Journal of Computer Science

Research Article

An Optimized Multi-Layer Perceptron Framework for Detecting and Classifying IoT Attacks

Sanchit Vashisht and Shalli Rani

Chitkara Institute of Engineering and Technology, Chitkara University, Punjab, India

Article history
Received: 36-06-2025
Revised: 20-10-2025
Accepted: 25-10-2025

Corresponding Author:
Shalli Rani
Chitkara Institute of
Engineering and Technology,
Chitkara University, Punjab,
India
Email: shallir79@gmail.com

Abstract: The Internet of Things (IoT) is revolutionizing industries by connecting billions of smart devices, enabling automation and information exchange. The expansion of IoT ecosystems has simultaneously increased the surface area for cyberattacks. These environments are particularly vulnerable to a wide range of threats, such as Distributed Denial-of-Service (DDoS), poisoning, brute-force SSH intrusions, and various network reconnaissance techniques. The dynamic nature of IoT traffic makes traditional security measures inadequate, thereby necessitating intelligent and adaptive solutions. This study leverages Artificial Intelligence (AI) to combat the growing cybersecurity challenges in IoT. An optimized Multi-Layer Perceptron model is designed to identify and classify cyberattacks with high precision. Using the RT-IoT2022 dataset, which includes realistic network traffic from IoT devices and multiple attack vectors, the model is trained on the 35 most relevant features selected from a total of 85 using permutation importance. The dataset encompasses both benign and adversarial traffic collected via advanced monitoring tools like Wireshark and Zeek. Through rigorous preprocessing, feature engineering, and hyperparameter tuning, the proposed MLP model shows exceptional performance with an accuracy of 99.98. Comparative analysis further shows the superiority of the optimized MLP model over traditional ML algorithms.

Keywords: Cybersecurity, Internet of Things, Artificial Intelligence, Cyberattacks, Intrusion Detection

Introduction

IoT has revolutionized modern digital infrastructure by enabling seamless interaction between physical devices through the internet (Farooqi *et al*., 2023). Initially conceptualized in the early 2000s, IoT has rapidly evolved from basic sensor networks to a sophisticated ecosystem encompassing smart homes, healthcare systems, industrial automation, and smart cities (Venčkauskas *et al*., 2024). By embedding sensors and actuators in everyday objects, IoT enables real-time data collection, processing, and decision-making (Cherfi *et al*., 2025). With the integration of advanced communication protocols like 5G, edge computing, and cloud platforms, IoT now supports billions of interconnected devices globally. Current trends highlight the growing adoption of AI-powered IoT, Digital Twins, and blockchain-enabled secure transactions, emphasizing the shift toward intelligent, autonomous, and self-learning systems (Hisham *et al*., 2023).

The expansion of IoT has also widened the threat surface for cyberattacks, leading to significant security and privacy challenges. IoT systems are inherently vulnerable due to resource-constrained devices, heterogeneous architectures, and weak authentication mechanisms (Gürfidan, 2024). Common cyber threats include DoS, DDoS, sniffing, botnet attacks, data exfiltration, firmware manipulation, and ransomware. According to recent cybersecurity reports, there has been a surge of over 300% in IoT-based attacks in the last five years, with threat actors increasingly exploiting unpatched devices and misconfigured networks (Alhchaimi, 2024). These vulnerabilities not only jeopardize user privacy and data integrity but also threaten critical infrastructure operations, making security a top priority in the IoT landscape (Mehmood *et al*., 2025). To address these challenges, AI, ML, and DL have emerged as powerful tools for proactive threat

SCIENCE Publications

detection and classification in IoT environments. These technologies enable systems to learn from data, identify anomalous behavior, and adapt to evolving attack patterns in real time (Kamran *et al*., 2024). ML algorithms such as Random Forests, Support Vector Machines (SVM), and k-Nearest Neighbors (kNN), as well as DL architectures like Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), and Autoencoders, have been successfully applied to IDS for IoT. By leveraging pattern recognition, feature extraction, and model generalization, these intelligent models enhance the resilience and responsiveness of IoT networks (Airlangga, 2024). In this study, a Multi-Layer Perceptron (MLP) model is used on the RT-IoT 2022 dataset for the identification and classification of cyberattacks present in real-time IoT networks.

Literature Review

The literature on attack detection in IoT networks highlights various techniques, including ML, genetic algorithms, and statistical methods. While traditional methods face scalability and efficiency issues, recent advancements focus on lightweight, edge-based solutions that optimize feature selection and improve detection accuracy, addressing attacks. Malik and Dutta (2023) proposed an ML-based framework for detecting DDoS attacks in IoT networks using the IoT-CIDDS dataset. It emphasizes data enrichment, advanced feature engineering, and performance comparison of classifiers. Results show that RF achieved the best accuracy and efficiency with reduced false positives and optimal feature selection. The study focuses solely on DDoS attacks, uses limited models, and lacks evaluation across multiclass scenarios and real-time environments. Saiyed and Al-Anbagi (2024a) proposed DEEPShield, a deep ensemble learning system combining CNN and LSTM with unit pruning to detect high- and low-volume DDoS attacks in IoT environments. It introduces a novel HL-IoT dataset and demonstrates over 90% accuracy across multiple datasets, optimizing performance for edge deployment with reduced resource usage. The system may face generalization challenges across unseen attack types and lacks evaluation on real-world, heterogeneous IoT network configurations.

Srivastava *et al*. (2023). This study addresses IoT security by proposing a flexible strategy to detect and counteract malicious activities without burdening IoT devices. Using ML, it compares Linear and Non-Linear SVM, showing that Non-Linear SVM significantly improves detection accuracy from 93 to 97.8%, enhancing security in diverse IoT environments. The approach lacks testing on large-scale real-world IoT networks and may struggle with scalability and evolving attack patterns. Alabsi *et al*. (2023) introduce a dual CNN-based framework for detecting IoT network attacks, where the first CNN selects crucial features and the second performs detection. Using the BoT IoT 2020 dataset, the model achieves high accuracy and outperforms traditional DL methods, showcasing its effectiveness in IoT threat detection. The model is limited by reliance on a single dataset and may face challenges when generalized to other IoT network scenarios.

Alani and Damiani (2023) present a lightweight, explainable ensemble-based machine learning system for detecting reconnaissance attacks on IoT devices. Designed for resource-constrained environments, the system accurately identifies scanning behavior early in attack campaigns, achieving 99\% accuracy with very low false positive (0.6\%) and false negative (0.05\%) rates during testing. The system’s performance may vary with different attack types or newer datasets, and scalability across diverse IoT platforms remains untested. Saiyed and Al-Anbagi (2024b) propose the Genetic Algorithm and t-Test for DDoS Attack Detection (GADAD) system for IoT networks. It uses edge-based technologies and a custom HL-IoT dataset to optimize feature selection with GAStats. The system trains various tree-based models, demonstrating improved detection efficiency and reduced computation time for DDoS attacks. The system may face scalability challenges in large-scale IoT networks with diverse attack types beyond DDoS.

Preliminaries

The growing ecosystem of the IoT has introduced new opportunities for innovation across multiple domains. However, its rapid expansion has also introduced various security vulnerabilities, prompting the need for robust cyberthreat detection mechanisms. This section outlines the fundamental concepts related to IoT and the cyberthreat landscape, followed by a detailed discussion of AI models commonly used for threat detection in IoT systems.

IoT: Overview and Cybersecurity Challenges

IoT represents a network of connected devices capable of gathering, transmitting, and processing data without any human intervention. While IoT enables enhanced automation, monitoring, and data-driven decision-making, it also presents significant security and privacy concerns (Li, 2024). The heterogeneous nature of devices, constrained computational resources, lack of standardized security protocols, and the use of outdated firmware contribute to a wide attack surface. IoT systems are increasingly targeted by cyberattacks such as DoS, DDoS, MitM, spoofing, phishing, firmware modification, and botnet infiltration. These attacks aim to compromise data integrity, confidentiality, and availability, which can have catastrophic consequences, especially in sectors like healthcare, smart grids, and industrial control systems (Sharma and Babbar, 2023).

AI Models for Threat Detection

To detect and classify cyberthreats in IoT environments, a wide range of ML and DL algorithms are employed. Below is an overview of several prominent models along with their mathematical formulations (Srinivasan, 2024).

K-Nearest Neighbors

Using the majority class of a data point's k closest neighbors in the feature space, KNN is a non-parametric learning method for classification. The Euclidean distance is commonly used to calculate the similarity using Eq. 1:

$$d(u,v) = \sqrt{\sum_{i=1}^{n}(u_i - v_i)^2} \tag{1}$$

Where $(u = u_1, u_2, \dots, u_n)$ and $(v = v_1, v_2, \dots, v_n)$ are two data points in (n)-dimension space.

Logistic Regression

A linear model called LR is applied to situations involving binary classification. It uses the logistic function, which is provided in Eq. 2, to forecast the likelihood that a given input (x) belongs to a specific class:

$$\mathrm{P(v = 1|u) = \sigma(w^T u + b)} \tag{2}$$

Where $P(v = 1|u), w,\ b,$ and u are the conditional probability that u belongs to 1, the weight vector, bias, and input feature vector.

Naïve Bayes

According to Eq. 3, NB is a probabilistic classifier that relies on the Bayes theorem and assumes feature independence:

$$P(v|U) = \big(P(v) * \Pi P(u_i|v)\big)/P(U) \tag{3}$$

Where $P(v|U)$ is the posterior probability of class v given predictor U. Bayes' theorem, expressed as Eq. 4. It assumes conditional independence between features, meaning that each feature contributes independently to the probability of the class:

$$\text{Posterior}\, Prior\ \times\ Likelihood\ /\ Evidence \tag{4}$$

Support Vector Machine

As shown in Eq. 5, SVM seeks to identify the best hyperplane that divides the classes with the greatest margin:

$$f(x) = w^T x + b \tag{5}$$

Classification is determined by the sign of $f(x)\colon R^n \to R,$ where w and b are the model parameters.

Decision Tree

To maximize information gain, a DT iteratively divides the dataset into subgroups according to feature values. The entropy-based Information Gain (IG) is determined by Eq. 6:

$$IG(S,A) = E(S) - \sum_{v\in \mathrm{Values}(A)} \frac{|S_v|}{|S|} E(S_v) \tag{6}$$

Where $S,\ E(S),$ and A are the dataset, entropy, and attribute used for splitting, respectively, and

$$E(S) = -\sum_{i=1}^{k} p_i \log_2 p_i \tag{7}$$

Where p_i represents the proportion of class i in the dataset S, and k is the total number of classes. Entropy quantifies the impurity or uncertainty in the dataset.

Random Forest

RF is an ensemble of DT that outputs the mode of class predictions from individual trees. The general prediction for classification is given by Eq. 8:

$$\hat{v} = \mathrm{mode}\big(T_1(u), T_2(u), \dots, T_m(u)\big) \tag{8}$$

Where $\mathrm{T_i(u)}$is is the prediction of the i-th decision tree. The most frequently predicted class across DT is the mode, which can be defined as Eq. 9:

$$\mathrm{mode}(a_1, a_2, \dots, a_m) =\ \arg\max_{c} \sum_{i=1}^{m} 1(a_i = c) \tag{9}$$

Where $1(\mathrm{a_i = c})$ is an indicator function that equals 1 if $\mathrm{a_i = c}$, and 0 otherwise.

Multi-Layer Perceptron

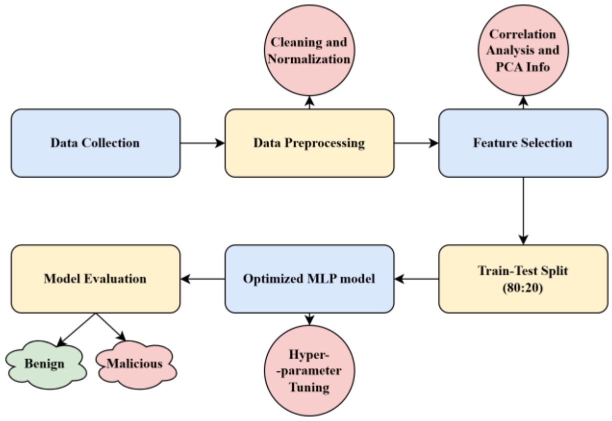
MLP is a feedforward NN with one or more hidden layers. The output of a neuron in layer l is computed as Eq. 10:

$$a^{(l)} = \sigma\big(W^{(l)} a^{(l-1)} + b^{(l)}\big) \tag{10}$$

Where $\mathrm{a^{(l)}}$ is the activation at layer l, $\mathrm{W^{(l)}}$ is the weight matrix, $\mathrm{b^{(l)}}$ is the bias, and σ is the activation function. These models' capacity to manage intricate patterns and abnormalities has led to their widespread adoption for IoT intrusion detection. This work uses the RT-IoT 2022 dataset and an MLP model to identify and categorize different kinds of cyberattacks in real-time IoT networks.

Materials and Methods

This section outlines the methodological framework adopted for detecting and classifying cyberattacks in real-time IoT environments. The proposed pipeline consists of four key stages: Data acquisition, preprocessing, feature selection and hyperparameter optimization, and classification using an MLP model. The overall objective is to accurately identify and categorize malicious activity within a heterogeneous IoT infrastructure. Figure 1 depicts the framework for the proposed model.



Data Collection

The vast, unique RT-IoT2022 dataset from the Kaggle repository simulates normal and adversarial processes in real-time IoT systems. It simulates real-world network environments with a variety of devices and attack scenarios. The dataset includes ThingSpeak-LED, Wipro-Bulb, and MQTT-Temperature sensor network activity. Metasploit, Hping, Slowloris, and Nmap-based reconnaissance patterns (TCP scan, UDP scan, OS detection, FIN scan, and XMAS Tree scan) are simulated cyberattacks. ARP poisoning and DOS SYN attacks are added to represent further threats. By connecting attacker and victim devices to a router, network traffic was recorded. Zeek (previously Bro) with the Flowmeter plugin extracts bidirectional traffic features. Traffic was captured and exported as PCAP files using Wireshark, an open-source network monitoring program. Assailants have 50 machines while victims have 5 departments, 420 workstations, and 30 servers. The dataset encapsulates system logs and network traces from each device, with 80 extracted features. It includes 123,117 records spanning 9 attack types: DOSSYNHping, ARPPoisoning, NMAPUDPScan, NMAPXMASTreeScan, NMAPOSDetection, NMAPTCPScan, DDOSSlowloris, MetasploitBruteForceSSH, NMAPFINScan; and 3 normal traffic patterns: MQTT, ThingSpeak, and WiproBulbDataset.

Due to its comprehensiveness, real-world relevance, and deep coverage of multiple attack scenarios, the RT-IoT 2022 dataset was chosen over other publicly available IoT security datasets. RT-IoT 2022 mimics a genuine IoT architecture with a wide range of normal and harmful behaviors, unlike previous datasets that focused on network traffic or limited attack types. It uses data from ThingSpeak-LED, Wipro-Bulb, and MQTT-based IoT devices and meticulously simulated attacks, including DDoS (Hping, Slowloris), ARP poisoning, SSH brute-force, and different Nmap scanning methods. This dataset shows current IoT network traffic more realistically with 85 features collected using Wireshark, Zeek, and Flowmeter. Its bidirectional traffic and system logs across 420 machines and 50 attacker systems make it ideal for training sophisticated detection algorithms.

Data Preprocessing

The raw dataset was initially processed to handle missing values and remove redundant or irrelevant records. Noise and duplicates were eliminated to improve training consistency. Following data cleaning, normalization, and standardization were applied to ensure all feature values fall within a uniform range, particularly critical for MLP models sensitive to scale variations. The dataset was then divided into training and testing sets using an 80:20 split ratio. This ensured that the model was trained on a representative majority of the data while retaining sufficient unseen samples for performance evaluation.

Feature Selection and Hyperparameter Optimization

The selection of features has been performed using correlation-based analysis and feature importance metrics to decrease dimensionality and enhance model effectiveness. Out of the original 85 features, the top 35 most relevant features were selected based on their contribution to classification accuracy. Table 1 shows the selected features along with their permutation importance. To optimize the MLP model, hyperparameter tuning was conducted using grid search and cross-validation. The objective was to minimize loss and maximize classification accuracy. The best-performing parameters identified through this process are: Activation Function: ReLU, hidden layer sizes: (100), (150), (200), L2 penalties: 0.0001, 0.00005, batch sizes: 32, 64, 128, Learning Rate: Adaptive, and Solver: Adam. These refined hyperparameters were selected after extensive testing to balance learning capacity, convergence speed, and generalization.

Table 1: Features Based on Permutation Importance from DT Model

Rank	Feature Name	Permutation Importance	Rank	Feature Name	Permutation Importance
1	flowACK_flagcount	0.345	19	flowiat_avg	0.020
2	bwdpkts_payloadavg	0.065	20	fwdiat_min	0.018
3	fwdpkts_payloadavg	0.048	21	bwdbulk_bytes	0.016
4	bwdiat_min	0.038	22	bwddata_pktstot	0.015
5	fwdpkts_tot	0.032	23	flowpkts_payloadstd	0.014
6	bwdpkts_payloadmin	0.031	24	bwdheader_sizetot	0.014
7	flowSYN_flagcount	0.030	25	bwdiat_tot	0.013
8	flowiat_min	0.029	26	bwdpkts_tot	0.012
9	bwdpkts_payloadmax	0.027	27	bwdpkts_persec	0.011
10	bwdpkts_persec	0.026	28	flowpkts_payloadavg	0.010
11	bwdpkts_payloadstd	0.026	29	fwdheader_sizemin	0.009
12	flowFIN_flagcount	0.025	30	flowpkts_payloadmin	0.009
13	flowCWR_flagcount	0.024	31	flowpkts_payloadtot	0.008
14	bwdbulk_rate	0.024	32	flowpkts_persec	0.008
15	bwdURG_flagcount	0.023	33	flowduration	0.007
16	bwdpkts_payloadtot	0.023	34	bwdiat_max	0.007
17	fwdpkts_payloadmax	0.021	35	fwdpkts_payloadstd	0.006
18	fwdpkts_payloadtot	0.020			

Attack Detection and Classification Using Optimized MLP

The preprocessed and feature-reduced dataset was used to train the improved MLP model, which was then used to categorize the traffic into harmful and normal categories. An input layer, many hidden layers, and an output layer make up MLP, a deep feedforward neural network. For weight optimization, it uses stochastic gradient descent and backpropagation. Accuracy, precision, recall, F1-score, and ROC-AUC were among the performance measures used to assess the final model. The effectiveness of the suggested strategy was validated by the trained model's great skill in differentiating between benign and malevolent patterns in real-time IoT contexts.

Results and Discussion

This section presents the evaluation metrics used to evaluate the performance of the proposed model for IoT attack detection. All experiments have been implemented in Python 3.10 using the Scikit-learn and TensorFlow libraries. The experiments are conducted on a system equipped with an Intel Core i7-13700 CPU, 32 GB RAM, and an NVIDIA RTX 4060 GPU with 16 GB memory. The following subsections provide detailed insights into each metric, along with their respective equations and the performance outcomes derived from experimental analysis.

Accuracy

The percentage of correctly categorized cases out of all the samples is known as accuracy. It gives a general indication of how accurate the model is. Higher accuracy indicates the model performs well in both positive and negative class predictions. Figure 2 and Table 2 illustrate the performance improvement of the MLP model after optimization. Before optimization, the model achieved an accuracy of 99.7888%, precision of 99.79%, recall of 99.78%, and F1-score of 99.78%. After optimization, there was a noticeable enhancement across all metrics, with the accuracy rising to 99.9838%, precision to 99.98%, recall to 99.98%, and F1-score also reaching 99.98%. These results highlight the significant impact of the optimization process on the overall performance of the MLP model.

Precision

The precision measures the percentage of positive identifications that were truly accurate. When false positive costs are substantial, it is essential. High precision suggests that the model returns more relevant than irrelevant results.

Recall

The ratio of true positives that are correctly identified is measured by recall, which is sometimes referred to as sensitivity or true positive rate. Recall is essential in applications where missing a positive prediction (false negative) has severe consequences.

Table 2: Comparison of MLP Model Performance Before and After Optimization

Metric	Before Optimization (%)	After Optimization (%)
Accuracy	99.788	99.9838
Precision	99.79	99.98
Recall	99.78	99.98
F1-Score	99.78	99.98

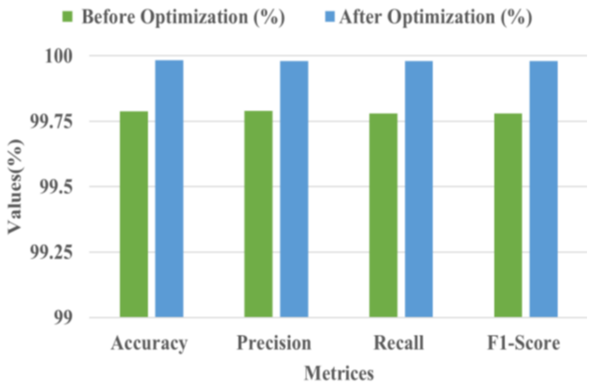


Fig. 2: Before optimization and after optimization analysis of the MLP model

F1-Score

The harmonic mean of recall and precision is known as the F1-Score. It balances their trade-offs, particularly in cases where the distribution of classes is unbalanced. A higher F1-score indicates both high precision and high recall.

Figure 3 and Table 3 present a performance comparison of the optimized MLP with various traditional ML models. The KNN model achieved 98.67% accuracy, 98.70% precision, 98.65% recall, and 98.66% F1-score. LR followed with 97.85% accuracy, 97.90% precision, 97.82% recall, and 97.85% F1-score. NB showed relatively lower performance with 94.32% accuracy, 94.10% precision, 94.50% recall, and 94.30% F1-score. SVM performed well, scoring 98.92% across accuracy, precision, recall, and F1-score. DT achieved 99.21% in all four metrics, while RF slightly improved with 99.78%. The optimized MLP model recorded 99.79% in accuracy, precision, recall, and F1-score. Notably, the proposed model outperformed all others, achieving the highest scores of 99.98% across all evaluation metrics.

Table 3: Comparison of Proposed Model with Other Machine Learning Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN	98.67	98.70	98.65	98.66
LR	97.85	97.90	97.82	97.85
NB	94.32	94.10	94.50	94.30
SVM	98.92	98.95	98.89	98.92
DT	99.21	99.23	99.20	99.21
RF	99.78	99.79	99.77	99.78
MLP	99.79	99.79	99.78	99.78
Proposed	99.98	99.98	99.98	99.98

Table 4: Performance of Proposed Model under 5-Fold Stratified Cross-Validation

Metric	Mean ± Std(%)
Accuracy	99.96±0.01
Precision	99.95±0.02
Recall	99.96±0.01
F1-Score	99.95±0.01
ROC-AUC	99.98±0.01

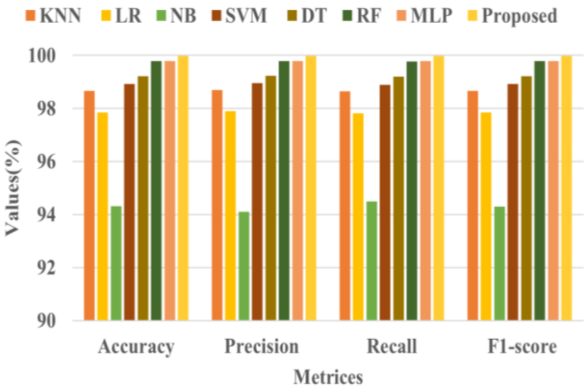


Fig. 3: Comparison analysis of the proposed model with other ML algorithms

This study used the MLP model because it can learn complicated, non-linear patterns in high-dimensional datasets like RT-IoT 2022. Traditional ML models like KNN, NB, LR, and ensemble techniques like RF and SVM work well, but they struggle to capture the complex correlations between data in current IoT traffic. DL model MLP is a flexible architecture for automatic feature extraction and representation learning. To maximize its potential, adjust activation function, learning rate, hidden layers, and solver approaches using hyperparameter tuning and optimization procedures. The improved MLP surpassed all benchmark models in detecting and classifying IoT security threats. Table 4 presents the performance of the proposed model under 5-fold stratified cross-validation. The proposed optimized MLP model is lightweight, with a size of approximately 2.1 MB, making it suitable for resource-constrained IoT environments. It achieves an inference latency of ~2.3 ms per flow on a CPU and under 1 ms on a GPU, with a memory footprint below 250 MB during runtime. These characteristics confirm its feasibility for real-time intrusion detection at IoT gateways. Although the optimized MLP is lightweight, further comparisons with TinyML and pruned DT would provide insights into edge deployment trade-offs.

Real World Deployment and Scalability Considerations

While the optimized MLP model demonstrates excellent performance in controlled experimental settings, practical deployment in large-scale IoT ecosystems introduces additional challenges. IoT devices are typically resource-constrained in terms of memory, power, and processing capability. To address this, the proposed model can be deployed in a hierarchical manner, where lightweight intrusion detection agents operate on edge nodes or gateways, and more computationally demanding analysis is performed in fog or cloud layers. The scalability of the approach is ensured by the reduced feature set (35 out of 85), which lowers the computational overhead and minimizes latency in classification. Furthermore, the high precision and recall rates reduce the risk of false alarms, making the system suitable for real-time monitoring across thousands of heterogeneous IoT devices. Future enhancements such as model compression, quantization, and incremental learning can further support deployment in dynamic and large-scale environments.

Limitations and Ethical Considerations

Although the optimized MLP framework achieves near-perfect detection performance, several limitations remain. First, the RT-IoT2022 dataset, while realistic, may not fully capture the diversity of IoT hardware platforms, network topologies, and evolving attack types such as zero-day or ransomware, introducing potential dataset bias. This raises concerns about the generalizability of the results when deployed in heterogeneous or large-scale IoT environments. Second, the exceptionally high accuracy scores suggest possible risks of overfitting to the dataset’s characteristics, highlighting the need for further evaluation on unseen or real-world data streams. Third, the study did not incorporate adversarial robustness testing, which is increasingly relevant given adaptive attack strategies. From a deployment perspective, while the model is lightweight, integration into distributed IoT networks may require additional considerations such as secure update mechanisms, compatibility with low-power devices, and scalability under high traffic loads. Ethical concerns also arise in intrusion detection, as monitoring network traffic could inadvertently expose sensitive user information. Ensuring data anonymization, compliance with privacy regulations, and responsible use of detection outputs are therefore essential. Future research should explore privacy-preserving training methods, such as differential privacy and federated learning, alongside robustness enhancements to strengthen the trustworthiness and adaptability of IoT intrusion detection systems.

Conclusion

IoT connects smart devices, sensors, and cloud-based infrastructures, transforming many industries. IoT settings are becoming more vulnerable to a variety of threats as they grow in size and complexity. DoS, DDoS, brute-force SSH attacks, network scans, and ARP poisoning threaten IoT ecosystem data confidentiality, system availability, and operational integrity. ML and DL have become useful tools for real-time attack detection and threat classification to address cybersecurity issues. We used the RT-IoT2022 dataset to optimize an MLP model that detects and classifies various cyberattacks. The realistic IoT dataset captured normal and hostile behaviors, providing a solid foundation for model training and evaluation. The optimized MLP model, with hyperparameter tweaking and feature selection, achieved high accuracy (99.9838%), precision, recall, and F1-scores nearing 99.98%. The model outperforms KNN, LR, NB, SVM, DT, and RF in distinguishing normal and malicious network activities with excellent reliability and resilience. Deploying the optimized model in real-time IoT contexts for live threat detection can expand this research. Federated learning for privacy-preserving training across distributed IoT devices and transformer-based models for feature extraction can be explored in the future. Adding attack scenarios and multi-source data streams could also increase the detection framework's generalization and adaptability in dynamic IoT environments.

Acknowledgment

The authors acknowledge the support provided by Chitkara University for facilitating the research environment required for this study.

Funding Information

This research did not receive any funding.

Authors Contributions

Sanchit Vashisht: Performed the data preprocessing, model design, optimization, experiments, and result analysis.

Shalli Rani: Supervised the research work, provided technical guidance, validated the methodology, and reviewed the overall manuscript. Both authors reviewed and approved the final version of the manuscript.

Ethics

This research uses a publicly available dataset and does not involve human participants, animals, or sensitive personal information. The authors declare no conflict of interest.

References

Airlangga, G. (2024). Deep Learning for Anomaly Detection and Fraud Analysis in Blockchain Transactions of the Open Metaverse. *Jurnal Informatika Ekonomi Bisnis*, *6*(2), 324–329. https://doi.org/10.37034/infeb.v6i2.865

Alabsi, B., Anbar, M., & Rihan, S. (2023). CNN-CNN: Dual Convolutional Neural Network Approach for Feature Selection and Attack Detection on Internet of Things Networks. *Sensors*, *23*(14), 6507. https://doi.org/10.3390/s23146507

Alani, M. M., & Damiani, E. (2023). XRecon: An Explainbale IoT Reconnaissance Attack Detection System Based on Ensemble Learning. *Sensors*, *23*(11), 5298. https://doi.org/10.3390/s23115298

Alhchaimi, A. (2024). Cloud-Based Transaction Fraud Detection: An In-depth Analysis of ML Algorithms. *Wasit Journal of Computer and Mathematics Science*, *3*(2), 19–31. https://doi.org/10.31185/wjcms.253

Cherfi, S., Lemouari, A., & Boulaiche, A. (2025). MLP-Based Intrusion Detection for Securing IoT Networks. *Journal of Network and Systems Management*, *33*(1), 1–31. https://doi.org/10.1007/s10922-024-09889-7

Farooqi, A. H., Akhtar, S., Rahman, H., Sadiq, T., & Abbass, W. (2023). Enhancing Network Intrusion Detection Using an Ensemble Voting Classifier for Internet of Things. *Sensors*, *24*(1), 127. https://doi.org/10.3390/s24010127

Gürfidan, R. (2024). Suspicious transaction alert and blocking system for cryptocurrency exchanges in metaverse's social media universes: RG-guard. *Neural Computing and Applications*, *36*(30), 18825–18840. https://doi.org/10.1007/s00521-024-10122-4

Hisham, S., Makhtar, M., & Abdul Aziz, A. (2023). Anomaly detection in smart contracts based on optimal relevance hybrid features analysis in the Ethereum blockchain employing ensemble learning. *International Journal of Advanced Technology and Engineering Exploration*, *10*(109), 1552–1579. https://doi.org/10.19101/ijatee.2023.10102216

Malik, K. M., & Dutta, M. (2023). Feature Engineering and Machine Learning Framework for DDoS Attack Detection in the Standardized Internet of Things. *IEEE Internet of Things Journal*, *10*(10), 8658–8669. https://doi.org/10.1109/jiot.2023.3245153

Kamran, M., Rehan, M. M., Nisar, W., & Rehan, M. W. (2024). AHEAD: A Novel Technique Combining Anti-Adversarial Hierarchical Ensemble Learning with Multi-Layer Multi-Anomaly Detection for Blockchain Systems. *Big Data and Cognitive Computing*, *8*(9), 103. https://doi.org/10.3390/bdcc8090103

Li, M. (2024). Meta-universe Financial Transaction Anomaly Detection and Risk Prediction based on Machine Learning. *Proceedings of the 2024 2nd International Conference on Image, Algorithms and Artificial Intelligence (ICIAAI 2024)*, *276*, 117–129. https://doi.org/10.2991/978-94-6463-540-9_14

Mehmood, S., Amin, R., Mustafa, J., Hussain, M., Alsubaei, F. S., & Zakaria, M. D. (2025). Distributed Denial of Services (DDoS) attack detection in SDN using Optimizer-equipped CNN-MLP. *PLOS ONE*, *20*(1), e0312425. https://doi.org/10.1371/journal.pone.0312425

Saiyed, M. F., & Al-Anbagi, I. (2024a). A Genetic Algorithm- and t-Test-Based System for DDoS Attack Detection in IoT Networks. *IEEE Access*, *12*, 25623–25641. https://doi.org/10.1109/access.2024.3367357

Saiyedand, M. F., & Al-Anbagi, I. (2024b). Deep Ensemble Learning With Pruning for DDoS Attack Detection in IoT Networks. *IEEE Transactions on Machine Learning in Communications and Networking*, *2*, 596–616. https://doi.org/10.1109/tmlcn.2024.3395419

Sharma, A., & Babbar, H. (2023). Machine Learning-Driven Detection and Prevention of Cryptocurrency Fraud. *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*. https://doi.org/10.1109/rmkmate59243.2023.10369055

Srinivasan, B. (2024). Navigating Financial Transactions in the Metaverse: Risk Analysis, Anomaly Detection, and Regulatory Implications. *International Journal Research on Metaverse*, *1*(1), 59–76. https://doi.org/10.47738/ijrm.v1i1.5

Srivastava, A., Tiwari, S., Saini, P. K., Sawan, V., & Dhondiyal, S. A. (2023). Attack Detection and Mitigation in IoT using SVM. *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*. https://doi.org/10.1109/icaiss58487.2023.10250530

Venčkauskas, A., Grigaliūnas, Š., Pocius, L., Brūzgienė, R., & Romanovs, A. (2025). Machine Learning in Money Laundering Detection Over Blockchain Technology. *IEEE Access*, *13*, 7555–7573. https://doi.org/10.1109/access.2024.3452003