.	AlexNet + AE	94.32%	High compute; Overfitting risk
6	Vinayakumar R. et al.	DT, RF, SVM, DNN	95–97%	High resource requirement

Proposed Methodology

The proposed system is a real-time, machine learning-based network intrusion detection system (NIDS) designed to monitor and analyze network traffic locally. This projeect combines packet sniffing, feature extraction, anomaly detection using an autoencoder neural network, and real-time visualization to offer a proactive solution against cyber threats. The architec- ture of the system is modular and consists of four major components: packet acquisition, feature extraction, anomaly detection, and live reporting.

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

A. Packet Acquisition

Efficient and continuous packet capture forms the backbone of this system. To achieve high-performance monitoring, this system uses the Scapy library, a powerful packet manipulation tool written in Python. Rather than capturing packets in a blocking or sequential fashion, this system employs AsyncSniffer, which enables asynchronous, non-blocking packet sniffing. This is crucial for real-time systems as it allows simultaneous processing and acquisition. The sniffer is configured to listen on a selected network interface (e.g., eth0 or Wi-Fi) and capture

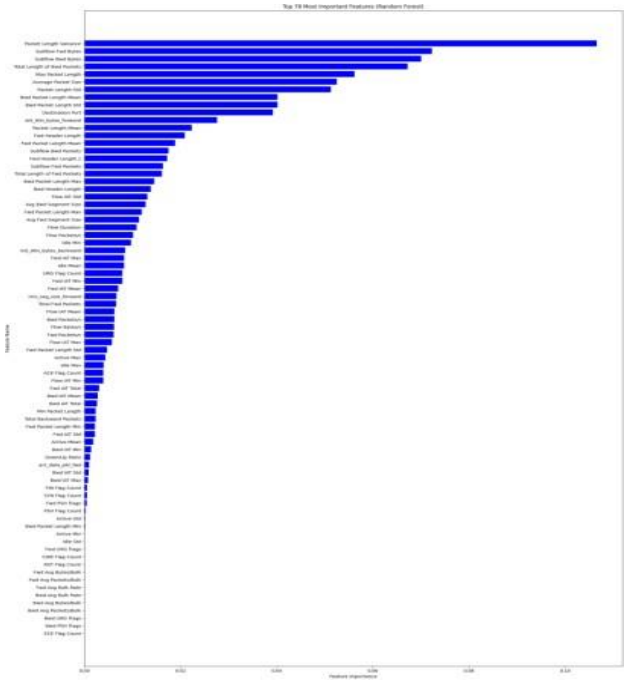

Figure 3: Enter Caption

live traffic. This system applies filters to discard irrelevant traffic such as multicast, ARP, or broadcast packets that are not useful for intrusion detection. The captured packets are sent to a feature extraction module via an internal processing queue, ensuring separation of concerns and better concurrency handling.

B. Feature Extraction

Feature extraction is the process of converting raw network packets into structured, meaningful metrics that can be analyzed by a machine learning model. In this system, this step plays a critical role in intrusion detection because raw packets are too low-level for models to understand directly. This system performs robust feature engineering by extracting 71 statistical and behavioral features from live network packet flows. These features are derived in real time from captured traffic using the Scapy library, and they represent critical

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

indicators of network behavior, including timing metrics, packet sizes, flow durations, TCP flag counts, and direction-based activity.

C. Real-Time Detection

0.1 a) Autoencoder-Based Anomaly Detection

The feature vector is first normalized using a pre-fitted MinMaxScaler and passed through a trained autoencoder model implemented in Keras. The autoencoder attempts to reconstruct the input vector, and the reconstruction error (typically measured as Mean Squared Error, MSE) is computed. A low error indicates normal traffic, while a high error suggests abnormal behaviour. Autoencoder reconstruction error

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

0.2 b) DNN-Based Attack Classification:

In parallel, the same feature vector is passed through a supervised deep neural network (DNN) classifier trained to differentiate between multiple network attack types (e.g., DoS, DDoS, Heart bleed, SQL Injection). The DNN outputs class probabilities using a softmax activation layer, and the label with the highest probability is selected.
Softmax classifier:

Classification Loss (Cross-Entropy):

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

$$\text{Loss} = -\sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i$$

0.3 c) Hybrid Decision Logic:

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

To enhance robustness, this system fuses the anomaly score from the autoencoder and the class probabilities from the DNN. If the anomaly score exceeds a calibrated threshold, the flow is flagged as suspicious regardless of the DNN prediction. Otherwise, the DNNs classi- fication is taken as the final label. This fusion allows the system to detect both known and novel (zero-day) threats.

A combined Autoencoder and Deep Neural Network (DNN) model integrates unsuper- vised and supervised learning in a single architecture. The autoencoder compresses high- dimensional input data into a low-dimensional latent space using an encoder and reconstructs

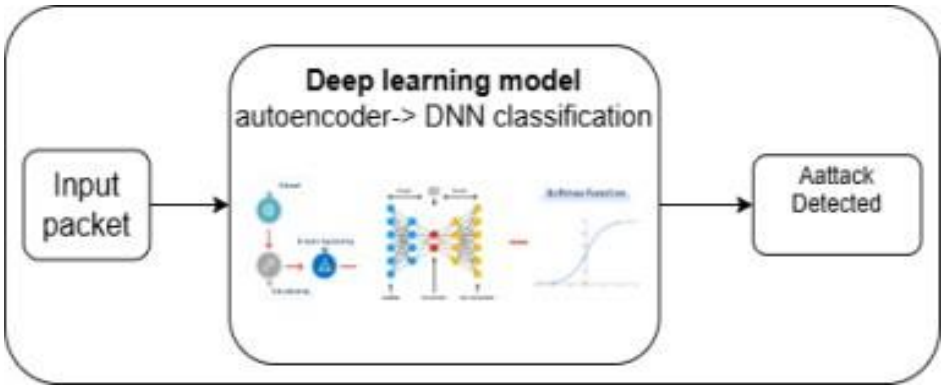

Figure 4: hybrid model it using a decoder, minimizing the reconstruction loss:

Total Loss: - 1.0.MSE+0.5.Cross Entropy

This hybrid model enables better generalization and effective feature extraction and is ideal for tasks like anomaly detection and multi-class classification.

D. Real-Time Visualization

In this system, real-time visualization is achieved through a dynamic dashboard that streams detection data from the backend to the front-end via Web-socket connections. The backend continuously sends updates, including classification results, anomaly scores, and attack type predictions, which are rendered on a Chart.js-powered threat level chart. This chart dy- namically visualizes the number of attacks, anomaly trends, and attack frequency, providing security analysts with up-to-date insights into network activity.

In parallel, a real-time packet table displays key packet information such as timestamps, IP addresses, attack labels, and actions taken, ensuring analysts can quickly respond to emerging threats. Additionally, the backend uses Matplotlib to generate a live plot that

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

displays attack trends and anomaly distributions over time. This plot is periodically updated and served as a PNG image to the frontend. The system also features alerting mechanisms, such as flashing banners and modal popups, to highlight critical detections. The real-time visualization is optimized for low latency and high performance, ensuring quick response times even with multiple concurrent users.

E. WebSocket Broadcasting

WebSocket broadcasting is a technique used to send real-time updates to multiple clients connected to a server over WebSocket connections. In this system, this method is employed to efficiently transmit detection results, attack predictions, and real-time visualizations to all active frontend clients. When a new packet is captured, analyzed, or classified, the backend triggers an update, which is broadcasted to all connected WebSocket clients. This allows each client to receive the most current network security information without the need for constant polling.

WebSocket broadcasting is essential for real-time systems like ShieldAI because it ensures that the frontend can immediately reflect changes in the detection pipeline, including visual-izations like threat level charts and updated packet data. The broadcasting is implemented in the backend using WebSocket server libraries such as websockets or FastAPI's WebSocket functionality, which enables asynchronous message pushing to multiple clients. This allows for efficient, low-latency data transmission, ensuring that security analysts and system ad- ministrators always have access to the latest insights for immediate decision-making

F. Notification and Alert System

Notifications in the context of this system refer to real-time alerts or messages sent to users to inform them about significant events, such as potential security threats or attack predictions. These notifications are essential for ensuring that users can quickly respond to detected intrusions or unusual network behaviour. In this system, notifications are typically triggered by the detection of anomalies or specific attack patterns identified by the deep neural network (DNN) and autoencoder model.

Once an attack is predicted or an anomaly is detected, a notification is generated and sent to users through various channels, such as the frontend interface or email. In the frontend, notifications may appear as visual pop-ups, modals, or alerts, providing users with immediate feedback about the detection event. On the backend, notifications may be implemented using WebSocket messages, which are broadcasted to all connected clients in real-time, ensuring that all users receive updates without delay. Additionally, notifications can be configured to provide details such as the type of attack, severity level, and recommended actions, helping users make informed decisions to mitigate potential threats.

Experimental Design

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Experimental Setup

The real time Intrusion Detection System was developed and tested using Python, incorporating FastAPI for backend services, React for the frontend interface, and advanced machine learning models for real-time intrusion detection. Two primary models were used: a Keras-based autoencoder and DNN for anomaly detection and an DNN classifier for multi-class attack prediction. Matplotlib was employed for real-time visualization, while Scapy enabled packet-level sniffing.

The dataset consisted of real-world network traffic, containing both benign and malicious packets across multiple attack categories (e.g., DDoS, Port Scan, SQL Injection). Prepro- cessing involved packet feature extraction (e.g., IP length, TTL, protocol flags), handling missing values, and normalization using MinMaxScaler to ensure all features were scaled between 0 and 1.

A 70/30 train-test split was applied. The autoencoder was trained solely on normal traffic to learn the typical behavior of the network, while XGBoost was trained on the full labeled dataset. Real-time evaluation was enabled via a WebSocket stream that broadcasted detection results and visual alerts to the frontend. The system also included an OTP-secured login mechanism and optional IP blocking for attack mitigation.

0.4 Evaluation Parameters

The primary evaluation parameter used in this study is the Reconstruction Error of the autoencoder, which measures the mean squared difference between the input packet features and their reconstructed outputs. A higher reconstruction error indicates abnormal or unseen behavior in the network traffic. Based on experimental analysis, a threshold of 0.015 was set, where samples exceeding this value are classified as anomalies, while those below it are considered normal.

Results

The integrated Hybrid-classification model in this system achieved high performance, with training accuracy reaching 99.10% and validation accuracy peaking at 98.87% by Epoch

20. The test set accuracy was 98.81%, indicating strong generalization. Classification loss steadily decreased, while reconstruction metrics remained stable, showing the model effectively learned to classify and detect anomalies in network traffic.

Table 2: Metric Evaluation parameters

Model	Metric	Value

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 12s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Hybrid model	Final Classification accuracy	99.10%
	Test Classification accuracy	98.81%
	Final Classification Loss	0.04

Conclusion

The autoencoder-based intrusion detection model has shown strong and consistent performance throughout training and validation. It combines both reconstruction loss—capturing anomalies through reconstruction error—and a classification head to precisely categorize network traffic. Over 20 training epochs, the model maintained a high classification accuracy, reaching 99.10% on the training set and 98.81% on the test set. These results confirm the model's ability to generalize well and effectively detect intrusions in real time. The integration of dual objectives (reconstruction and classification) has proven to be a robust approach, making this model suitable for deployment in live cybersecurity environments where both accuracy and reliability are critical.

References

[1] Cybersecurity Ventures, "Global Ransomware Damage Costs Predicted To Reach 25*BillionBy*2023*,* "*Annual Cybersecurity Report, vol.*5*, pp.*12 − 24*,* 2023.

[2] M. Akbanov, V. G. Vassilakis, and M. D. Logothetis, "Ransomware detection and mitigation using software-defined networking: A survey," *Computer Networks*, vol. 145, pp. 196–209, 2018.

[3] J. A. Gomez-Hernandez, L. Alvarez-Gonzalez, and P. Garcia-Teodoro, "R-Locker: Thwarting ransomware action through a honeyfile-based approach," *Computers & Security*, vol. 73, pp. 389–398, 2018.

[4] A. Kharraz, W. Robertson, D. Balzarotti, L. Bilge, and E. Kirda, "Cutting the Gordian Knot: A Look Under the Hood of Ransomware Attacks," *DIMVA 2015: Detection of Intrusions and Malware, and Vulnerability Assessment*, vol. 9148, pp. 3–24, 2015.

[5] A. Zimba, L. Simukonda, and M. Chishimba, "Demystifying ransomware attacks: Reverse engineering and dynamic malware analysis of WannaCry for network and information secu- rity," *Zambia ICT Journal*, vol. 1, no. 1, pp. 35–40, 2017.

[6] M. Robers, "Ransomware Legal Issues: Reporting Requirements and Liability Implications,"

Cybersecurity Law & Strategy, vol. 37, no. 2, pp. 1–7, 2021.

Received: August 03, 2025