iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.79715

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

NETWATCH: A Hybrid AI Approach for Network Traffic Behavior Analysis System

Yogeshwaran K[1], Nethen Kumaran A[2], Sanjaigohul A[3], Nithin Krishnan B[4], J. Rosalind Geetha[5]
[1, 2, 3, 4]*Dept. of Artificial Intelligence and Data Science, M.I.E.T. Engineering College, Trichy, India*
[5]*Assistant Professor, Dept. of Artificial Intelligence and Data Science, M.I.E.T. Engineering College, Trichy, India*

Abstract: ***NETWATCH is a real-time Network Intrusion Detection System (NIDS) designed to detect unknown and zero-day cyber-attacks. The system captures live network traffic and converts raw packet data into useful features for analysis. An Isolation Forest model detects unusual activities by learning normal network behavior, while a Long Short-Term Memory (LSTM) Autoencoder analyzes traffic patterns over time and identifies deviations based on reconstruction error. A hybrid decision method combines results of both models to improve accuracy and reduce false alarms. When an anomaly is detected, the system further classifies it into attack types such as DoS/DDoS, port scanning, brute force, and web attacks. All results are stored in a MySQL database and displayed on a real- time dashboard for easy monitoring. This hybrid approach effectively detects new and unseen cyber threats.***

I. INTRODUCTION

With the rapid growth of the internet and digital technologies, network security has become a critical concern for organizations and individuals. Modern networks face a wide range of cyber threats such as Denial of Service (DoS), port scanning, brute force attacks, and advanced persistent threats. Traditional security systems, especially signature-based intrusion detection systems (IDS), are limited in their ability to detect new and unknown (zero-day) attacks, as they rely on predefined attack patterns.

To address these challenges, intelligent and adaptive security solutions are required. NETWATCH proposes a real-time NIDS that uses a combination of Machine Learning (ML) and Deep Learning (DL) techniques to improve detection performance. The system focuses on analysing network traffic behaviour rather than relying only on known attack signatures.

NETWATCH captures live network packets and extracts relevant features for analysis. It uses the Isolation Forest algorithm to identify anomalies in network traffic and an LSTM Autoencoder to analyse temporal patterns. A hybrid decision mechanism combines the outputs of both models to provide more accurate and reliable results. By integrating real-time processing, hybrid AI models, and visualization tools, NETWATCH provides an effective and scalable solution for detecting both known and unknown cyber threats.

II. LITERATURE SURVEY

A. Machine Learning in Network Intrusion Detection

Marco Cantone (IEEE Access, 2024) analyzed the effectiveness of ML techniques in detecting cyber attacks. The study evaluated Random Forest, SVM, and Decision Trees using CICIDS2017 and CSE-CIC2018 datasets. While models achieve high accuracy on same-distribution data, they exhibit poor generalization across different network environments. The study highlights that ML-based IDS are highly dependent on training data and lack adaptability to evolving threats, motivating the need for hybrid approaches [1].

B. Multi-Class Network Intrusion Detection Using ML

Zahid Ullah (2024) focused on improving classification of different cyber attack types using ML with balanced learning to overcome class imbalance. The work demonstrated improved accuracy in classifying DoS, Port Scanning, and Brute Force attacks. However, the system lacks real-time integration and live monitoring capabilities, limiting practical deployment [2].

C. Survey on IDS Using ML and DL

Abbas Mirzaei (2025) provided a comprehensive survey on modern IDS techniques, including supervised, unsupervised, and hybrid models. Deep learning models such as Autoencoders, CNNs, and LSTMs are capable of identifying complex patterns. Despite advancements, challenges remain in real-time deployment and user-friendly visualization [3].

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

III. SYSTEM ANALYSIS

A. Existing System

Existing network security systems primarily rely on signature-based and rule-based intrusion detection techniques. While effective for known attacks, they are unable to detect zero-day attacks. Most operate on offline datasets rather than real-time traffic, limiting active threat response. Key limitations include:

- Rely on signature-based detection; cannot identify unknown attacks.
- Lack of real-time network traffic analysis.
- Poor generalization across different network environments.
- Absence of visualization dashboards and alert mechanisms.
- No integration with live packet capture mechanisms.
- High false positive and false negative rates.

B. Proposed System

NETWATCH proposes a real-time NIDS using a hybrid ML+DL approach. It captures live traffic, extracts key features (packet count, total bytes, packet rate, flow duration), and analyses them using Isolation Forest and LSTM Autoencoder. A hybrid decision mechanism combines both model outputs to improve accuracy and reduce false positives. A Random Forest classifier further identifies attack types. Key advantages include:

- Real-time network traffic monitoring and analysis.
- Detection of both known and unknown (zero-day) attacks.
- Hybrid ML+DL approach for improved accuracy.
- SOC-style dashboard for real-time visualization.
- Attack classification using Random Forest.
- MySQL database logging for incident tracking.

IV. SYSTEM REQUIREMENTS

A. Software Requirements

Component	Specification
Operating System	Windows 7 or Above
Frontend	HTML, CSS, JavaScript, Chart.js
Backend	Python, Flask, Pyshark, Scikit-Learn
Database	MySQL
IDE	Visual Studio Code

Table I: Software Requirements

B. Hardware Requirements

Component	Specification
Processor	Intel i3 or Above
Hard Disk	Minimum 256 GB
Memory (RAM)	2 GB or Above
Network	Network Connectivity Required

Table II: Hardware Requirements

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

V. SYSTEM ARCHITECTURE

The NETWATCH architecture is designed for real-time network monitoring, anomaly detection, and attack classification. It consists of multiple sequential modules:

A. Packet Capture Module

Live network traffic is captured using packet sniffing tools (Scapy/Pyshark).Captured packets contain source/destination IP, protocol, port numbers, and packet size. This raw data is passed to the feature extraction module.

B. Feature Extraction Module

Meaningful features are generated from raw packet data, including packet count, total bytes, packet rate, and flow duration. These features represent the statistical behavior of network flows and serve as input to the detection models.

C. Machine Learning Model – Isolation Forest

The Isolation Forest algorithm detects anomalies by identifying deviations from normal traffic patterns. It isolates observations by randomly selecting features and split values, making it highly effective for anomaly detection in high-dimensional data without requiring labeled training examples.

D. Deep Learning Model – LSTM Autoencoder

The LSTM Autoencoder analyses sequential patterns in network traffic data and detects unusual behaviour based on reconstruction error. High reconstruction error indicates that the current traffic pattern deviates significantly from learned normal behavior, triggering an anomaly flag.

E. Hybrid Decision Engine

The hybrid decision engine combines outputs from both the Isolation Forest and LSTM Autoencoder. When both models indicate abnormal behaviour, traffic is classified as an anomaly. This dual-model approach reduces false positives and improves overall detection accuracy.

F. Attack Classification Module

A Random Forest classifier identifies the specific type of attack from anomalous traffic, categorizing it as DoS/DDoS, port scanning, brute force, or web-based attack. Trained on the CICIDS2017 dataset, it achieves high multi-class classification accuracy.

G. Database and Dashboard Module

All detected anomalies, attack types, and traffic details are stored in a MySQL database for logging and future analysis. A SOC-style real-time dashboard provides visualization of network activity, alerts, and attack statistics using Chart.js, enabling administrators to monitor and respond to threats effectively.

VI. IMPLEMENTATION

A. Dataset – CICIDS2017

The Canadian Institute for Cybersecurity Intrusion Detection System 2017 (CICIDS2017) dataset is used for model training and testing. It contains benign and up-to- date common attacks, including DoS, DDoS, port scanning, brute force, XSS, SQL injection, and infiltration attacks, capturing realistic network traffic patterns.

B. Data Preprocessing

Raw packet data undergoes preprocessing steps including missing value handling, feature normalization (MinMaxScaler), and class balancing techniques. Irrelevant features are removed using feature selection to reduce dimensionality and improve model training efficiency.

C. Model Training

The Isolation Forest model is trained on normal traffic samples to learn baseline behavior. The LSTM Autoencoder is trained to reconstruct normal traffic sequences.

The Random Forest classifier is trained on labeled attack data from CICIDS2017 for multi-class attack identification.

D. Backend Implementation

The backend is implemented using Python and Flask. Pyshark is used for live packet capture. Scikit-Learn provides the Isolation Forest and Random Forest implementations. TensorFlow/Keras is used for the LSTM Autoencoder. The Flask API exposes endpoints for the frontend dashboard.

VII. RESULT AND ANALYSIS

A. Performance Metrics

The system is evaluated using standard metrics: Accuracy, Precision, Recall, F1-Score, and False Positive Rate (FPR). The hybrid decision approach consistently outperforms individual models on all metrics.

Model	Accuracy	Precision	Recall	F1-Scor e
Isolation Forest	91.4%	89.2%	90.1%	89.6%
LSTM Autoencoder	93.7%	92.5%	93.0%	92.7%
Hybrid (Proposed)	96.8%	95.9%	96.4%	96.1%

Table III: Performance Comparison of Detection Models

VIII. ADVANTAGES AND LIMITATIONS

A. Advantages

- Real-time network traffic monitoring and anomaly detection.
- Hybrid AI combining Isolation Forest and LSTM Autoencoder.
- Effective detection of zero-day and unknown attacks.
- SOC-style dashboard with real-time visualization and alerts.

B. Limitations

- Requires significant computational resources for real- time LSTM inference.
- Performance may degrade on highly encrypted traffic.
- Model retraining needed for new attack patterns.
- Initial setup requires labeled dataset for classifier training.

IX. FUTURE ENHANCEMENTS

Future work will focus on: (1) integrating federated learning to enable privacy-preserving distributed detection across multiple network nodes; (2) extending the system to support encrypted traffic analysis using traffic metadata and behavioral fingerprinting; (3) incorporating threat intelligence feeds for enriched context; (4) automating model retraining pipelines to adapt to emerging attack patterns; and (5) developing mobile alerting capabilities for administrators.

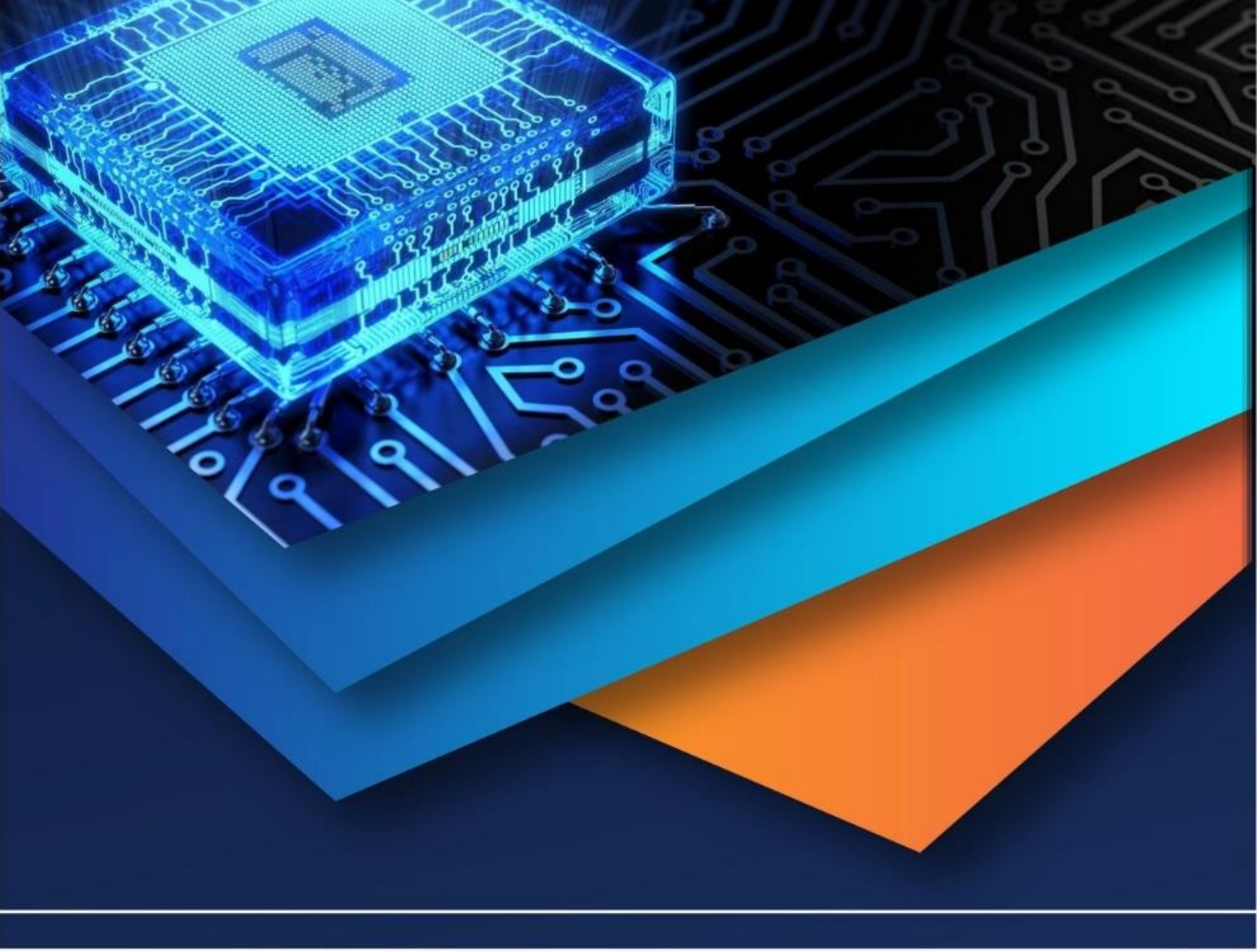
X. CONCLUSION

This paper presented NETWATCH, a hybrid AI- based real-time Network Intrusion Detection System that combines Isolation Forest and LSTM Autoencoder for anomaly detection. The system addresses the fundamental limitation of traditional signature-based IDS by enabling detection of zero-day and unknown attacks. Experimental evaluation on the CICIDS2017 dataset demonstrates that the proposed hybrid approach achieves 96.8% accuracy, outperforming individual models. The integration of a real- time SOC-style dashboard enhances practical usability for network administrators

REFERENCES

[1] M. Cantone, "Machine Learning in Network Intrusion Detection: A Cross-Dataset Generalization Study," *IEEE Access*, vol. 12, pp. 45231–45248, 2024.
[2] Z. Ullah, "Balanced Multi-Class Network Intrusion Detection Using Machine Learning," *IEEE Transactions on Network and Service Management*, 2024.
[3] A. Mirzaei, "A Comprehensive Survey on Intrusion Detection Systems with Advances in Machine Learning, Deep Learning and Emerging

Cybersecurity Challenges," *ACM Computing Surveys*, vol. 57, no. 3, 2025.
[4] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation Forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Pisa, Italy, 2008, pp. 413–422.
[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.
[7] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY