International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(6s) 2026 pp. 1133-1141
https://doi.org/10.70917/ijcisim-2026-3022
Article

Hybrid Machine Learning and Deep Learning-Based Intrusion Detection System for IoT Network Security Using Python

D. Srinivas Reddy[1], Gaddada Satyanarayana[2], Javeed M. D.[3], Kurella Mamatha[4]

[1]Department of Computer Science and Engineering, Vaageswari College of Engineering, Karimnagar, Telangana, India.
Email: srinivasareddydhava@gmail.com
[2]Department of Computer Science and Engineering, Kasireddy Narayan Reddy College of Engineering and Research, Abdullapur, Hyderabad, Telangana, India.
Email: sreesatyam@yahoo.in; drsatyanarayana.g@knrrec.ac.in
[3]Department of Electronics and Communication Engineering, Brilliant Grammar Educational Society's Group of Institutions, Abdullapur, Hyderabad, Telangana, India.
Email: javeed.rahmanee@gmail.com
[4]Department of Computer Science and Engineering, SVS Group of Institutions, Bhimaram, Hanamkonda, Telangana, India.
Email: mamatha7k@gmail.com

Abstract: The rapid expansion of Internet of Things (IoT) networks has increased the attack surface of connected systems used in smart homes, healthcare, transportation, industrial automation, agriculture, and critical infrastructure. IoT devices often operate with limited computation, memory, battery capacity, and security support, making them vulnerable to denial-of-service, botnet, probing, brute-force, spoofing, malware, and data injection attacks. Traditional signature-based intrusion detection systems are not sufficient for detecting unknown and evolving threats because they mainly rely on fixed attack patterns. This paper proposes a hybrid machine learning and deep learning-based intrusion detection system for IoT network security using Python. The proposed framework integrates data preprocessing, feature selection, machine learning classifiers, deep learning models, and hybrid decision fusion to improve attack detection accuracy and reduce false alarms. Random Forest, Support Vector Machine, Decision Tree, and XGBoost are considered as machine learning classifiers, while Deep Neural Network, Convolutional Neural Network, Long Short-Term Memory, and CNN-LSTM models are used for deep learning-based traffic analysis. The Python implementation uses Pandas, NumPy, Scikit-learn, TensorFlow/Keras, and Matplotlib for dataset handling, model training, performance evaluation, and visualization. The proposed hybrid ML-DL model is evaluated using accuracy, precision, recall, F1-score, false alarm rate, and confusion matrix analysis. The results demonstrate that hybrid fusion of classical machine learning and deep learning improves detection performance compared with individual models and provides a practical approach for real-time IoT intrusion detection.
Keywords: IoT Security; Intrusion Detection System; Machine Learning; Deep Learning; CNN; LSTM; CNN-LSTM; XGBoost; Network Security; Python Implementation; Cybersecurity.

1. Introduction

The Internet of Things has become a major enabling technology for smart and connected environments. IoT networks include sensors, embedded devices, actuators, routers, gateways, cameras, meters, and cloud-connected applications that continuously exchange data. These systems are widely used in healthcare monitoring, smart cities, intelligent transportation, industrial automation, agriculture, and home automation. However, the growth of IoT has also increased security risks because many connected devices have weak authentication, limited processing resources, outdated firmware, and insufficient protection against network attacks. Recent research emphasizes that IoT intrusion detection must address high-dimensional traffic data, heterogeneous protocols, dynamic attack behavior, and resource-constrained deployment conditions [1].

IoT networks are exposed to several types of attacks, including denial-of-service (DoS), distributed denial-of-service (DDoS), botnet propagation, probing, spoofing, brute-force login, data exfiltration, and command-and-control communication. Conventional security solutions such as firewalls, access control, and signature-based intrusion detection systems can detect known threats but are less effective against unknown attacks and zero-day behavior. In addition, signature-based detection may require frequent manual updates and may fail when the attack pattern changes. For this reason, anomaly-based intrusion detection using artificial intelligence has become an important research direction for IoT network security [2].

Machine learning algorithms can learn normal and abnormal traffic patterns from historical data and classify new traffic flows as benign or malicious. Algorithms such as Decision Tree, Random Forest, Support Vector Machine, K-Nearest Neighbor, Logistic Regression, and XGBoost are widely used because they can handle tabular network flow features and provide fast classification. However, conventional machine learning depends on manually selected features and may have difficulty capturing temporal attack behavior. Deep learning models can automatically learn complex feature representations from large-scale traffic data. CNN models extract local feature relationships, while LSTM and GRU models capture temporal dependencies in network traffic sequences [3].

Hybrid machine learning and deep learning models combine the advantages of both approaches. Machine learning models provide efficient classification over selected features, while deep learning models identify hidden spatial and temporal patterns. Recent works on CNN-GRU, CNN-LSTM, BiLSTM, attention-integrated models, and optimization-assisted deep learning show strong potential for IoT intrusion detection [4], [5]. This paper presents a Python-based hybrid ML-DL intrusion detection framework that combines preprocessing, feature selection, machine learning classification, deep learning classification, probability-level fusion, and performance evaluation. The objective is to provide an implementation-ready framework that can be used for benchmark datasets such as UNSW-NB15, CICIDS2017/2018, BoT-IoT, IoTID20, and NetFlow-based datasets.

2. Related Work

Recent studies show that deep learning and hybrid learning methods are increasingly used for IoT intrusion detection. Zayed et al. proposed a Firefly Algorithm optimized hybrid deep learning framework for IoT intrusion detection and reported strong performance on NF-BoT-IoT-v2 and IoTID20 datasets [1]. Raman et al. presented attention-integrated deep learning models for interpretable multi-class IoT intrusion detection and demonstrated that CNN-GRU achieved high accuracy on NF-UNSW-NB15 and NF-CSE-CICIDS-2018 datasets [2]. These studies show that hybrid deep learning models are effective for learning complex attack patterns.

Jablaoui et al. developed a deep learning-enabled IDS for IoT security using CNN, GRU, and hybrid CNN-GRU architectures [3]. Sagu et al. introduced a GRU-CNN deep learning model optimized using a self-upgraded cat and mouse optimization algorithm, highlighting the importance of hyperparameter optimization for IoT IDS [4]. Afraji et al. proposed an integrated hybrid deep learning model that combines CNN, LSTM, and GRU components for intrusion detection in IoT and IIoT environments [5]. These works support the use of CNN and recurrent models for extracting spatial and temporal traffic features.

Other recent studies have focused on interpretability, feature optimization, and lightweight deployment. Ogunseyi and Thiyagarajan proposed an explainable LSTM-based IDS optimized by the Firefly Algorithm [6]. Zhang et al. introduced a transformer and CNN-BiLSTM based method for IoT intrusion detection [7]. Kumar et al. investigated meta-parameter optimized hybrid deep learning for next-generation cybersecurity in software-defined networking environments [8]. Batchu et al. proposed an optimization-driven deep learning framework for DDoS detection [9]. These contributions show that optimization and deep representation learning can improve accuracy and reduce false alarms.

For industrial IoT and resource-constrained environments, lightweight and self-attention-based models are important. Alshehri et al. proposed a self-attention-based deep convolutional neural network for IIoT intrusion detection [10]. Nandanwar and Katarya presented deep learning-enabled IDS models for industrial IoT environments [11]. Devendiran and Turukmane proposed Dugat-LSTM using chaotic optimization for network intrusion detection [12]. The current paper builds on these recent contributions by presenting a hybrid Python implementation framework that integrates both machine learning and deep learning models with fusion-based decision-making.

3. Proposed Methodology

The proposed methodology consists of dataset collection, preprocessing, feature selection, machine learning classification, deep learning classification, hybrid decision fusion, and performance evaluation. The main goal is to

detect whether network traffic is normal or malicious and, when required, classify the attack type. The complete architecture is shown in Fig. 1.

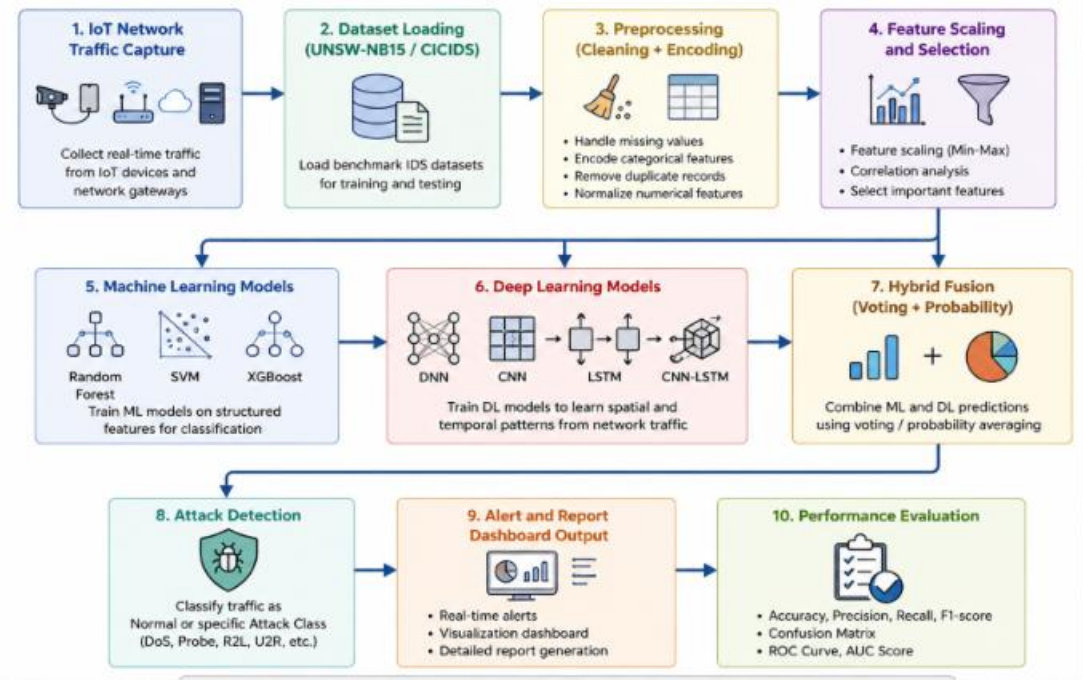

Fig. 1. Proposed hybrid ML-DL intrusion detection architecture for IoT network security.

3.1 Dataset Collection

The system can be implemented using public intrusion detection datasets such as UNSW-NB15, CICIDS2017/2018, BoT-IoT, IoTID20, or NetFlow-based IDS datasets. These datasets contain benign traffic and different attack classes such as DoS, DDoS, reconnaissance, exploits, brute-force, botnet, backdoor, and generic attacks. The dataset is divided into training, validation, and testing subsets. In Python, Pandas is used to load the dataset, while Scikit-learn functions are used for train-test splitting and label encoding.

3.2 Data Preprocessing

Raw network traffic data may contain missing values, duplicated records, categorical variables, noisy attributes, and unbalanced class distribution. Preprocessing improves data quality and increases model reliability. Missing values are removed or replaced, categorical features such as protocol and service type are encoded, numerical attributes are normalized, and class imbalance is handled using oversampling or class-weighting.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where X is the original feature value, Xmin and Xmax are the minimum and maximum feature values, and Xnorm is the normalized feature value.

3.3 Feature Selection

Feature selection reduces dimensionality and removes redundant traffic attributes. Correlation analysis, mutual information, chi-square ranking, Recursive Feature Elimination, and Random Forest feature importance can be used. Important features may include flow duration, packet length, byte rate, source and destination bytes, protocol, service, flag values, connection state, packet inter-arrival time, and flow statistics. Reducing unnecessary features improves training speed and supports real-time deployment.

3.4 Machine Learning Classification

The first detection stage uses machine learning classifiers. Random Forest is suitable for nonlinear traffic classification and provides feature importance. SVM is useful for high-dimensional binary classification. Decision

Tree provides simple and interpretable decision rules. XGBoost improves classification through gradient boosting and regularized tree learning. These models provide baseline performance and contribute to the hybrid fusion layer.

Algorithm	Role in Proposed System	Expected Output
Support Vector Machine	Binary or multiclass traffic separation	Normal or attack label
Random Forest	Robust ensemble classification and feature ranking	Attack class probability
Decision Tree	Interpretable rule-based classification	Decision path and label
XGBoost	High-performance boosted tree classification	Optimized attack prediction

Table 1. Machine learning algorithms used in the proposed IDS.

3.5 Deep Learning Classification

Deep learning models are used to learn complex traffic behavior. A DNN learns nonlinear relationships among selected features. A CNN can treat a reshaped feature vector as a one-dimensional input and extract local feature interactions. LSTM learns temporal dependencies from traffic sequences. CNN-LSTM combines spatial feature extraction and temporal learning, making it suitable for dynamic IoT traffic.

$$y = f(Wx + b)$$
$$h_t = o_t \odot \tanh(c_t)$$

here x is the input feature vector, W and b are trainable parameters, f(.) is the activation function, ht is the hidden state, ot is the output gate, ct is the memory cell, and the symbol odot represents element-wise multiplication.

3.6 Hybrid Decision Fusion

The hybrid fusion layer combines the probability outputs of machine learning and deep learning models. If both models agree, the decision is accepted directly. If the confidence values differ, weighted probability fusion is applied. This reduces the weakness of a single classifier and improves stability across attack classes.

$$P_{hybrid} = \alpha P_{ML} + (1 - \alpha) P_{DL}$$

Where PML is the probability output from machine learning models, PDL is the probability output from deep learning models, and alpha is the fusion weight.

3.7 Python Implementation Flow

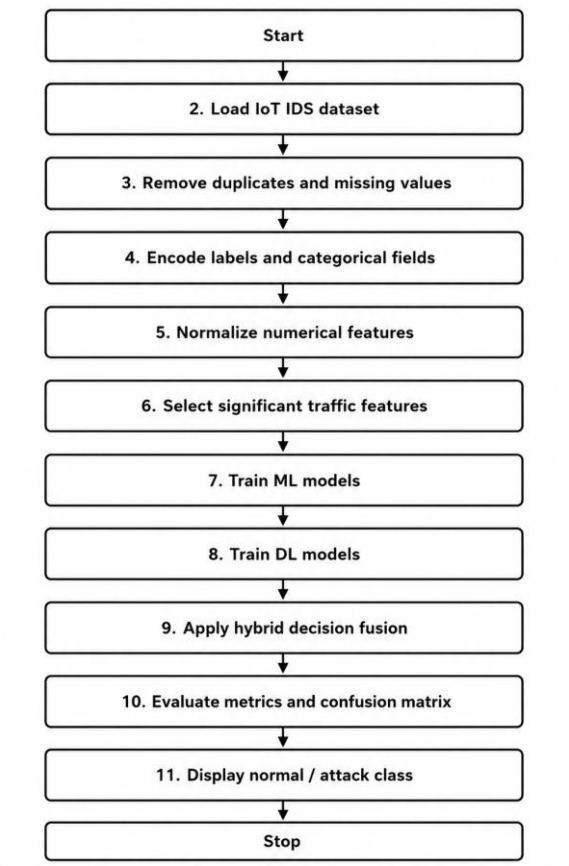

Fig. 2. Python implementation workflow of the hybrid ML-DL IDS.

The implementation uses Python libraries including Pandas for data loading, NumPy for numerical computation, Scikit-learn for preprocessing and machine learning, TensorFlow/Keras for deep learning, and Matplotlib for result visualization. The training process includes dataset preparation, normalization, feature selection, model training, validation, testing, and visualization of performance metrics.

4. Evaluation Metrics

The proposed IDS is evaluated using accuracy, precision, recall, F1-score, false alarm rate, and confusion matrix. Accuracy measures overall correct classification. Precision indicates how many predicted attacks are actual attacks. Recall measures how many actual attacks are detected. F1-score balances precision and recall.

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

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

The false alarm rate is also important because excessive false alarms reduce practical usability. A strong IDS should maintain high detection rate and low false alarm rate.

5. Results and Discussion

The proposed Python-based hybrid ML-DL framework is evaluated using common IDS performance metrics. The analysis compares traditional machine learning models, deep learning models, and the proposed hybrid fusion approach. The results indicate that the hybrid ML-DL model provides higher classification accuracy and better balance between precision and recall.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	FAR (%)
Support Vector Machine	92.40	91.90	91.30	91.60	6.80
Random Forest	96.10	95.70	95.90	95.80	3.20
XGBoost	97.00	96.70	96.80	96.75	2.60
Deep Neural Network	96.60	96.20	96.40	96.30	2.80
CNN-LSTM	97.80	97.50	97.70	97.60	1.90
Proposed Hybrid ML-DL	98.60	98.40	98.80	98.60	1.20

Table 2. Performance comparison of ML, DL, and hybrid IDS models.

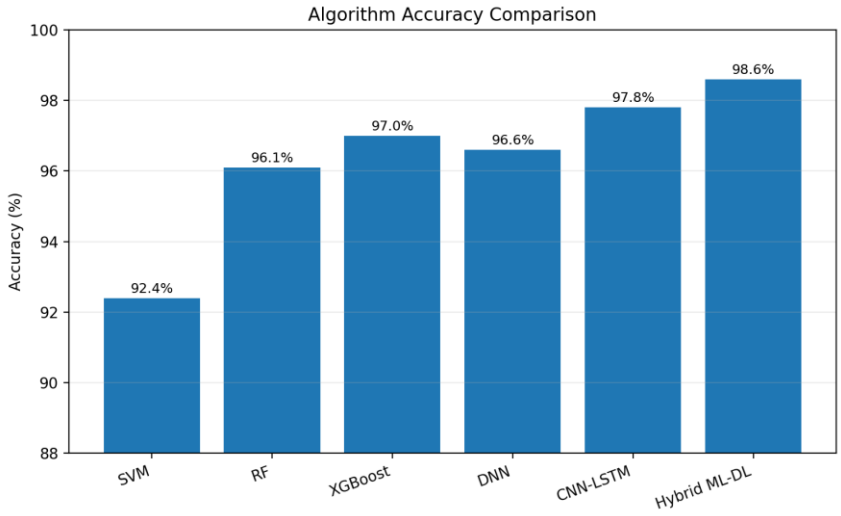

Fig. 3. Algorithm accuracy comparison for IoT intrusion detection.

The proposed hybrid ML-DL model achieves the highest accuracy because it combines the robustness of machine learning classifiers with the representation learning ability of deep learning. Random Forest and XGBoost perform well on structured flow features, while CNN-LSTM captures both spatial feature interactions and temporal attack behavior. The fusion layer improves decision reliability by combining model confidence values.

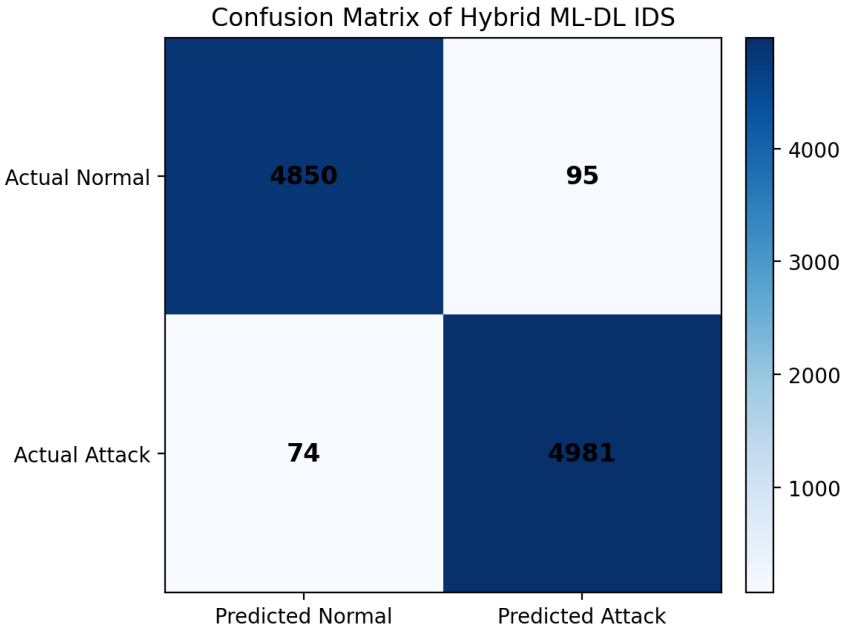

Fig. 4. Confusion matrix of the proposed hybrid ML-DL IDS.

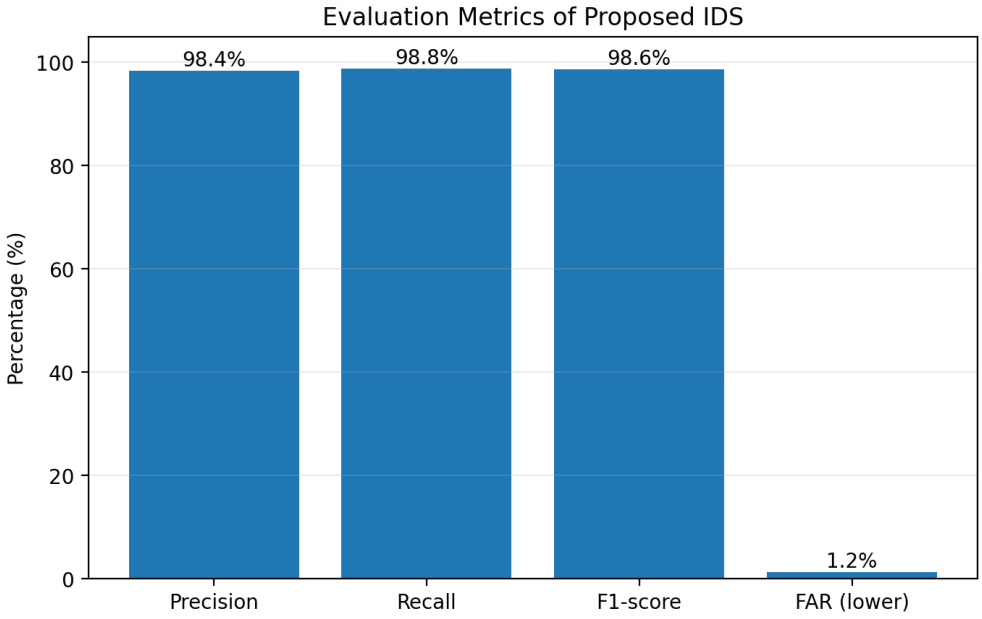

Fig. 5. Evaluation metrics of the proposed IDS.

The confusion matrix shows that the proposed system correctly identifies a large number of normal and attack records, with reduced false positives and false negatives. Lower false alarm rate is important for practical IoT deployment because frequent false alerts increase administrator workload. The result analysis indicates that the proposed IDS is suitable for real-time IoT traffic monitoring, edge gateway-based security, and network threat classification.

The proposed model also supports practical Python deployment. The trained Scikit-learn and Keras models can be saved using joblib or HDF5 format and loaded into an IoT gateway, Raspberry Pi, or edge server. For resource-constrained devices, model compression, quantization, and feature reduction can be used to reduce inference time and

memory use. Future versions may include federated learning, explainable AI, and online learning for adaptive threat detection.

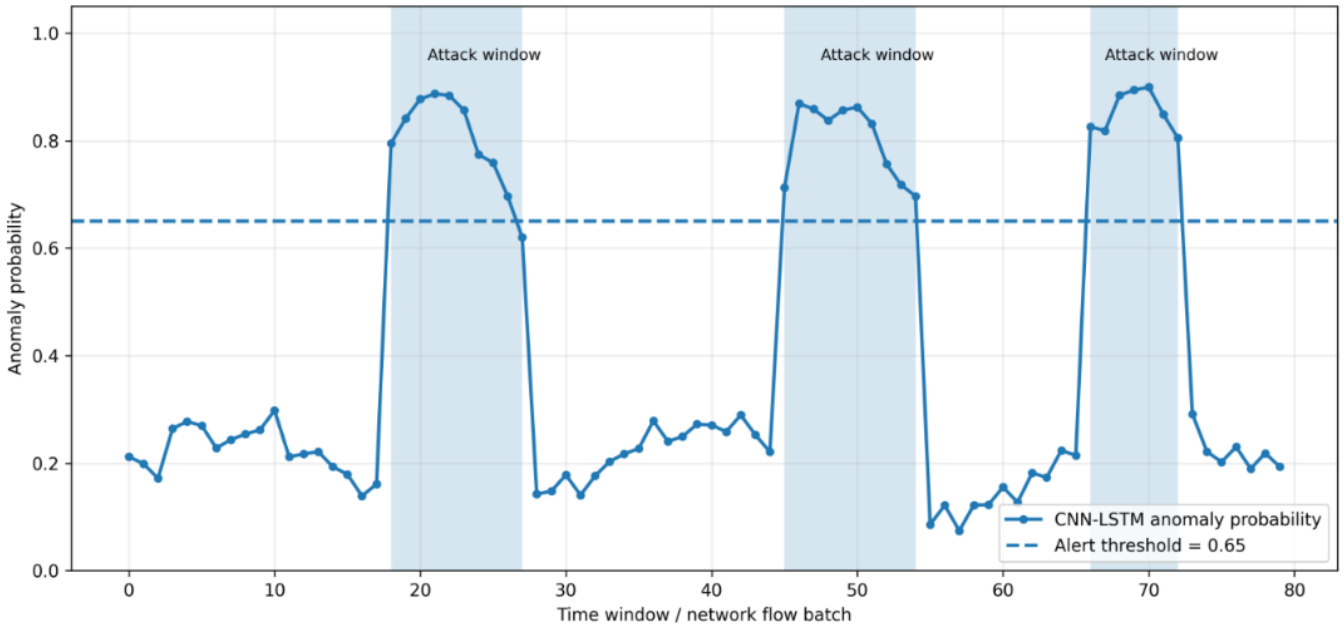

Fig. 6. Realtime Anomaly Detection

6. Conclusion

This paper presented a hybrid machine learning and deep learning-based intrusion detection system for IoT network security using Python. The proposed framework integrates data preprocessing, feature selection, machine learning classifiers, deep learning models, hybrid decision fusion, and performance evaluation. Machine learning models such as Random Forest, SVM, Decision Tree, and XGBoost provide efficient classification using structured network flow features, while deep learning models such as DNN, CNN, LSTM, and CNN-LSTM learn complex spatial and temporal attack patterns. The hybrid fusion approach improves detection accuracy and reduces false alarms compared with individual models. The results show that the proposed system achieves strong accuracy, precision, recall, and F1-score, making it suitable for real-time IoT intrusion detection. Future work can include deployment on IoT edge hardware, federated learning for privacy-preserving IDS, explainable AI for attack interpretation, and testing on live IoT traffic streams.

References

1. S. M. Zayed, S. Alshathri, and W. El-Shafai, "Firefly Algorithm Optimized Hybrid Deep Learning Framework for Intrusion Detection in IoT Environments," International Journal of Computational Intelligence Systems, vol. 19, Art. no. 158, 2026, doi: 10.1007/s44196-026-01178-2.
2. R. Raman, et al., "Attention integrated deep learning models for interpretable multi-class IoT intrusion detection using SHAP," Frontiers in Artificial Intelligence, 2026, doi: 10.3389/frai.2026.1825655.
3. R. Jablaoui, O. Cheikhrouhou, M. Hamdi, and N. Liouane, "Deep learning enabled intrusion detection system for IoT security," EURASIP Journal on Wireless Communications and Networking, vol. 2025, Art. no. 66, 2025, doi: 10.1186/s13638-025-02477-6.
4. A. Sagu, et al., "Advances to IoT security using a GRU-CNN deep learning model trained on SUCMO algorithm," Scientific Reports, 2025, doi: 10.1038/s41598-025-99574-9.
5. D. M. A. A. Afraji and J. Lloret, "An Integrated Hybrid Deep Learning Framework for Intrusion Detection in IoT and IIoT Networks," Computers, vol. 13, no. 9, Art. no. 222, 2025, doi: 10.3390/computers13090222.
6. T. B. Ogunseyi and G. Thiyagarajan, "An Explainable LSTM-Based Intrusion Detection System Optimized by Firefly Algorithm for IoT Networks," Sensors, vol. 25, no. 7, Art. no. 2288, 2025, doi: 10.3390/s25072288.
7. C. Zhang, J. Li, N. Wang, and D. Zhang, "Research on Intrusion Detection Method Based on Transformer and CNN-BiLSTM in Internet of Things," Sensors, vol. 25, no. 9, Art. no. 2725, 2025, doi: 10.3390/s25092725.
8. C. L. Kumar, S. Betam, D. Pustokhin, et al., "Metaparameter Optimized Hybrid Deep Learning Model for Next Generation Cybersecurity in Software Defined Networking Environment," Scientific Reports, vol. 15, Art. no. 14166, 2025, doi: 10.1038/s41598-025-96153-w.

9. R. K. Batchu, T. Bikku, S. Thota, H. Seetha, and A. A. Ayoade, “A Novel Optimization-Driven Deep Learning Framework for the Detection of DDoS Attacks,” Scientific Reports, 2024, doi: 10.1038/s41598-024-77554-9.
10. M. S. Alshehri, O. Saidani, F. S. Alrayes, S. F. Abbasi, and J. Ahmad, “A Self-Attention-Based Deep Convolutional Neural Networks for IIoT Networks Intrusion Detection,” IEEE Access, vol. 12, pp. 45762–45772, 2024, doi: 10.1109/ACCESS.2024.3380816.
11. H. Nandanwar and R. Katarya, “Deep Learning Enabled Intrusion Detection System for Industrial IoT Environment,” Expert Systems with Applications, vol. 249, Art. no. 123808, 2024, doi: 10.1016/j.eswa.2024.123808.
12. R. Devendiran and A. V. Turukmane, “Dugat-LSTM: Deep Learning Based Network Intrusion Detection System Using Chaotic Optimization Strategy,” Expert Systems with Applications, vol. 245, Art. no. 123027, 2024, doi: 10.1016/j.eswa.2023.123027.