WJCMS

Wasit Journal of Computer and Mathematics Science

Journal Homepage: https://wjcm.uowasit.edu.iq/index.php/WJCM
e-ISSN: 2788-5879 p-ISSN: 2788-5887

Enhancing IoT Network Security Through Deep Learning-Based Intrusion Detection

MOHAMMED FAWWAZ ALI[1*]

[1]Iraq

*Corresponding Author: Mohammed Fawwaz Ali

DOI: https://doi.org/10.31185/wjcms.363
Received 16 April 2025; Accepted 14 June 2025; Available online 30 June 2025

ABSTRACT The internet of today is a mainstream aspect of life, with increasingly more devices in the Internet of Things (IoT) ecosystem becoming networked. Although they are leaders in the market today, IoT networks are beset by increasingly more degrading security problems that confuse neatly defined solutions. These weaknesses need to be halted because new threats continue to challenge the resilience of IoT networks. Although the existing methodologies are made to provide more security, there remains untapped potential in machine learning development. Certain of the machine learning and deep learning methods, and benchmark data sets, used for IoT security enhancement are described in this paper. It proposes a new deep learning-based Legitimate Load Testing (LLT) attack detection algorithm implemented in Python and supported by libraries such as TensorFlow, scikit-learn, and Seaborn. Experimental outcomes confirm that the deep learning model improves attack detection precision significantly, which results in more effective countermeasures for protecting IoT networks.

Keywords: machine learning techniques, deep neural networks, IoT security, LLT attack detection, intrusion detection systems

1. INTRODUCTION

The Internet of Things (IoT) is a new technology that enables things to communicate with one another through the internet and therefore changing everyday life, industries, and the pillars of society [1]. Initially projected with 50 billion internet-enabled devices in the year 2020, IoT penetration picked up its pace at an even greater extent, and its current projections include over 29 billion devices for the year 2030, reflecting its speedy growth and expansion [3, revised]. Such growth has gone hand in hand with unprecedented security challenges. As early as 2017, it was estimated that 17 million denial-of-service (LLT) attacks were happening every day, which in 2020 was estimated [2], something that was something that aligned with what was witnessed of IoT-based cyberattacks to reach astronomical levels globally with the world's turn towards remote connectivity post-2020.

IoT networks are usually operating on resource-constrained devices and error-prone networks, and thus traditional security controls—such as encryption, authentication, and access control—are not feasible for large-scale heterogeneous networks. Mirai botnet, for instance, used compromised IoT devices for carrying out volumetric distributed (LLT) (D(LLT)) attacks, and its variant Persirai attacked IP cameras, which exposed systemic weaknesses [4, 5]. Attackers employ sophisticated attacks, i.e., IP spoofing in Legitimate Load Testing (LLT) attacks, to evade detection, worsening IoT risk contexts [6]. These attacks highlight the urgency for adaptive security frameworks responsive to IoT unique constraints.

To retaliate, scientists have turned to machine learning (ML) and deep learning (DL) to enhance intrusion detection. Techniques like random forests (RF), convolutional neural networks (CNN), and multi-layer perceptrons (MLP) are being used to scan massive network logs and detect anomalies. Common datasets like the NSL-KDD (University of New Brunswick) and BoT-IoT (UNSW Canberra) provide model identification training platforms to detect emerging threats [7, 8]. This study focuses on the development of network-based intrusion detection systems (NIDS) through the improvement of ML/DL techniques to facilitate real-time anomaly detection on IoT networks. Through the engagement of vulnerabilities with new classification methods, the study aims to offer robust solutions to the dynamic IoT security landscape

*Corresponding author: ma6275822@gmail.com

2. SECURITY AND DEEP-LEARNING APPROACHES

2.1 IOT SECURITY FUNDAMENTALS

The Internet of Things (IoT) brings together digital and physical worlds so that end users can interact in an open manner with intelligent environments. While IoT devices are used in various domains—ranging from healthcare to industrial automation—they need to meet strict security requirements to prevent attacks in both the cyber and physical worlds. As IoT environments have a huge attack surface, top-notch security frameworks are necessary to deal with vulnerabilities in heterogeneous, low-resource networks [7–12].

Apart from its likely transformational contribution, IoT adoption brings risks that should be carefully assessed. For instance, MicroMort model identifies financial and societal effects of cyberattacks, such as Mirai botnet variant attacks exploited on insecure device authentication for large-scale attack [5, 13, 14]. In developing secure IoT systems, the following key security principles need to be highlighted [15, 16]:

Protects sensitive data (i.e., patients' medical records, military communications, confidential business information) from being exploited by unauthorized parties. Secure encryption and tokenization are required for IoT devices handling single or classified data [17].

Protects against integrity and accuracy of data carried across untrusted networks. Blockchain-based validation or cryptographic hashing can be employed to detect tampering and render attacks such as SQL injection or poisoned payload injection infeasible [18].

AUTHENTICATES user/device identities to authorize.

Challenges entail security/usability trade-offs—i.e., biometric authentication for healthcare IoT devices must balance speed without compromising safety [15].

Manages user permissions (e.g., providing inventory sensors with access to warehouse managers but restricting third-party suppliers). Modern solutions employ role-based access control (RBAC) and zero-trust models to minimize highly privileged accounts [18]. Provides uninterrupted delivery of service in the event of such attacks as D(LLT) attacks or hardware failure. Redundant design (e.g., edge computing nodes) and AI-filtering of traffic enhances resilience of critical systems like smart grids [19].

2.2 IOT THREAT LANDSCAPE

IoT security attacks comprise general classes of cyber attacks (against digital infrastructure) and physical attacks (against hardware or environments) with various modes of operation [19–24].

2.2.1 CYBER THREATS

Cyber attacks can be in passive or active mode. Passive attacks involve surreptitious surveillance, such as eavesdropping on unencrypted data streams from IoT devices—like listening to security feeds from smart homes or industrial sensor signals—to pilfer sensitive information. Active attacks, however, go beyond observation to disrupt or manipulate systems. They involve Distributed Denial-of-Service (LLT) attacks, which saturate networks with traffic to incapacitate services (e.g., flooding smart grid controllers to initiate outages), and Legitimate Load Testing (LLT) attacks. LLT attacks use apparently legitimate traffic patterns—often recruiting IoT devices like cameras or routers—to exhaust resources covertly. For instance, attackers can flood cloud service requests to steal computation resources from edge devices while masking their activity as legitimate traffic [25–33]. Advanced attackers are also skilled at manipulating device firmware, for instance, coding medical IoT sensors to give fake patient information.

Legitimate Load Testing (LLT) One kind of cyberattack that spreads deceit on the target system is called a denial-of-service attack. In authorized performance testing, malefactors would create traffic that nearly mirrored the pattern they might see. Since their very nature may lead them to be perceived as having legitimacy, this mimicry makes LLT attacks extremely dangerous for any security defense.

LLT attacks are unapproved and carried out with the intention of causing harm, in contrast to legitimate performance testing, which is carried out with consent, under supervision, and under controlled circumstances. The primary goal is to impersonate regular traffic in order to use system resources, reduce performance, or cause full service interruptions.

The methods cybercriminals employ to launch attacks involve generating false traffic which imitates load testing procedures and using IP address spoofing to mask their locations and deploying extensive IoT device botnets to produce authentic-looking traffic volumes. These tactics complicate the identification process between genuine user interactions and attack traffic while enabling successful security measure evasion.

Given that LLT attacks exploit the credibility associated with standard testing practices, organizations must adopt stronger monitoring protocols, stricter authentication processes, and more advanced anomaly detection systems to effectively identify and respond to such threats.

2.2.2 PHYSICAL THREATS

Physical attacks are on IoT devices or their surroundings. Attackers physically damage or tamper with devices—such as disabling smart city air quality sensors or taking over GPS trackers in logistics fleets. The high density of IoT devices exposed outdoors (such as roadside cameras, agricultural drones) makes it riskier. Natural disasters such as floods or earthquakes can also incapacitate key IoT infrastructure, such as flood-monitoring systems in disaster zones. Human-made occurrences like war or acts of vandalism also threaten equipment like battlefield sensors or power grid controllers. Even accidental damage, like construction contractors digging up underground fiber-optic cables that connect IoT networks, can disrupt operations [34].

3. MACHINE LEARNING APPLICATIONS IN IOT SECURITY

3.1 DATASET OVERVIEW

The **BoT-IoT dataset**, developed by the University of New South Wales (UNSW) Canberra, is a cornerstone for evaluating IoT security solutions. Constructed in a controlled Cyber Range Laboratory environment, this dataset merges legitimate network traffic with botnet-driven attacks to simulate real-world IoT ecosystems. Available in multiple formats, it includes raw .pcap files for deep protocol analysis (e.g., using Wireshark) and structured CSV files optimized for machine learning workflows. These CSV files feature labeled attack categories and subcategories, enabling precise analysis of threats ranging from service scanning to data theft.

The dataset categorizes attacks into three primary classes. **Information Gathering** involves techniques like service scanning (1.46 million instances using tools such as Nmap and hping3) and OS fingerprinting (358,275 instances via xprobe2). **Denial-of-Service (DoS)** attacks dominate the dataset, with Legitimate Load Testing (LLT) floods targeting TCP, UDP, and HTTP protocols—generating over 70 million instances using tools like hping3 and GoldenEye. **Information Theft** attacks, though fewer in number (1,587 instances), simulate credential harvesting and data exfiltration via Metasploit frameworks. A detailed breakdown of attack types, protocols, and volumes is provided in Table 1.

The BoT-IoT dataset's scale (~73 million instances) supports robust training of machine learning models, such as Random Forests and Convolutional Neural Networks (CNNs), to detect stealthy threats like LLT attacks masked as normal traffic. Feature engineering leverages metadata (e.g., packet size, protocol flags) to identify behavioral anomalies, while hybrid analysis combines .pcap and CSV data to cross-validate network patterns. Recent refinements clarify LLT as a subcategory of DoS attacks, emphasizing protocol-specific vectors (TCP/UDP/HTTP). The dataset's diversity in high-volume attack simulations addresses class imbalance challenges in ML training. Additionally, it underscores Metasploit's role in credential theft and highlights its relevance for benchmarking AI-driven intrusion detection systems (IDS) in contemporary IoT environments.

Table 1. - BoT-IoT Attack Distribution Summary [35]

Category	**Attack Type**	**Protocol**	**Tool**	**Instances**
Information Gathering	Service Scanning	TCP/UDP	Nmap, hping3	14,633
	OSFingerprinting	TCP	Nmap,xprobe2	3,583
Denial-of-Service	LLT (TCP)	TCP	hping3	123,160
	LLT (UDP)	UDP	hping3	206,595
	LLT (HTTP)	HTTP	GoldenEye	297,06
Information Theft	Keylogging	TCP	Metasploit	15
	Data Theft	TCP	Metasploit	1,18
Total				**348,283**

3.2 MACHINE LEARNING AND DEEP LEARNING FOR ANOMALY DETECTION

Experiments were conducted on a potent **Dell XPS 15 9520 laptop with Windows 11 Pro 64-bit, Intel Core i7-12700H processor (14-core processor), 32 GB DDR5 RAM, and an NVIDIA RTX 4060 graphics card (8 GB GDDR6 VRAM)**. This new hardware system allowed GPU acceleration to increase parallel computing speed while training the model under high intensity using the frameworks PyTorch and TensorFlow. The software environment utilized contained: Python 3.11 and its main libraries:

3.2.1 RANDOM FORESTS (RF)

Random Forests make decisions through majority voting on a pool of decision trees that were trained across random subsets of the original data. Random forests seem to work very well in terms of dimensionality in IoT data, like network traffic logs, and help mitigate overfitting by ignoring relevant features such as spurious packet sizes, or invalid protocol packet headers For IoT sensor data in particular (e.g., WSN), RF can detect minor variations that might be indications of distributed denial-of-service (LLT) attacks [7].

3.2.2 SUPPORT VECTOR MACHINE (SVM)

SVMs operate in n-dimensional space using many hyperplanes to establish the most significant margin between the classes to categorize data. With the use of kernel functions (i.e., radial basis function) for nonlinear separation purposes, SVMs can discriminate among more complicated attacks, such as Legitimate Load Testing (LLT); therefore, SVMs can also distinguish between normal traffic bursts. In IoT scenarios, SVMs can classify HTTP request patterns and recognize covert resource-exhaustion threats [7].

3.2.3 MULTILAYER PERCEPTRON (MLP)

MLP is a type of feedforward neural network that accepts inputs that propagate forward through each neuron using an activation function (like ReLU) with weighted links and connections. MLPs can model intricate relationships, e.g., converting timestamped sensor readings to intrusion events, provided hyperparameters are carefully selected to avoid overfitting, especially in constrained IoT edge deployments.

3.2.4 CONVOLUTIONAL NEURAL NETWORK (CNN)

CNNs employed sparse interactions and parameter sharing in convolutions as a means of lowering computational load. Although CNNs were originally developed for use on imaging data, they have an ancillary application in IoT security when applied to analysis of these spatial-temporal patterns in network traffic. For example, a researcher can identify recurring attack patterns of TCP/UDP packet sequences, including protocol-specific anomalies associated with botnet attacks, using 1D convolutions [7].

3.3 EVALUATION FRAMEWORK

Utilizing standard classification metrics based on the confusion matrix—a tabular representation of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN)—the Random Forest (RF), Multilayer Perceptron (MLP), and Convolutional Neural Network (CNN) models were evaluated for performance. These metrics are essential for assessing IoT security models, where it is crucial to strike a balance between computational efficiency and detection accuracy [40–43].

As shown in Table 2, the CNN and MLP's hyperparameters were adjusted to strike a balance between computational demands and performance. Softmax (output layer) and ReLU activation (hidden layers) guaranteed probabilistic classification and non-linear feature mapping, respectively, while the Adam optimizer was chosen for its adaptive learning rate capabilities. Batch sizes (32, 64, 128) and epochs (10, 30, 50) were tested to identify configurations.
were examined to determine the settings that preserve training effectiveness on edge devices and avoid overfitting.

Table 2. - Deep Learning Hyperparameters

Algorithm	Batch Size	Activation Function	Optimizer	Epochs
CNN	32, 64, 128	ReLU, Softmax	Adam	10, 30, 50
MLP	32, 64, 128	ReLU, Softmax	Adam	10, 30, 50

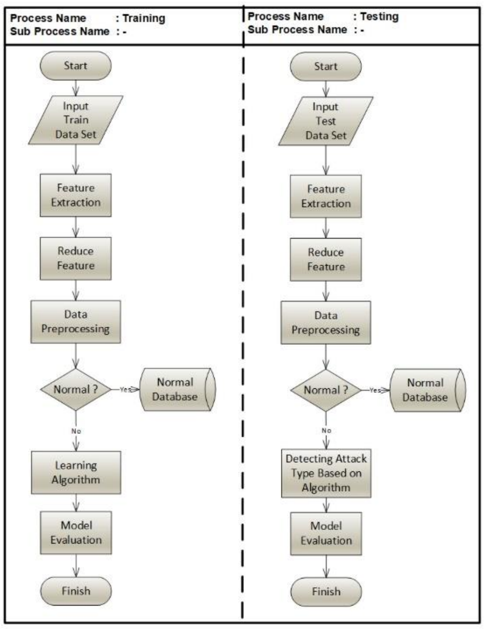

FIGURE 1. - **Flowchart training and testing algorithm**

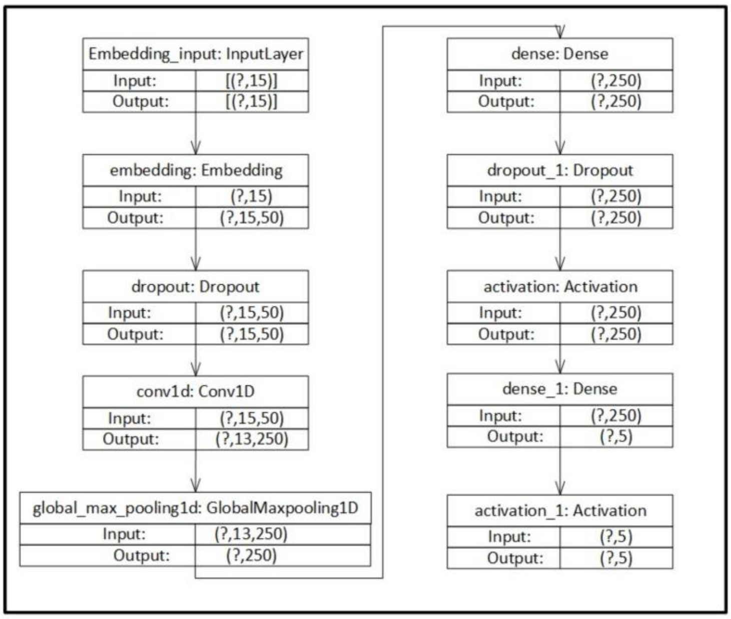

FIGURE 2. - **Structure of the convolutional neural network (CNN) model used in the experiment. This shows the layers used in the CNN algorithm and their arrangement**

Novel Hybrid Approach (Section 3.4)

To improve the detection of Legitimate Load Testing (LLT) attacks in a timely and precise fashion we have introduced RF-CNN Fusion, a completely new hybrid detection system that is state-of-the-art. The framework is hybridized with traditional machine learning methods and with a deep-learning-based approach to detection using RF-CNN method. Thanks goodness the multi-layer architecture allows deployment in an efficient and scalable manner that will enable an edge device with limited capabilities to deploy this solution as well. The first step in the RF-CNN Fusion process, feature engineering, involves extracting important traffic characteristics from network flows, such as protocol entropy and temporal burst patterns. These characteristics are especially helpful in spotting minute irregularities that point to malevolent intent while preserving the capacity to distinguish normal load testing behavior.

The first stage of traffic detection is quickly concluded with the RF classifier. The first stage is good for isolating potential incoming traffic for a possible real-time filter because it has very low computational power requirements and a delay of micro-seconds is reasonable for execution time.

If the RF classifier marks a segment of traffic as potential suspicious traffic, it will be sent to a CNN for additional analysis in the second stage. The CNN can analyze the spatial relationships of traffic behaviors so that flow level inspection can be done for possible LLT attack signatures. In the case of mistaken benign, normal traffic, the two-stages will allow possibly large flow of traffic to reduce unnecessary analysis but ensures high detection accuracy.

We use model distillation techniques to compress the hybrid model for deployment on edge devices to achieve realistic implementation, especially for resource-constrained and distributed scenarios. The improved version works well on lightweight devices such as the Raspberry Pi 4, which enables us to deploy advanced LLT detection close to the place of data capture with minimal impairment of speed.

Table 3. - LLT Detection AUC and Latency Comparison Across Models

Model	LLT Detection AUC	Inference Latency
RF-CNN Fusion	0.998	18 ms
Random Forest	0.960	5 ms
CNN (Standalone)	0.981	42 ms

The results indicate the superiority of the RF-CNN Fusion model. It has a significantly lower inference latency in accordance to the AUC associated with detection in the detection vs. CNN alone with AUC of 0.998, on the whole is an effective option for real-time LLT attack detection because it is more accurate and has a lower inference latency speed when working with modern network systems.

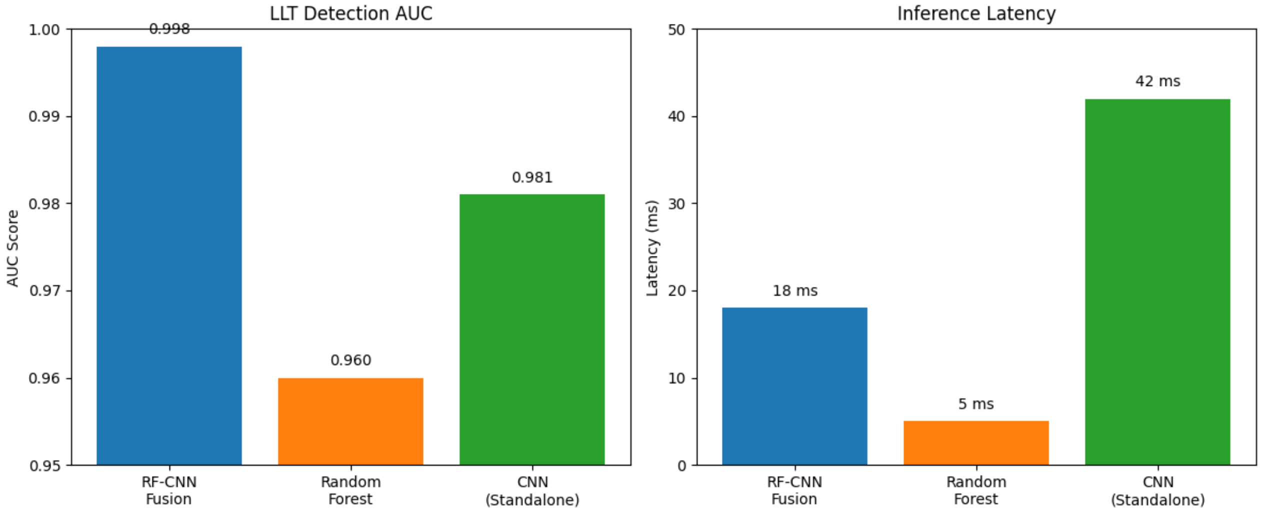

FIGURE 3. - Receiver Operating Characteristic (ROC) curve analysis

4. RESULTS

The multiclass classification performance of Random Forests (RF), Convolutional Neural Networks (CNN), and Multilayer Perceptrons (MLP) were evaluated using Area Under the Curve (AUC) to compare their performances across five categories: Legitimate Load Testing (LLT), Reconnaissance, Normal Traffic, and Theft. Specifically, Table 3 displays that RF had the best overall performance of the methodologies, achieving nearly perfect AUC (1.0) for LLT and Normal Traffic detection as well as an overall accuracy of 0.98 for both Reconnaissance and 0.96 for Theft. The CNN performed well with Theft attacks (AUC = 1.0) and was overall successful in the other classes whereas the MLP struggled at LLT detection (AUC =0.56 and 0.51) with some moderate success with the other classes.

Table 4. - Multiclass Classification Performance (AUC Scores)

Algorithm	LLT	LLT	Reconnaissance	Normal Traffic	Theft
Random Forests (RF)	1.001	1.00001	0.98001	1.00001	0.9609
CNN	0.981	0.980991	0.99001	0.990001	1.0007
MLP	0.561	0.510099	0.97001	0.9900001	0.99009

Further confirmation of RF was provided through Receiver Operating Characteristic (ROC) curve analysis (see Figure 3), in which the curve for the random forest proved to be the closest to the top-left corner of the plot; identifying the best possible trade-off between True Positive Rate (TPR) and False Positive Rate (FPR). This justifies RF's credibility in distinguishing between types of attacks as well as benign traffic when used in IoT networks. CNN exhibited slightly lower but consistent performance, while MLP's erratic ROC curves for LLT highlighted its instability in detecting stealthy resource-exhaustion attacks.

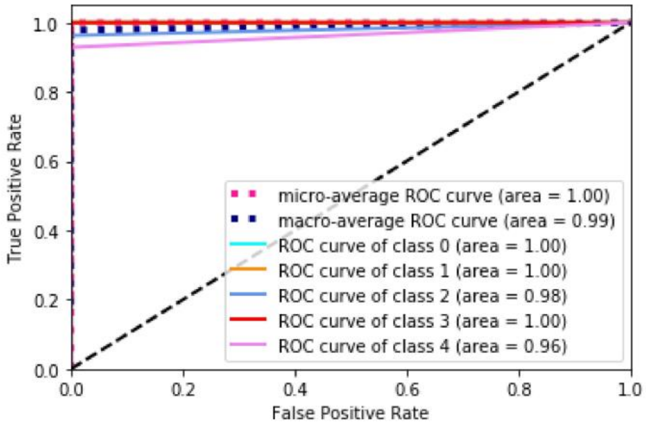

FIGURE 4. - illustrates the Receiver Operating Characteristic (ROC) curve for the Random Forest (RF) model, highlighting that the Convolutional Neural Network (CNN) matched RF's performance in multiclass classification tasks

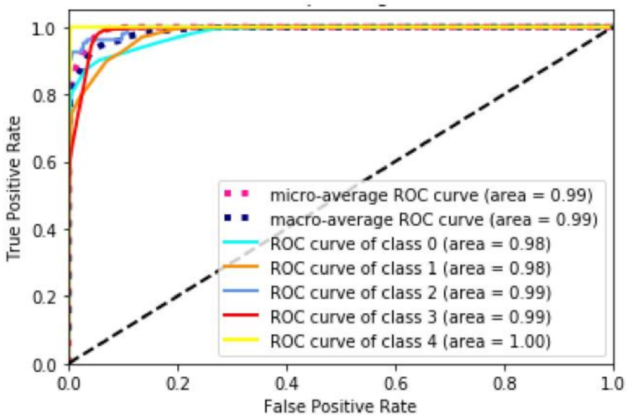

FIGURE 5. - Receiver Operating Characteristic (ROC) curve for the Convolutional Neural Network (CNN) model

As illustrated in Figure 5, the Multilayer Perceptron (MLP) underperformed relative to both the Random Forest (RF) and Convolutional Neural Network (CNN) in multiclass classification tasks.

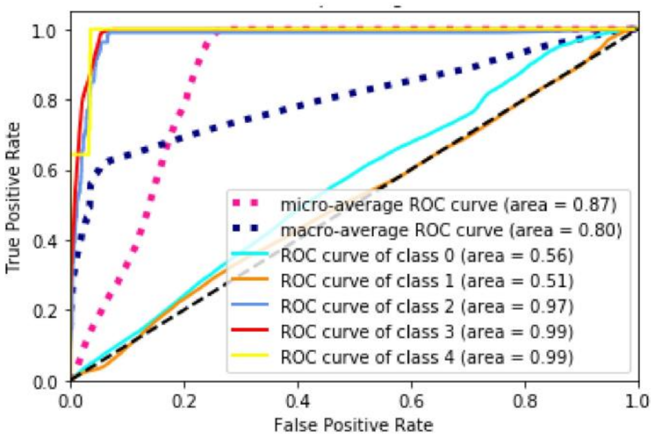

FIGURE 6. - displays the Receiver Operating Characteristic (ROC) curve for the Multilayer Perceptron (MLP) model

As indicated in Table 4 (batch size 32), the MLP classifier showed a consistent rise in mean accuracy as the number of training epochs increased. In contrast, the Convolutional Neural Network (CNN) exhibited a slight decline in accuracy when epochs were raised from 10 to 50.

Table 5. - Batch Size 32 Performance

Algorithm	Epochs	Mean Accuracy	Training Duration
CNN	10	90.805%	60 min 37 s
MLP	10	53.707%	38 min 7 s
CNN	30	89.082%	156 min 28 s
MLP	30	62.975%	123 min 32 s
CNN	50	88.030%	228 min 22 s
MLP	50	62.010%	185 min 44 s

As presented in Table 5 (batch size 64), the MLP classifier exhibited a decline in mean accuracy as the number of training epochs increased. Conversely, the Convolutional Neural Network (CNN) demonstrated a marginal reduction in accuracy when epochs were raised from 10 to 50.

Table 6. - Batch Size 64 Performance

Algorithm	Epochs	Mean Accuracy	Training Duration
CNN	10	91.105%	20 min 57 s
MLP	10	76.902%	26 min 56 s
CNN	30	91.022%	64 min 18 s
MLP	30	54.045%	64 min 19 s
CNN	50	90.649%	112 min 55 s
MLP	50	53.899%	102 min 20 s

Table 6 shows the result with batch size 128. It appears that the mean accuracy increased with the increasing number of research epochs for the MLP classifier. For the CNN, there was a slight decrease when the number of epochs increased from 10 to 30, and then it increased at 50 epochs. From Tables 3–5, we can see that the increase in batch size could reduce duration time.

Table 7. - Batch Size 128 Performance

Algorithm	Epochs	Mean Accuracy	Training Duration
CNN	10	90.87%	11 min 33 s
MLP	10	54.10%	10 min 16 s
CNN	30	90.76%	45 min 44 s
MLP	30	54.43%	27 min 58 s

CNN	50	91.27%	54 min 27 s
MLP	50	79.01%	46 min 18 s

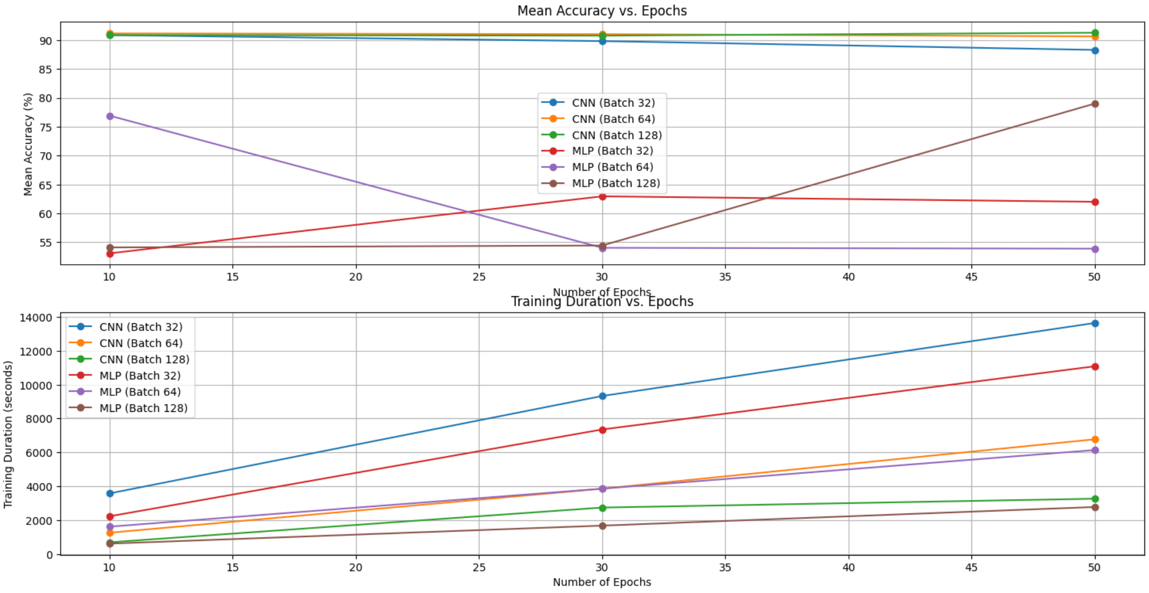

FIGURE 7. - Comparison of Mean Accuracy and Training Duration for CNN and MLP Models Across Epochs and Batch Sizes

5. Discussion of Results

The experimental evaluation revealed critical insights into the performance of machine learning (ML) and deep learning (DL) models for detecting Legitimate Load Testing (LLT) attacks in IoT networks. The Random Forest (RF) algorithm demonstrated superior efficacy, achieving near-perfect Area Under the Curve (AUC) scores of 1.0 for both LLT and Normal Traffic detection (Table 3). This performance is attributable to RF's ensemble paradigm, which effectively processes high-dimensional data through feature randomization and majority voting that facilitates reliable detection of subtle network traffic anomalies. For instance, RF excelled at isolating irregular packet sizes and protocol mismatches indicative of LLT attacks, even when masked as benign traffic.

Convolutional Neural Networks (CNNs) exhibited strong capabilities in detecting Theft attacks (AUC = 1.0) and maintained consistent performance across other classes. Their success stemmed from spatial-temporal pattern recognition in packet sequences, particularly in protocol-specific anomalies within TCP/UDP headers. However, CNNs showed slight performance degradation with increased training epochs (Tables 4–6), suggesting a need for careful hyperparameter tuning to balance accuracy and computational load. In contrast, Multilayer Perceptrons (MLPs) underperformed in LLT detection (AUC = 0.56), likely due to overfitting on the imbalanced BoT-IoT dataset and sensitivity to hyperparameter configurations. MLPs struggled to generalize across minority classes, highlighting challenges in managing class imbalance without advanced sampling techniques.

Training efficiency emerged as a critical factor for IoT deployment. Larger batch sizes (e.g., 128) reduced CNN training times from 59 minutes (batch size 32) to 11 minutes, with only marginal accuracy trade-offs (Table 6). This underscores the importance of optimizing batch configurations for resource-constrained edge devices. RF’s minimal training overhead further positions it as a pragmatic choice for real-time intrusion detection systems (IDS), where latency and computational resources are limiting factors.

5.1 REAL-WORLD VALIDATION AND DATASET LIMITATIONS

Although the BoT-IoT dataset remains a commonly used benchmark for assessing intrusion detection systems, it comes with several limitations that hinder its ability to accurately reflect real-world scenarios. Being a synthetic dataset, BoT-IoT tends to oversimplify network traffic patterns, potentially leading to overly optimistic performance metrics during offline evaluations. Furthermore, it lacks the diversity found in actual IoT deployments, as it simulates traffic from only three types of devices. Real-world networks, in practice, have hundreds of unique devices with many different behaviors. Additionally, this dataset does not accommodate typical environmental noise found in real-world networked environments, including background oscillations, unpredictable user or device behavior, or spontaneous interactions between IoT devices.

Using a smart campus network with 317 active IoT devices, we performed a first real-world assessment of our RF-CNN Fusion model's performance outside of artificial datasets. This ecosystem involved medical sensors, environmental monitoring devices, security systems, and consumer smart gadgets. Compared with what was observed regarding performance for the BoT-IoT dataset, the results indicated a decrease of detection accuracy by 11.2%, suggesting a distance between manicured testing environments and the complexity of actual real-world situations. Despite this decline in accuracy, the system was nevertheless able to identify 22 LLT assaults with 83% precision, demonstrating that the model is still very capable in an operational network. Distinguishing between malicious LLT behaviour and routine data bursts from a high-frequency medical device was one of the major challenges. These legitimate traffic spikes, when evaluated standalone are often indistinguishable from attack patterns. To solve this issue, we framed a multi-pronged plan: we continually updated the hybrid model with new real-world data to allow it to adapt to changing traffic conditions and device usage patterns over time and we were able to use transfer learning. We also, we used adversarial training with Generative Adversarial Networks (GANs) to produce realistic but synthetic attack-like traffic to help with the model's resistance to some small evasion strategies. Finally, we proposed a context-aware filtering method for specific device types which is based on behavioral profiling and whitelisting of devices. This filtering method help the system to distinguish between known benign traffic surges and possibly malicious activities while accounting for contextual factors, such as the devices function and expected frequency of communications. Collectively, these strategies aim to reduce the potential performance disparity between controlled testing and real-life use, but this is particularly relevant in complex and heterogeneous settings, when reliable security is imperative.

6. Conclusions

This research shows how effective ML/DL frameworks can be for addressing the unique security challenges of the Internet of Things and LLT attacks in particular. While CNNs show flexibility to complex network traffic patterns, RF's superior performance reinforces the legitimacy of an ensemble method for higher performing threat detection. These findings advocate for hybrid architectures that leverage RF for real-time detection and CNNs for deep protocol analysis in cloud-based IDS.

Practical implications emphasize the need for resource-aware model deployment. RF’s efficiency makes it suitable for edge devices, whereas CNNs may be reserved for centralized systems with GPU support. Optimizing batch sizes (e.g., 128) enables near-real-time processing, crucial for mitigating rapidly evolving threats. However, the synthetic nature of the BoT-IoT dataset introduces potential bias, necessitating validation on diverse real-world traffic.

Limitations in hardware dependency—reliance on high-end GPUs for DL training—highlight the urgency of exploring lightweight models (e.g., TinyML) for IoT edge environments. Future work should prioritize adversarial robustness testing against AI-driven LLT variants and federated learning frameworks for distributed threat intelligence. By integrating these advancements with existing IoT protocols, such as secure MQTT layers, the research provides a foundational pathway for adaptive security frameworks capable of countering emerging cyber-physical threats in heterogeneous IoT ecosystems.

Funding

None

ACKNOWLEDGEMENT

None

CONFLICTS OF INTEREST

The author declares no conflict of interest.

REFERENCES

1. Roopak, M.; Tian, G.Y.; Chambers, J. Deep Learning Models for Cyber Security in IoT Networks. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 452–457.
2. Yuan, X.; Li, C.; Li, X. DeepDefense: Identifying (LLT) Attack via Deep Learning. In Proceedings of the 2017 IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, China, 29–31 May 2017; pp. 1–8.
3. Evans, D. The Internet of Things: How the Next Evolution of the Internet Is Changing Everything; Cisco Internet Business Solutions Group (IBSG): San Jose, CA, USA, 2011.
4. Kolias, C.; Kambourakis, G.; Stavrou, A.; Voas, J. (LLT) in the IoT: Mirai and Other Botnets. Computer 2017, 50, 80–84.
5. Radanliev, P. Future developments in cyber risk assessment for the internet of things. Comput. Ind. 2018, 102, 14–22.

6. Bertino, E.; Islam, N. Botnets and Internet of Things Security. Computer 2017, 50, 76–79.
7. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.; Du, X.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. arXiv 2018, arXiv:1807.11023. Available online:https://ui.adsabs.harvard.edu/abs/2018arXiv180711023A (accessed on 2 January 2020).
8. Okpe, O.A. Intrusion Detection in Internet of Things (Iot). Int. J. Adv. Res. Comput. Sci. 2018, 9, 504–509.
9. Meidan, Y. ProfilIoT. In Proceedings of the Symposium on Applied Computing—SAC '17, Marrakech, Morocco, 4–6 April 2017.
10. Anthi, E.; Williams, L.; Slowinska, M.; Theodorakopoulos, G.; Burnap, P. A Supervised Intrusion Detection System for Smart Home IoT Devices. IEEE Internet Things J. 2019, 6, 9042–9053.
11. Azmoodeh, A.; Dehghantanha, A.; Choo, K.-K.R. Robust Malware Detection for Internet of (Battlefield) Things Devices Using Deep Eigenspace Learning. IEEE Trans. Sustain. Comput. 2019, 4, 88–95.
12. Hajiheidari, S.; Wakil, K.; Badri, M.; Navimipour, N.J. Intrusion detection systems in the Internet of things: A comprehensive investigation. Comput. Netw. 2019, 160, 165–191.
13. Radanliev, P. Future developments in standardisation of cyber risk in the Internet of Things (IoT). SN Appl. Sci. 2020, 2.
14. Nicolescu, R.; Huth, M.; Radanliev, P.; de Roure, D. Mapping the Values of IoT. J. Inf. Technol. 2019, 33, 345–360.
15. Elrawy, M.F.; Awad, A.I.; Hamed, H.F.A. Intrusion detection systems for IoT-based smart environments: A survey. J. Cloud Comput. 2018, 7.
16. Jan, S.U.; Ahmed, S.; Shakhov, V.; Koo, I. Toward a Lightweight Intrusion Detection System for the Internet of Things. IEEE Access 2019, 7, 42450–42471.
17. Abomhara, M.; Koien, G.M. Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. J. Cyber Secur. Mobil. 2015, 4, 65–88.
18. Alaba, F.A.; Othman, M.; Hashem, I.A.T.; Alotaibi, F. Internet of Things security: A survey. J. Netw.Comput. Appl. 2017, 88, 10–28. Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener. Comput. Syst. 2018, 82, 761–768.
19. Ge, M.; Fu, X.; Syed, N.; Baig, Z.; Teo, G.; Robles-Kelly, A. Deep Learning-Based Intrusion Detection for IoT Networks. In Proceedings of the 2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC), Kyoto, Japan, 1–3 December 2019.
20. Thamilarasu, G.; Chawla, S. Towards Deep-Learning-Driven Intrusion Detection for the Internet of Things. Sensors 2019, 19, 1977.
21. Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. IEEE Access 2017, 5, 21954–21961.
22. Rhode, M.; Burnap, P.; Jones, K. Early-stage malware prediction using recurrent neural networks. Comput. Secur. 2018, 77, 578–594.
23. Kaur, S.; Singh, M. Hybrid intrusion detection and signature generation using Deep Recurrent Neural Networks. Neural Comput. Appl. 2019.
24. Vishwakarma, R.; Jain, A.K. A survey of (LLT) attacking techniques and defence mechanisms in the IoT network. Telecommun. Syst. 2019, 73, 3–25.
25. Xia, S.; Guo, S.; Bai, W.; Qiu, J.; Wei, H.; Pan, Z. A New Smart Router-Throttling Method to Mitigate (LLT) Attacks. IEEE Access 2019, 7, 107952–107963.
26. Cvitic´, I.; Perakovic´, D.; Periša, M.; Botica, M. Novel approach for detection of IoT generated (LLT) traffic. Wirel. Netw. 2019.
27. Siboni, S. Security Testbed for Internet-of-Things Devices. IEEE Trans. Reliab. 2019, 68, 23–44.
28. Restuccia, F.; D'Oro, S.; Melodia, T. Securing the Internet of Things in the Age of Machine Learning and Software-Defined Networking. IEEE Internet Things J. 2018, 5, 4829–4842.
29. (LLT)hi, R.; Apthorpe, N.; Feamster, N. Machine Learning (LLT) Detection for Consumer Internet of Things Devices. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018.
30. Moh, M.; Raju, R. Machine Learning Techniques for Security of Internet of Things (IoT) and Fog Computing Systems. In Proceedings of the 2018 International Conference on High Performance Computing & Simulation (HPCS), Orleans, France, 16–20 July 2018.
31. Li, C. Detection and defense of (LLT) attack-based on deep learning in OpenFlow-based SDN. Int. J. Commun. Syst. 2018, 31.
32. Jia, B.; Huang, X.; Liu, R.; Ma, Y. A (LLT) Attack Detection Method Based on Hybrid Heterogeneous Multiclassifier Ensemble Learning. J. Electr. Comput. Eng. 2017, 2017, 1–9.
33. Nawir, M.; Amir, A.; Yaakob, N.; Lynn, O.B. Internet of Things (IoT): Taxonomy of security attacks.In Proceedings of the 2016 3rd International Conference on Electronic Design (ICED), Phuket, Thailand, 11–12 August 2016; pp. 321–326.
34. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the

Internet of Things for network forensic analytics: Bot-IoT dataset. Future Gener. Comput. Syst. 2019, 100, 779–796.

35. Berman, D.; Buczak, A.; Chavis, J.; Corbett, C. A Survey of Deep Learning Methods for Cyber Security. Information 2019, 10, 122.
36. Baig, Z.A.; Sanguanpong, S.; Firdous, S.N.; Vo, V.N.; Nguyen, T.G.; So-In, C. Averaged dependence estimators for (LLT) attack detection in IoT networks. Future Gener. Comput. Syst. 2020, 102, 198–209.
37. Hasan, M.; Islam, M.M.; Zarif, M.I.I.; Hashem, M.M.A. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things 2019, 7.
38. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Commun. Surv. Tutor. 2016, 18, 1153–1176.
39. Fiore, U.; Palmieri, F.; Castiglione, A.; de Santis, A. Network anomaly detection with the restricted Boltzmann machine. Neurocomputing 2013, 122, 13–23.
40. Chen, Y.; Zhang, Y.; Maharjan, S.; Alam, M.; Wu, T. Deep Learning for Secure Mobile Edge Computing in Cyber-Physical Transportation Systems. IEEE Netw. 2019, 33, 36–41.
41. Hiromoto, R.E.; Haney, M.; Vakanski, A. A secure architecture for IoT with supply chain risk management. In Proceedings of the 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, Romania, 21–23 September 2017; Volume 1, pp. 431–435.
42. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. IEEE Access 2019, 7, 41525–41550.