Minimax Theory and its Applications

Volume 11 (2026), No. 3s, 774 to 797

AI-Powered Safeguards for NB-IoT: An Experimental Analysis of Security Models against Botnet Attacks

R. Reshma[1], N. Mohanasundaram[2], R. Santhosh[3]

[1]Research Scholar, Department of Computer Science and Engineering, Karpagam Academy of Higher Education, Coimbatore

[2,3]Professor, Department of Computer Science and Engineering, Karpagam Academy of Higher Education, Coimbatore

Email: reshmateenu@gmail.com

Abstract

Narrowband Internet of Things (NB-IoT)has become a prominent technology in communication in both large-scale low-power IoT deployment in real-time (e.g., smart cities, health monitoring), and automation in industries (e.g.,automated guided vehicles (AGVs), AI-driven inventory systems). Although NB-IoT is efficient and widely covered, it is susceptible to changing cyber-threats such as botnet attacks that aggressively utilize resource-constrained devices and centralized network architecture. The conventional security systems, the largest part of which relies heavily on fixed rules and signature-driven identification, cannot identify advanced and dynamic attack patterns on the fly. In this regard, the purpose of this paper is to comment on the application of AI in enhancing security systems in NB-IoT networks. To explore the AI security models, the paper explores efficient algorithms such as Deep Neural Network (DNN), Convolutional Neural Network (CNN), Long Short-Term Model (LSTM), and Support Vector Machine (SVM) in learning how a network behaves, how to detect anomalies, and how to react proactively to intrusion through botnets. The machine learning and deep learning models will be compared in terms of identifying malicious traffic, affected devices, and coordinated attack patterns to ensure higher detection and resilience at reduced latency. We demonstrate that AI-based defences are much more effective in detection and resiliency than conventional ones, without perceptibly reducing the low-power and low-bandwidth characteristics of NB-IoT systems. This paper emphasizes the necessity of smart and dynamically adaptable security solutions in real time and provides insight into how secure and resilient IoT Infrastructures can be developed to address large-scale botnet attacks in the future.

Keywords: Security, Internet of Things, Narrow-Band IoT, Artificial Intelligence, Botnet Attacks, Intrusion Detection

Introduction

The NB-IoT technology has transformed the low-power wide-area networking by offering the possibility of having a very large number of devices to transmit over a long distance while using less power in the system. It has demonstrated its use in various areas, such as bringing about a smart city, automation of operations in various industrial settings, data collection in the environment, and utility management in the system [1-3]. The technology would offer the system a few amazing advantages, such as deep penetration support of the system indoors, the devices in the system have a long battery life, and the system has an efficient utilization of the power spectrum, all of which are significant in managing machine-type communication on a large scale within the system. However, the strengths observed in this area even introduce new significant security challenges.This scarcity of hardware, lightweight protocol structures, and extended lifecycle in NB IoT end devices creates some security loopholes that can be exploited by attackers. These features help NB-IoT nodes become attractive targets of botnet operators who can use them as redistributed resources in targeted malicious campaigns [4].

Once botnets become controlled over NB-IoT devices, they convert them into disruption instruments, able to cause harm both at the network and in the most important services. Hijacked devices may be used to issue distributed denial-of-service (DDoS) attacks, control telemetry and control signals in cyber-physical systems, or empty battery resources to fast-track device theft [5]. In addition to these direct impacts, the extensive and systematic breaches compromise the trust in the NB-IoT deployments and place the operators with significant remediation costs.

Following traditional protection measures, including access control lists, firmware authentication, and lightweight cryptographic tools are still the necessary element of protection. Intrusion detection systems (IDS) based on signature and rule-based detection methods are commonly investigated in the environment of IoT and NB-IoT. These systems are based on a set of attack signatures or static threshold rules to detect malicious activity. Although they work well against known threats in real-time, it is observed that they have a severe limitation in terms of identifying zero-day attacks and adaptive botnets, thus restricting their applicability to dynamic NB-IoT networks. Besides, large scales have further difficulties due to frequent manual updates and a lack of scalability [6].

To overcome the rule-based system rigidity, statistical anomaly detection methods have been suggested as lightweight alternatives [7-8]. These techniques are used to identify abnormal traffic behavior based on statistical measures of the mean, variance, entropy, or distribution of traffic volume. Despite their computational efficiency, statistical solutions are often associated with large false-positive rates, a lack of contextual awareness, and, most importantly in heterogeneous NB-IoT systems where traffic patterns can be significantly different between different applications.

Nevertheless, these options are not usually sufficient against adaptive attackers who take advantage of zero-day vulnerabilities, incorrectly configured devices, or vulnerabilities added with a compromise of the supply-chain. Furthermore, the resource limitations of NB-IoT devices are inherent, and it is not possible to deploy computationally intensive security agents to them, which is another indication of the practical limitations of device-centered protections only. These limitations have increased the effort in enhancing NB IoT security with AI-based mechanisms [9].

The techniques of artificial intelligence and machine learning can be deployed in the central hub on the cloud, in edge nodes operated by an operator, or in more sophisticated gateways to extract traffic patterns and identify anomalies that may signal botnet activity [10]. In general, many AI models can learn normal behaviours, noticeably frequency of reporting, normal packets sizes and normal signallingbehaviour, and identify slight deviations that a hard-written rule would overlook. Known malicious behaviour can be classified using supervised methods of learning, whereas unsupervised and semi-supervised methods are capable of detecting new or emerging attack tactics without the use of large labelled datasets [11-12]. Adaptive responses can also be facilitated by reinforcement learning and online learning plans which can adapt to the threat landscape.

Although AI promises something, any combination of such abilities into NB IoT security architecture must be carefully designed to consider the limitations of the ecosystem and the issues of operation. Centralized analysis can take advantage of massive compute power and massive data to train advanced models, but may add latency, and dependency on network connectivity, and raise privacy concerns [13]. Edge and gateway solutions may also offer quicker detection and localized control and minimize data transfer, but have to trade model complexity with resources. Hybrid schemes that spread intelligence both to the cloud and the edge can be used to address these trade-offs, however, they require coordination schemes, sound model update trails, and model explainability and trustworthiness strategies [14-16].

Testing AI-based safeguards is also associated with assessing trade-offs in a variety of dimensions. Detective and false-positive rates have a direct impact on operational load: a high false alarm rate may saturate the analyst team, causing alert fatigue, and missed hits will compromise security [18]. Latency is important when timely mitigation is necessary to break the ongoing botnet medical attention. The resource usage of CPU, memory, and battery should be minimized to maintain the battery life and cost-efficiency of devices to the operators. Another important consideration is scalability, as NB-IoT networks can include millions of devices spread across a variety of geographies and administrative areas [19]. The models should be less exposed to sensitive telemetry and comply with data protection principles when it is necessary, so that no one overlooks privacy and regulatory compliance.

The ability to implement NB IoT networks in real-time has opened possibilities of connectivity, but devices have also become vulnerable to the security threats of large scale, especially botnet attacks. The present work examines the ways in which NB IoT security models can be enhanced with the use of artificial intelligence to enhance resilience to this kind of threat. We consider various architectural strategies to deploy intelligence, i.e., centralized, edge-based, and hybrid, and evaluate the ways machine learning approaches can be tailored to the specific traffic pattern of constrained devices. The mitigation strategies are studied at various levels, with automated node isolation as the bottom layer and network-level throttling as the top one.

The evaluation will be informed by adversary capabilities models as part of maintaining realism. Thisincludes mass device compromise, command-and-control channels evasion, and stealthy low-rate attacker behaviors. In

this context, we examine the effect of design decisions on the robustness, operational efficiency, and privacy of the system for the user. The article introduces a systematic approach to evaluation, identifies typical AI methods that can be used in the context of NB IoT, and provides the results of the real-life experiment and simulation-based research in a variety of attack scenarios.

Next, we provide a systematic assessment framework, outline the common methods of AI and their applicability in NB IoT settings, and comment on the empirical and simulation-based findings that shed some light on the performance in a range of attack scenarios. This article then concludes with the open issues, like secure model-updating mechanisms, cross-domain coordination of incident response, andadversarial robustness of detection models.Finally, suggested with the research directions and operational recommendations to network operators, device manufacturers, and policymakers aiming to make NB IoT ecosystems resilient to botnet threats.

Literature Review

The rise in NB-IoT has heightened security issues because it is implemented in critical and high-volume applications like smart infrastructure, health care monitoring, and automation of the industry. Initial studies on NB-IoT security were mainly based on lightweight cryptography and authentication that were based on cellular communication standards [20]. These techniques were to provide data confidentiality, data integrity, and secure access to the devices and also meet the stringent data energy and bandwidth requirements of NB-IoT equipment. Nevertheless, these preventative measures do not suffice to face sophisticated threats after the compromise of the devices.

Chaabouni et al. (2019) [21] emphasized the significance of IoT security with reference to the significant cyber attacks like the 2016 Dyn attack that demonstrated how vulnerable smart devices can be when affected. Still on NB-IoT, Du Feng et al. (2024) [22] found that some of the primary security issues are data leakage, physical device tampering, and man-in-the-middle. All of these concerns still prevent confidence in NB-IoT systems and challenge their mass application.

Cryptographic Security Frameworks

The recent studies have aimed at coming up with a full cryptography solution to IoT environment. The proposed framework is the ADCS-IoT (Sonam Lata et al., (2024) [23]) that combines the encryption of data with the help of the Advanced Encryption Standard (AES) along with the application of machine learning algorithms, such as Random Forest (RF), Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) to identify anomalies. It is a hybrid solution that covers both the aspects of safe data transmission and threat detection. Zhao et al. (2024) [24] go further to establish what they call a lightweight encryption security framework, which has been developed using both symmetric and asymmetric encryption strengths. Their work illustrates the use of blockchain technology in improving NB-IoT security by increasing data credibility through decentralization and resilience of the system. This is augmented by Gao-Feng Zhan et al. (2021) [25], who present cryptographic schemes that safeguard the whole data flow between the source output and visual display in NB-IoT applications.

Signature-Based Intrusion Detection Systems

In recent times, there has been an interest in enhancing the security of IoT using sophisticated cryptographic initiatives. Similarly, Ahmed et al. (2025) [26] also advanced signature-based IoT IDS using machine learning to overcome the limitations of zero-day attacks, using pattern recognition to increase accuracy, and changing based on the constraints of the resources provided by the IoT. Their task was to enhance the processes of detecting changing threats, but literature has mentioned a set of difficulties connected to the heterogeneity of the gained data, computational complexity, and the inability to address new threats as the constant constraints of ML-IDSS in the IoT. Also, Rehman et al. (2024) [27] suggested a host-based intrusion detection system that integrated signature-based schemes with an AI-based anomaly detection system to improve cybersecurity. Although the hybrid model enhanced both flexibility and detection rates regarding new threats, the research mentioned the essential drawbacks of high computational cost, reliance on various datasets, and the inability to scale and high false alarms.

The role of signature-based detection in the security of IoT remains significant due to its great effectiveness in detecting familiar attack patterns. The researchers in Taj Uddin Sheikh et al. (2019) [28] demonstrated that the problem of detecting attack patterns, especially DDoS attacks, can be successfully addressed with optimized pattern-matching techniques even under the resource-toxic conditions of IoT environments. It is based on this

that N. Sheikh et al. (2018) [29] suggested a lightweight signature-based intrusion detection system (IDS) architecture specifically geared toward the limited processing and memory resources of IoT devices.

The newer studies have contributed towards improving the traditional signature-based detection through the incorporation of emerging technologies. Muhammad Hassan Nasir et al. (2023) [30] proposed the BTC_SIGBDS scheme that relies on blockchain technology to facilitate collaborative and trustworthy signature-based detection of botnets. Their testing on benchmark datasets, including ISOT, IoT23, and BoTIoT showed that they can successfully identify malicious traffic with minimal use of CPU and memory. Moreover, Mansoor Farooq et al. (2022) [31] investigated the relevance of signature-based detection methods to next-generation 6G-enabled IoT networks and pointed to the fact that they remain applicable despite the changing network structure.

Rule-Based Detection Methodologies

Over the last few years, the rule-based detection systems have gotten better such that they can better address new and sophisticated attack patterns. At this, Sanjay Chakraborty et al. (2023) [32] proved a rule-based deep neural network model that is capable of over 99% detection accuracy on unseen attacks with CICIDS 2017 dataset, and which has a suitable level of balance of the detection accuracy, and the false positive/negative rates. In line with this, J. B. Awotunde et al. (2021) [33] designed an intrusion detection model using deep learning to attack Industrial IoT (IIoT) systems and was founded on a hybrid rule-based classification of features. The created method achieved an accuracy of 99.0 on NSL-KDD and 98.9 on UNSW-NB15. Fal Sadikin et al. (2023) [34], proposed a hybrid intrusion detection system that combines machine learning-based anomaly detection and rule-based detection, which would be more effective in improving the security of the IoT networks that are working with ZigBee communications.

Statistical and Machine Learning-Based Anomaly Detection

The deep learning techniques have hugely empowered the statistical anomaly detection methods. The Pearson Correlation Coefficient was introduced in Mansi Bhavsar et al. (2023) [35] as PCC-CNN, which also uses CNN to enhance the correlation of features and the detection of anomalies. Their model was found to be high detection accurate in various benchmark datasets, such as NSL-KDD, CICIDS-2017, and IOTID20. On the same note, T. Sava et al. (2022) [36] designed CNN-based anomaly detecting systems, which recorded good performance on both NID and BoT-IoT data sets.

In addition to this, further advancement in the detection of anomalies has been made possible by effective feature selection strategies. Supongmen Walling et al. (2024) [37] used statistical feature diction like one-way ANOVA and Pearson correlation to increase the level of detection of NSL-KDD and UNSW-NB15. Yair Meidan et al. (2018) [38] had presented the N-BaIoT framework in earlier work, which employed deep autoencoders in learning the normal behavior of devices and anomalies in case of compromised IoT devices. Moreover, Otoum et al. (2019) [39] introduced DL -IDs that are based on the spider monkey optimization and the use of stacked deep polynomial networks to optimize the process of feature selection and improve the classification results.

Hybrid and Multi-Layered Approaches

The latest tendencies of research are strongly oriented toward the integration of various methods of intrusion detection in order to enhance accuracy and efficiency. As Hichem Sedjelmaci et al. (2017) [40] showed, a combination of anomaly-based and signature-based detection techniques can offer a high detection rate and a low number of false positives. Their game-theoretic model is especially useful in saving energy by enabling the activation of anomaly detection only when new or unfamiliar patterns of attack are probable. On the same note, J Jose et al. (2020) [41] proposed a deep learning model that combines CNNs and Long Short-Term memory (LSTM) to learn the patterns of attacks as well as the abnormal actions. More recently, S. A. R. et al. (2025) [42] introduced a combination of Deep Neural Networks and Blockchain technology (DNNs-BCT): an integrated security approach. Their architecture reached a high detection rate and reduced response time efficiency, which underscores the possibilities of hybrid architectures in dealing with the current issues of IoT security.

Specialized Applications and Emerging Trends.

The problems of security in cellular and NB-IoT environment have been particularly discussed in several studies. Miloš Savić et al. (2021) [43] therefore targeted cellular IoT applications by incorporating deep learning-based anomaly detection in the 3GPP mobile IoT architecture. They employ auto encoder models on the edge devices and on the mobile core network allowing successful identification of abnormal traffic on the various layers of the network. Going somewhat in the same direction, Pablo Salva-Garcia et al. (2018) [44]

studied the topic of the security of 5G-enabled NB-IoT systems and suggested an effective method of filtering network traffic at the multitenant cellular IoT networks to alleviate the effects of malicious traffic.

Other studies have focused on intrusion detection within the Industrial IoT (IIoT) setups. Rehab Alanazi et al. (2023) [45] presented an intrusion detection model that was tested based on the X-IIoTID dataset, and the data of the model is high detection accuracy with a very low false positive rate. Furthermore, I. Mutambik et al. (2024) [46] introduced the IoT-FIDS system, which is a flow-based intrusion detector that is based on the analysis and study of the patterns of communication behavior without referring to the pre-marked data, which is suitable in dynamic and large-scale IoT networks.

Systematic Reviews and Future Directions

Some of the survey-based studies have been instrumental in offering important insights on intrusion detection methods of systems in the IoT. In a study by Muaadh A. Alsoufi et al. (2021) [47], 26 research studies were analyzed on the topic of anomaly-based detection of intrusion in IoT through deep learning. Their results show that overall, supervised deep learning techniques have a higher detection performance than unsupervised and semi-supervised algorithms. Similarly, a separate survey conducted by Jawad Hassan et al. (2023) [48] also investigated the use of blockchain-based intrusion detection systems in IoT and the advantages of decentralised blockchain infrastructure in enhancing the security and resiliency of the system against contemporary cyber threats.

Irrespective of such developments, much of the current literature focuses on general IoT or wireless sensor networks, but does not necessarily address NB-IoT architectures. Moreover, the literature frequently compares individual machine learning or deep learning models against each other, which does not offer much comparison in terms of the results of their performance under the NB-IoT-specific conditions of low data rates, low power usage, and centralized network control. These constraints disclose the apparent gap in the research and underline the necessity of a full-fledged assessment of both conventional and AI-driven security frameworks that are specifically designed to be used in NB-IoT-based conditions.

Table 1. Comparative Analysis of Existing Security Approaches

Approach Category	Technique	Key Strengths	Limitations	Representative Works
Cryptographic Security	Lightweight encryption, authentication	Low overhead, secure communication	No attack detection capability	[23-25]
Signature-Based IDS	Rule and pattern matching	Effective for known attacks	Fails for zero-day attacks, static	[26-32]
Rule-Based Detection	Threshold and policy rules	Simple, low computation	Poor adaptability, manual tuning	[33-35]
Statistical Anomaly Detection	Entropy, deviation analysis	Lightweight, signature-free	High false positives	[36-40]
Machine Learning and its advances	SVM, Decision Trees, DNN, CNN, LSTM	Better accuracy than rules, Automatic feature learning, Captures sequential attack behaviour	Feature-dependent, limited temporal learning, computational complexity, training complexity	[36-47]

Based on the literature reviewed, it can be concluded that conventional security solutions are not flexible, whereas available AI-oriented solutions are typically unable to address constraints of NB-IoT networks comprehensively or provide comparative understanding among different learning models. The movement to the deep learning integration, the use of blockchain technology, and the use of energy-efficient frameworks to allocate resources to resource-constrained devices is the modern state-of-the-art in NB-IoT security studies. As a result, it is evident that a combined evaluation framework is required to linearly assess conventional schemes and several AI-based models, including DNN, CNN, LSTM, and SVM, to optimally improve botnet detection in NB-IoT networks.

Research Methodology

The potential solutions can be found in artificial intelligence (AI) that allows adaptive detection of threats, intelligent feature selection, and real-time classification. Resilience to botnets in NB-IoT networks can be enhanced by an AI-enhanced security architecture that, therefore, guarantees secure and sustainable operation of the critical infrastructures. Compared to the classic security measures that are frequently ill-equipped to handle the resource limitations of NB-IoT devices, AI-based approaches are able to learn dynamic tendencies in the malicious behavior, adjust to changing attack methods, and maximize the detection rates without causing extensive computational demands.

Lightweight machine learning with metaheuristic optimization, these types of architectures can achieve reasonable efficiency and robustness, and are used in large-scale applications in smart cities, industrial automation, environmental monitoring, and utility metering. Additionally, AI allows for acting in advance, detecting anomalies before turning into a coordinated botnet attack,minimizing downtime,safeguarding sensitive data, and preserving trust in IoT-enabled services. Such synergy between NB-IoT and AI is not only a solution to the current vulnerabilities, but it also forms a basis for scalable and future-ready security solutions that can sustain the dynamism of the cyber threat environment.

The figure below illustrates a tiered and integrated security strategy in an effort to defend NB-IoT networks against attacks by botnets without compromising the underlying theme of low power consumption, limited computation, and low bandwidth. Security intelligence is also spread at device, edge, and network layers instead of focusing on a centralized defense mechanism to provide timely detection of security threats, efficient mitigation, and scalable protection.

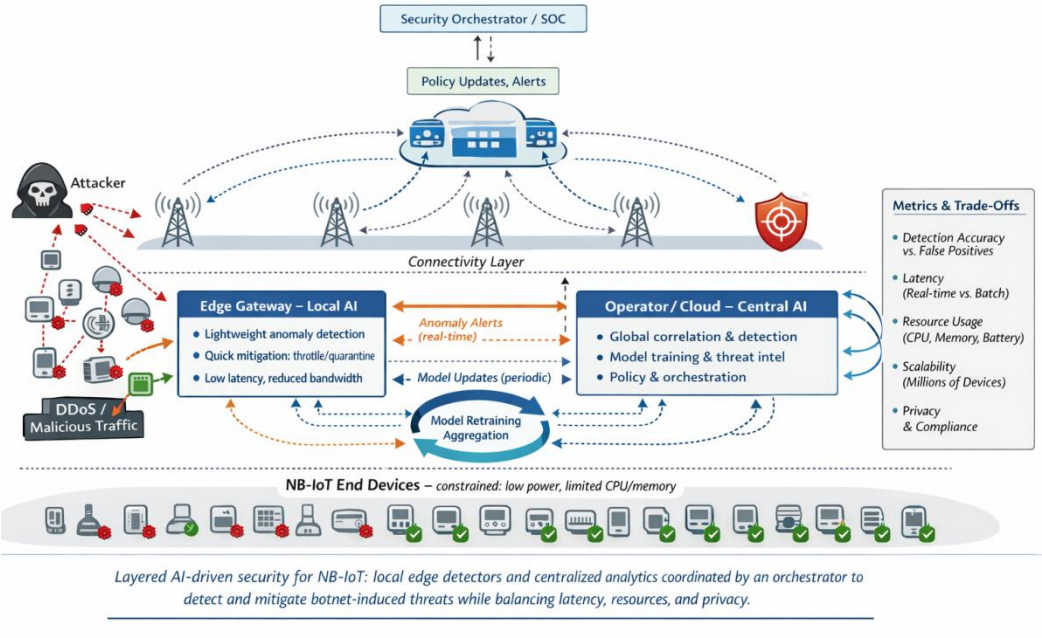


Fig. 1: AI-Enhanced NB-IoT Security Architecture Against Botnets

On the bottom, there aremany NB-IoT end devices, which are sensors, smart meters, and actuators, but which have a very strict energy and memory constraint. These devices do not carry sophisticated security processing in them and rather concentrate on regular data transfer, and simple telemetry and behavioral signals are sent out over narrowband connections. These devices are very appealing targets to botnet infections because of their confined nature, hence the need to have external monitoring.

NB-IoT base stations are located above the device layer as points of aggregation of uplink and downlink communication. Group traffic between groups of devices passes through these stations to edge gateways and the backhaul network of the operator. The layer of connectivity is an important observation point where maladaptive patterns of communication could initially be observed without any direct effects on device performance.

The introduction of local intelligence occurs at the edge gateway, where the lightweight AI models are deployed to detect the anomalies in real-time. The edge layer in the given model is constantly examining received telemetry to determine anomalies in expected traffic behavior, including bursts of traffic, multiple connection attempts, or coherent action on multiple devices.The edge gateway even takes immediate mitigation steps, including traffic shaping, isolating compromised devices, or blocking malicious traffic temporarily, whenever suspicious behavior is noticed in the NB-IoT network. Such a localized response will reduce the latency, minimize the backhaul traffic that is not necessary, and constrain the propagation of the botnet activity at an early stage.

The centralized AI system deployed in the operator or cloud environment at the network core is used to further analyze the collected aggregated data of several edge gateways and base stations. This layer is concerned with correlation globally, long term pattern, and coordinated attacks, which are detected in a large geographical area. At this level, more sophisticated models are trained and updated with denser datasets, enabling the system to identify more botnet evasion tactics that cannot be observed at the edge all on their own. Policies and intelligence about threats made in this layer are sent back to the edge layer to ensure that local detection models are updated.

The security orchestration layer is in charge of both edge and cloud intelligence, and it is commonly collaborative with a Security Operations Center (SOC). This element controls notifications, implements security rules, and organizes response measures on the level of the whole NB-IoT ecosystem. It is able to send control instructions, like quarantine of the device, rollback of the firmware, or rollout of security patches, to have a uniform and controlled mitigation. The bi-directional communication between the orchestrator, as shown in the figure, and the rest of the layers allows the orchestrator to provide feedback continuously, fine-tune its detection strategies, and adjust to new threats quickly.

The given architecture in this research work takes specific consideration of botnet behavior monitoring in real-time. Hacked systems can be used to create malicious traffic or be used in distributed denial-of-service attacks. The edge Anomaly alerts detect these attacker-driven flows and cloud correlation analysis with the objective of targeted intervention without interfering with the normal work of the network.

This multifaceted AI-enabled security architecture, in general, is a balance between real-time responsiveness and worldwide intelligence. It can ensure more efficient botnet mitigation by integrating local-level detection, centrally based analytics, and orchestrated coordination without compromising the operational limitations of NB-IoT networks. The characteristics like scalability, low latency, efficient use of resources, and privacy-conscious data processing make the design appropriate in large-scale, real-world deployments of NB-IoT.

In the context of this paper, a research methodology is described to create an experimental approach to assess the effectiveness of artificial intelligence-based models in security to detect and mitigate botnet threats in the context of the NB-IoT network. The research methodology, which is offered in the given study, will provide a comparison of the traditional security approaches with artificial intelligence ones, taking into account the operational limitations on NB-IoT networks.

Research Design

The study takes an experimental and comparative research approach. There are various AI-based models, such as DNN, CNN, LSTM, and SVM, in this study, which are chosen to apply and test their potential in identifying botnet-related attacks. They are also compared to standard rule-based and signature-based detection mechanisms in order to point out the better results in accuracy, adaptability, and response time.

Data Collection and Dataset Selection

Publicly available benchmark datasets with both normal and malicious network traffic, like NBaIoT datasets, are used to simulate a realistic NB-IoT attack. These data cover botnet behaviors like Distributed Denial of Service (DDoS), command-and-control communication, and bursts of abnormal traffic. The datasets are chosen in such a way that they are diverse in terms of the type of attack as well as the pattern of traffic that applies to the IoT and NB-IoT context.

Data Preprocessing

Before model training, the data is processed with some steps that enhance the quality of learning:

- Removal of irrelevant and duplicate records

- Handling of missing and noisy data
- Normalization and scaling of feature values
- Label encoding for supervised learning

The processed data are then separated into training, validation, and testing data in order to guarantee the model is evaluated in an unbiased manner.

Feature Extraction and Representation

Major traffic and behavioural characteristics including the rate of packets, session length, the use of the protocols, the amount of traffic, and the characteristics of the time flows are mined. In the case of traditional and machine learning model namely SVM, the handcrafted features are statistical and flow-based features. In the case of deep learning models (DNN, CNN and LSTM), feature vectors are fed with minimal processing so that they can automatically learn features.

Model Implementation

The AI models that are used in our system have the following purposes:

- SVM: This is used as a control model to help us to draw the line between normal and possibly malicious behavior.
- DNN: This model is especially helpful in revealing faint patterns that would otherwise remain undetected due to nonlinear relationships that are more often than not complicated to discover.
- CNN: The given model is concerned with the spatial correlations, and we get an opportunity to examine the relationships between various elements of traffic.
- LSTM: LSTM is used to process sequential data, and is particularly effective in following temporal dependencies; i.e., coordinated botnet behavior.

In this research methodology, wetrainall the selected models through the method of supervised learning, and the parameters are optimized to achieve the best possible performance.

Training and Validation Strategy

The training dataset is used to train the models, and the cross-validation techniques are used to test the models so as to minimize overfitting. Hyperparameter optimization is done to obtain the best performance but with low computational cost, which can be accommodated in NB-IoT environments.

Performance Evaluation Metrics

The performance of all models is evaluated closely according to the well-established intrusion detection metrics that reflect the accuracy and feasible performance. Accuracy indicates the general correctness of the system predictions, whereas precision indicates the capability of its ability to reduce the number of false alarms by accurately detecting malicious activity. On the other hand, recall keeps ensuring that real threats are not left out, and the F1-score gives a balanced score by integrating the precision and recall. In addition to these statistical measures, detection latency is also a factor that is looked into since it shows the rate at which a system can adequately react to a potential intrusion in real time. Collectively, these measures give a full picture of the detection capability, reliability, and responsiveness of the models so that the assessment process can be considered rigorous and relevant to the real-world security contexts.

Comparative Analysis

An extensive comparative analysis is conducted in detail to compare the strengths and drawbacks of various ways of securing NB-IoT networks. The paper compares AI-based models to traditional security machinery and shows the way intelligent algorithms can better adapt to new threats than rule-based systems. It also looks at the distinction between machine learning and deep learning methods, where machine learning offers ease and effectiveness, whereas deep learning offers more power in terms of capturing essence and complex patterns. This discussion in our research examines the trade-offs between detection accuracy and computational efficiency. Moreover, the work observed that highly accurate models can be resource-intensive, or lightweight models can be fast and scalable. Cumulatively, these comparisons can offer important information on the applicability of each model to the real-world implementation in NB-IoTs to balance the performance, reliability, and practicality.

Deployment Considerations

To meet NB-IoT requirements, the paper assesses the viability of implementing AI models at edge gateways and centralized network servers, as opposed to implementing them at the constrained end devices. This will provide a low power-consumption, low bandwidth, and effective threat response mechanism.The presented methodology allows conducting a systematic assessment of AI-driven security systems and proving that they can help improve the quality of botnet detection, minimize the response time, and increase the resilience of the network without violating the low-power and low-bandwidth demands of NB-IoT.The following figure (Fig. 2) illustrates the full workflow and security architecture that will identify botnet attacks in NB-IoT networks based on artificial intelligence. It describes the stratified architecture of the system with the NB-IoT end devices on the lower level, the communication network and the edge gateways in the middle, and the centralized cloud-based security analysis on the top.

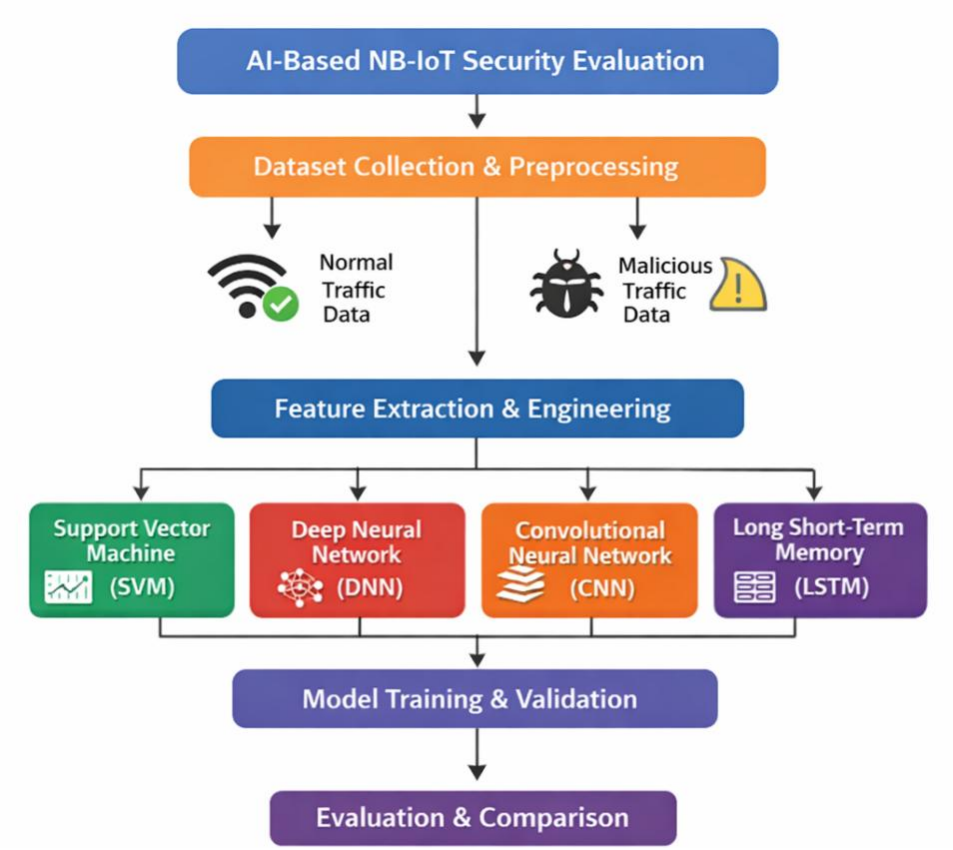


Fig. 2: Overview of the Research Methodology

The illustration of Fig.2 shows that the network traffic is followed, processed, and analyzed with the help of machine learning and deep learning algorithms to identify abnormal and malicious activities. It is then performed with evaluation and comparison that can uphold quick reaction and yet preserve the low-power and low-bandwidth characteristics of NB-IoT networks.

Experimental Analysis

This experimental study explores machine learning and deep learning schemes to identify botnet activity in NB IoT networks and both detection and trade-offs in their deployment. Here, one will discuss the data set chosen in the research, the process of cleaning and data preparation, feature engineering and extraction methods, and the developed, trained AI models and their evaluation.

Simulation Dataset Description

Figure 3 is sampled using the N-BaIoT dataset [49], and machine learning and deep-learning tools were evaluated based on detecting botnet activity in NB-IoT networks. The experiments in this paper utilize the N-BaIoT dataset, which is a dedicated set of network traces and flow logs that are acquired in the controlled benign and botnet-infected settings of real NB-IoT devices. The NBaIoT dataset used here consists of labeled

data representing benign NB-IoT traffic and affected botnet attacks with the feature sets of volumetric floods, fast and low-and-slow scans, and periodic C2 beacons.This was gathered in a controlled testbed and supplemented with publicly available IoT captures by the researchers in [49]. The dataset used for experimental analysis can offer real-world telemetry with NB-IoT timing, payload sizes, and protocol utilisation. The dataset instances of common botnet traffic like scanning, command-and-control communications, DDoS-style flooding, and data-exfiltration attempts, as well as regular telemetry traffic, retain the sparse, low-rate, duty-cycled communications patterns that characterize NB-IoT traffic.

We used both the given flow-level metrics, such as the number of bytes and packets per flow in each direction, flow duration, aggregates of packet sizes, statistics of inter-packet timing,etc., and the raw PCAPs, where we need to extract additional features. Individual device baselines for the evaluation were constructed using device identifiers and timestamps, and temporally aware validation is done to reduce overfitting to time-dependent patterns. The documentation of the dataset regarding the experimental setup and the attack generation is applied to situate the results and consider the generalization constraints related to a particular device model and the conditions within the testbed. This combination of realistic NB-IoT behavior, marked attack scenarios, and the availability of raw and preprocessed data makes N-BaIoT an appropriate benchmark when testing the detection models have to work under the restrictive traffic and privacy-sensitive conditions of NB-IoT.

	G	H	I	J	K	L	M	N	
_L3_	MI_dir_L1_	MI_dir_L1_	MI_dir_L1_	MI_dir_L0.1	MI_dir_L0.1	MI_dir_L0.1	MI_dir_L0.(	MI_dir_L0.(	MI_d
0	1	98	0	1	98	0	1	98	
0	1.492583	98	3.64E-12	1.93164	98	1.82E-12	1.992944	98	
7464	2.294102	84.05119	251.7926	2.904273	86.98175	231.1822	2.990102	87.29803	227
2693	3.280499	80.98727	196.4467	3.902546	83.65527	204.0614	3.989925	83.96512	204
2248	4.28049	79.35492	159.2943	4.902545	81.68583	177.5746	4.989924	81.96807	179
3247	5.280442	78.34081	133.5295	5.902539	80.38371	155.8026	5.989924	80.63783	15
0546	431.1896	74.06948	1.935537	613.8185	74.0951	2.65911	637.318	74.09948	2.7
3863	431.9115	74.06932	1.931067	614.7789	74.09494	2.6548	638.3139	74.09933	2.77
7187	431.7834	74.06916	1.926606	615.6182	74.09479	2.650502	639.2972	74.09917	2.76
0562	432.6281	74.069	1.922163	616.596	74.09463	2.646218	640.2949	74.09902	2.76
3956	432.7719	74.06884	1.917733	617.4739	74.09448	2.641947	641.2822	74.09886	2.75
7306	431.5586	74.06868	1.9133	618.1574	74.09433	2.637687	642.2493	74.09871	2.75
0715	432.556	74.06853	1.908888	619.157	74.09418	2.633441	643.2493	74.09856	2.75
1181	433.5536	74.06837	1.904496	620.1566	74.09402	2.629209	644.2493	74.0984	2.74
7704	434.5505	74.06821	1.900124	621.1562	74.09387	2.624991	645.2492	74.09825	2.74
1284	435.55	74.06805	1.895772	622.1561	74.09372	2.620786	646.2492	74.0981	2.73
4918	436.5463	74.0679	1.89144	623.1556	74.09357	2.616594	647.2492	74.09795	2.73
3484	434.7028	74.06774	1.887099	623.7485	74.09342	2.612413	648.2069	74.0978	2.72
5209	435.3627	74.06759	1.882775	624.6997	74.09327	2.608245	649.2018	74.09765	2.72
5743	436.1444	74.06743	1.878469	625.6684	74.09312	2.60409	650.1985	74.0975	2.72
9409	436.1732	74.06728	1.874172	626.5289	74.09297	2.599948	651.184	74.09735	2.71
8313	437.1713	74.06712	1.869896	627.5286	74.09283	2.595818	652.184	74.0972	2.71
5906	438.1698	74.06697	1.865639	628.5284	74.09268	2.591702	653.184	74.09705	2.70

Fig. 3 Sample Image of NB-aIoT Dataset

DataPreprocessing and Feature Engineering

The data preparation and preprocessing pipeline used in the N-BaIoT dataset helps this research in training and evaluating the selected SVM, CNN, DNN, and LSTM models. It could then produce succinct, significant, and time-sensitive feature representations. The process, as illustrated in Figure 4 starts by parsing raw PCAP files and assembling them back to device-specific, chronologically ordered flow records. Idle timeouts NB-IoT-friendly (usually 300900 seconds long) idle timeouts are used to prevent the division of naturally long sleep cycles into several arbitrary flows. Packet-level features such as timestamps, packet sizes, direction, protocol, and source/destination information are remembered, with inter-packet arrival times and size distributions computed. Payload information in the dataset is either discarded or summarized with a statistical measure such as entropy, byte-frequency histograms, or cryptographic hash in this phase.

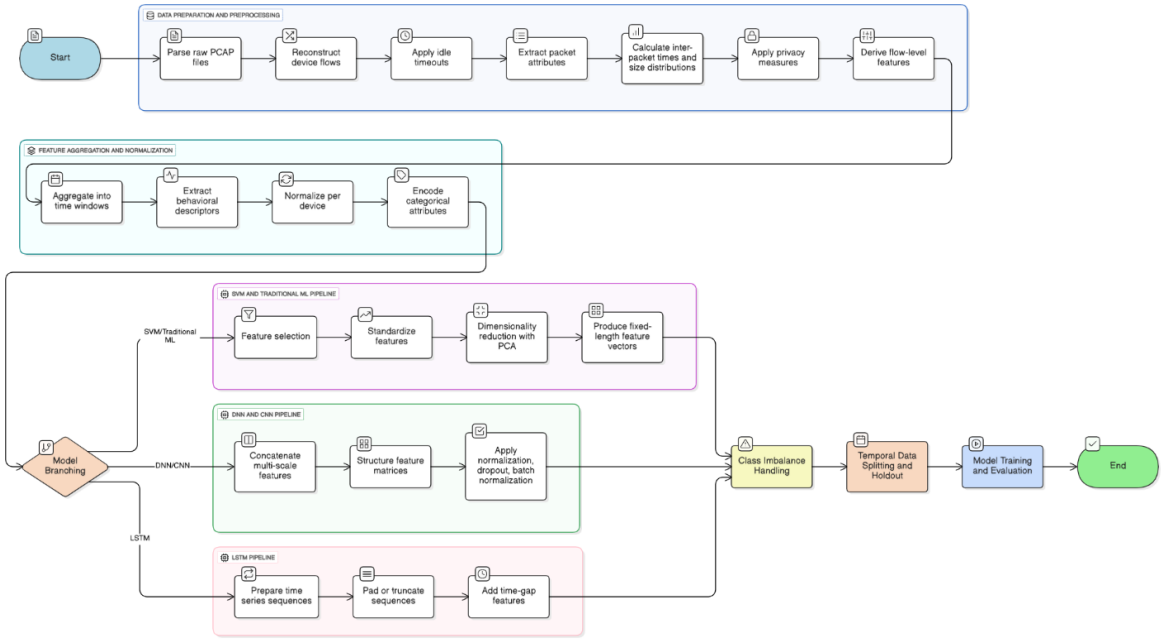


Fig. 4 Data Preprocessing and Feature Engineering for Model Training

Based on these flows, basic flow-level characteristics are obtained, such as flow duration, the total number of packets and bytes, the uplink/downlink ratios, the packet-size distributions, and inter-arrival time distributions. They are then summed into device-level windows in multiple time resolutions (e.g., 30 seconds, 5 minutes, and 1 hour).Window-based descriptors capture behavioral characteristics such as the number of flows and short flows, duty-cycle indicators (active time fraction, average active and idle periods), burstiness measures, counts of distinct communication peers, and directional timing asymmetries. To reduce variability across devices, per-device normalization (using z-scores or min–max scaling based on benign reference windows) is applied, along with appropriate encoding for categorical attributes such as device type or protocol.

For SVMs and other traditional machine learning methods, the pipeline produces a fixed-length, low-dimensional feature vector for each time window. Feature selection is performed using techniques such as mutual information or tree-based importance scores, followed by standardization and a dimensionality reduction algorithm named Principal Component Analysis (PCA) to control model complexity. In the case of DNNs and CNNs operating on tabular or time-aggregated inputs, multi-scale window features can be concatenated, or recent sequences of per-flow features can be structured into matrices (e.g., N recent flows × feature dimensions) to enable CNN kernels to learn local temporal and feature correlations. These models benefit from careful normalization, dropout, and batch normalization to improve generalization.

LSTM models are designed to process ordered sequences and therefore receive per-device time series of flow-level or window-level feature vectors. Sequence padding, truncation, or sliding windows are used to handle variable-length data. At the same time, explicit time-gap or elapsed-time features help the LSTM learn patterns across sparse NB-IoT transmissions. Across all model types, class imbalance is addressed using temporally aware resampling strategies, class weighting, or focal loss functions. Evaluation protocols maintain strict temporal separation between training and testing data and can include per-device holdout strategies to assess generalization to unseen devices.

At the end of this phase, a smaller set of features has been packaged that could build upon lightweight counters. It is simply exponentially weighted moving averages of packet rates and plain duty-cycle proxies, where edge or resource-constrained deployment is desired. Latency of inference and computational complexity are also profiled to make sure that the operational model of choice is realistic in the real-world NB-IoT.

Model Configurations and Training Environment

The experimental design was made to offer an understandable and reproducible framework of analyzing AI-based security models in NB-IoT networks. DNN, CNN, LSTM, SVM algorithms were applied, each of them was chosen due to the possibility to represent various aspects of traffic behavior. The experimental data was represented by the simulated NB-IoT traffic traces, well balanced between the benign traffic and the botnet-

based malicious traffic. Whereas both DNN and CNN were applied to acquire patterns of complex features, LSTM was utilized to acquire patterns of time-varying sequences, and SVM offered an effective baseline to classification.

The training scenario in this research was structured in such a way as to recreate realistic NB-IoT traffic with normal and malicious flows to provide a balanced learning environment. The preprocessing activities in this work, that includes normalization, feature selection, and dimensionality reduction, were performed before feeding data to the models to enhance efficiency and point out key security indicators. The systematically tuned hyperparameters were the learning rate, the batch size, and the number of epochs so that the detection accuracy was high and maintenance latency was low.

Experiments were run in an environment that is enabled with a GPU, and it has adequate memory and processing power to handle deep learning workloads. It was built on Python and Python libraries like TensorFlow, Keras, and Scikit-learn to provide flexibility and reproducibility. Random seeds were fixed to ensure consistency, and cross-validation was used among datasets. This setup provided a strong basis on which the performance of algorithms could be compared and emphasized the flexibility of AI models in identifying and preventing the threat of botnets in NB-IoT systems.

Mathematical Formulation of Each Model

Given a labeled training set of size N, and each individual data point is expressed as a pair$\{(a_x, b_x)\}_{x=1}^{d}$.In this case, the variablea_x represents the input feature vector, and $b_x$represents the class label. In the case of a binary classification problem, the class labels are in the set of only two values:$\{-1, +1\}$, and represent the two possible classes. The most popular algorithm, Principal Component Analysis (PCA) has been utilized to eliminate redundancy and correlation between features, and transform the original feature space into a smaller subspace, A_x^P:

$$A_x^P = E^T * A_X \tag{1}$$

where E holds the eigenvectors of the covariance matrix, which are the largest eigenvalues. Subsequently, the standardization of feature scaling is undertaken to guarantee that the contribution of all the input features in the training of the model is equal, and to avoid the chance that features with large scale values will dominate in the learning process. All the features are standardized with the standardization formula such that each feature has a zero mean and unit variance. It is expressed as follows,

$$\tilde{a}_{xy} = \frac{\mu_y - a_{xy}}{\sigma_y} \tag{2}$$

where the mean μ_y and standard deviation σ_y are computed for each feature $y$across the training dataset. This normalization process stabilizes the distribution of input values and significantly improves the numerical behavior of gradient-based optimization algorithms. As a result, the learning process becomes more stable and converges faster, particularly when training deep learning models such as CNNs and LSTMs, which are sensitive to variations in feature scale.

The standardized NB-IoT traffic characteristics are converted into a three-dimensional service of the format $A \in \mathbb{R}^{X,P,1}$where X is the overall count of traffic samples and P is the count of principal components that have been inferred following the implementation of PCA. This conversion is important in making the convolutional and recurrent layers process the data effectively. The dimension P is considered as a sequence length, where the models could study structured relationships among the reduced feature space instead of considering features as independent inputs. The last dimension is associated with the single input channel, which is suitable in the context of NB-IoT traffic data because the features that are extracted are a single stream of network behavior, and not multiple heterogeneous sources of signals. Such a reshaping strategy enables the CNN to use a one-dimensional convolution operation on sequences of features to learn local feature dependency, whereas the LSTM uses the identical arrangement to learn sequential dependencies and time dependencies encoded in the traffic representation. As a result, the organizational form of this data improves the capability of neural networks to acquire discriminative patterns that relate to the activity of botnets with a guarantee of computational efficiency and scalability to the NB-IoT conditions.

SVM

In the framework of NB-IoT security, the SVM model will be trained to develop discriminative separation between harmless traffic and malicious botnet traffic. Packet size distribution, connection frequency, device

activity patterns, and temporal correlations are features derived based on NB-IoT communication logs. The SVM can be used to differentiate between sophisticated attack patterns and normal traffic by mapping such features into a higher-dimensional space with the help of kernel functions.

In general, SVM is an algorithm that is trained to learn a margin between classes, and the nearest training points (support vectors) make up the decision boundary. The parameters of an SVM model include loss/regularization parameter (C), kernel family (linear, polynomial or RBF), and kernel parameters (degree, gamma) whose values are based on the primal/dual optimization models and are optimized by using convex optimization. In this case, the SVM can be formulated as a supervised learning model that is aimed at the classification of NB-IoT traffic as benign and malicious. SVM aims to establish a fine decision line that separates the two classes with the largest possible margin; however, it will tolerate a certain level of errors in classification, where one cannot be assured that the two classes can be separated perfectly. To do so SVM develops the following optimization problem:

The aim is to optimise a cost expression comprising two parts. The first part performs the maximization of the margin that leads to the reduction of the squared norm of the weight vector $\|w\|^2$, which varies the width of the separating margin. Next, the misclassification penalty, denoted by slack variables ζ_x. Slack variables, $\zeta_x$give some flexibility by letting some data points break the strict margin constraints, which is especially helpful when the data is noisy or non-linearly separable.

The optimization issue is then formulated to balance between maximizing the margin and minimizing the classification errors.Mathematically, the objective function is expressed as:

$$\min_{\zeta,w,b} \frac{1}{2}\|w\|^2 + R * \sum_{x=1}^{d} \zeta_x \quad (3)$$

where Rrepresentsa regularization parameter that helps balance the trade-off betweenmaximizing

the margin and minimizing classification errors.

The given constraints in the mathematical modelling ensure that each training sample is either correctly classified or lies within an acceptable margin of error. Specifically, each data point must satisfy $b_x(w \cdot \phi(a_x) + \beta) \geq 1 - \zeta_x, for\ all\ \zeta_x \geq 0$ (4)

where$\phi(a_x)$ in the equation (4), the mapping of the input information to a higher dimension feature space, and β is called the bias term. The slack variables ζ_x of this design are positive, and the violation is punished rather than reversed.

To successfully handle a non-linear boundary between the classes, SVM classifiers utilize kernel functions, which allow calculating inner products in a high-dimensional feature space without the need to undergo the transformation, which is done by applying a “kernel trick.” Among the already extant kernel functions, the RBF Kernel function happens to be one of the most commonly used kernel functions, with a mathematical definition as follows:

$$K(a_x, a_y) = exp^{(-\delta * ||a_x - a_y||^2)} \quad (5)$$

In this case, δ is a parameter of the kernel that regulates the impact of the separate training samples. The larger the value of δ, the more complex the decision boundaries will be, and the smaller, the smoother the decision boundaries will be. All in all, this formulation allows SVM to build powerful and adaptable classifiers that can perform both linear and nonlinear classification with ease. The SVM is effective in distinguishing normal and malicious NB-IoT traffic patterns, and is used as a reference model to compare deep learning methods.

CNN

The convolutional neural network that was created during this paper is specific in that it learns significant patterns based on features of the processed traffic to identify botnet-related anomalies in NB-IoT network traffic. Once the PCA algorithm has been performed, the input of each traffic sample is organized as an $(N^P, 1)$, -dimensional input-tensor, with the variable N^P representing the number of retained principal components following pre-processing, and the single channel representing the normalized feature sequence. This architecture is made up of several one-dimensional convolutional layers with more and more filters (64, 128, and 256) and enables the model to learn the finer, more abstract feature representations progressively. Each convolutional layer uses a kernel size of 1 so the network concentrates on feature-level changes and interactions as opposed to

larger spatial dependencies, which is also supported well in structured NB-IoT traffic data. Each of these convolutional blocks is followed by MaxPooling1D layers of pool size 1 to preserve the feature dimensionality and increase the dominance of feature activations, as well as to minimize superfluous computational costs. The trained feature maps are subsequently subjected to flattening and subsequently input into the dense layer having 128 neurons with ReLU as the activation function. This is useful in the non-linear relationship modelling of the features. The final result is the output of a single neuron with a sigmoid activation function, which also gives the probability scores on the possibility of the presence or absence of a botnet attack on the NB-IoT traffic pattern.

Generally, the essential functioning of the CNN is the one-dimensional convolution. It assists in allowing the model to inquire meaningful local patterns of the input feature sequence. The output feature map at each position k is calculated by taking a weighted sum over a local receptive field by each convolutional filter f before a non-linear activation function. Mathematically, this operation in the proposed model can be represented as:

$$C_f(k) = g\left(\sum_{i=1}^{k} w_i^f * a_{(k+i-1)} + \beta_f\right) \quad (6)$$

The variablew_i^f in the equation (6) denotes the convolutional weight associated with the i^{th} position of the f^{th} filter. Whereas, the variableβ_f is the corresponding bias term, and$k$represents the kernel size. The activation function $g(.)$ used in this study is the functional formula of ReLU, defined as

$$g(z) = max(0, z) \quad (7)$$

ReLU brings a non-linearity to the model and avoids the problem of vanishing gradients, thus allowing the model to converge much more quickly and learn complex patterns that occur in NB-IoT botnet traffic.

The last classification is performed by using a sigmoid-activated neuron of the output after the hierarchical feature extraction process. The sigmoid function uses the output of the network to a value between 0 and 1, which can be seen as the chances of a certain NB-IoT traffic flow being malicious. It is estimated that the prediction probability, $\hat{y}$as follows:

$$\hat{y} = \varphi(x) = \frac{1}{1+e^{-x}} \quad (8)$$

In this case, x is the weighted aggregate of inputs to the neuron where the output is produced. This probabilistic model $\varphi(x)$is especially appropriate in binary classification tasks since it can be used to perform decisive comparisons between regular and botnet-generated traffic, and this model allows threshold-based decisions to be made during the evaluation process.

In order to maximize the CNN model to detect binary botnets, the binary cross-entropy (BCE) loss is used. The measure of loss is based on the difference between the actual and the probabilities given by the model. It can be defined as:

$$\mathscr{L}^e = -\frac{1}{n}\sum_{j=1}^{n}\left(\ell_j * log\left(\widehat{\ell_j}\right) + \left(1-\ell_j\right) * log\left(\widehat{\ell_j}\right)\right) \quad (9)$$

In the above equation, n is a total number of training samples, ℓ_j is a true label of the j-th sample, and $\widehat{\ell_j}$is the predicted probability. A binary cross-entropy is increased when the model is sure, and the penalty is substantial when it is incorrect. This influences the level of accuracy with which the model makes probabilistic estimates. The suggested punishment is sufficient to be used in NB-IoT botnet pattern identification issues.

In general, the offered CNN architecture offers a good trade-off between the detection capacity and the computational efficiency. CNNs can capture fine-grained patterns that more often than not would be overlooked by conventional machine learning models, since the emphasis is placed on local dependencies of the features in the traffic. This is especially useful in determining behaviors associated with botnets, including coordinated packet transfers, recurring periods of communication, and coordinated attack signatures. Due to that fact, CNN-based models are more sensitive to the minor and shifting attack patterns, and hence, they can be successfully used to detect botnets in NB-IoT settings where the characteristics of traffic are lightweight, periodical, and highly organized.

LSTM

NB-IoT traffic has a temporal nature, especially when there exists a botnet attack, when the compromised devices usually communicate with command-and-control servers at regular intervals or in a sequence. These types of attack behavior are impossible to capture using models that take traffic samples as independent observations. The LSTM networks would thus be optimal in detecting NB-IoT botnets, because they are uniquely tailored to acquire long-term dependencies and temporal correlations in the sequential information. LSTM models are able to remember relevant information from the history and forget irrelevant patterns selectively by preserving an internal memory state, which is useful in modeling attack behavior based on time in NB-IoT systems.

This input sequenceis then processed by an LSTM model after feature extraction is done bya PCA algorithm and scaling. Therefore, it cancapture the temporal dependencies in botnet traffic effectively. The computations on LSTM model at time step$t$have beenexplained as follows:

Each of the time steps t has a set of computations performed by the LSTM cell based on gated mechanisms.

The forget gate determines which information from the previous cell state c_{t-1} should be retained:

$$\Gamma_t = \sigma(W_\Gamma * a_t + M_\Gamma * h_{t-1} + \beta_\Gamma) \quad (10)$$

The input gate controls the amount of new information that will be written into the cell state:

$$I_t = \sigma(W_I * a_t + M_I * h_{t-1} + \beta_I) \quad (11)$$

Simultaneously, a candidate memory vector is computed to generate potential updates for the cell state:

$$\widetilde{m}_t = tanh(W_m * a_t + M_m * h_{t-1} + \beta_m) \quad (12)$$

The cell state is then updated by combining the retained information and the candidate memory using element-wise multiplication:

$$c_t = \Gamma_t \circledast c_{t-1} + I_t \circledast \widetilde{m}_t \quad (13)$$

Next, the output gate decides which portion of the cell state contributes to the hidden state:

$$O_t = \sigma(W_O * a_t + M_O * h_{t-1} + \beta_O) \quad (14)$$

Finally, the hidden state representing the output of the LSTM at the time step t is given by:

$$h_t = O_t \circledast tanh(c_t) \quad (15)$$

Here, $\sigma(.)$ denotes the sigmoid activation function, $tanh(.)$is the hyperbolic tangent, $\circledast$ represents element-wise multiplication,a_t is the input at time t, h_{t-1} is the previous hidden state, and W, M, β are trainable weight matrices and bias vectors.

After processing the entire input sequence, the hidden state at the final time step h_T captures the accumulated temporal information. This state is passed through a fully connected layer with a sigmoid activation to generate the probability of a traffic flow being malicious:

$$\hat{y} = \sigma(W_d * h_T + \beta_d) \quad (16)$$

The output $\hat{y}$ lies between 0 and 1 and represents the likelihood of the sequence being associated with a botnet attack.

For binary classification, the binary cross-entropy (BCE) loss issimilar to CNN. To update the network parameters, the Adam optimizer is applied. Adam combines adaptive learning rates and momentum to achieve stable and efficient convergence:

$$\theta_{i+1} = \theta_i - (\eta * \frac{\widehat{X_i}}{\sqrt{\widehat{v_i}} - \varepsilon}) \quad (17)$$

Here, θ_i represents the model parameters at iteration i, η is the learning rate, $\hat{\chi}_1$ and $\hat{v}_1$ are bias-corrected estimates of the first and second moments of the gradients, and ε is a small constant to avoid division by zero. Mini-batch gradient descent with dropout is employed to prevent overfitting and ensure robust generalization across diverse NB-IoT botnet traffic patterns.

DNN

This work designed a DNN with a total of six layers to effectively learn hierarchical and nonlinear representations of NB-IoT network traffic for botnet attack detection. The architecture consists of an input layer that receives normalized NB-IoT traffic features, followed by four fully connected (dense) hidden layers and a final output layer responsible for classification. The incremental growth of the neurons in the hidden layers, which are established as 16, 32, 64 and 128 units respectively, enables the network to equally expand its representational capacity, where low-level traffic patterns can be converted to high-level discriminative features. This expansion plan in layers provides effective learning without sudden changes in model complexity that may result in a case of overfitting.

The input layer is a point of connection between the processed NB-IoT traffic characteristics and the learning architecture. Minmax scaling is used to normalize all the input features so that they can be trained numerically and contribute equally. The normalized input vector is sent to the first dense layer, which is the first stage of feature abstraction. It contains 16 neurons, and this layer represents some basic relationships, like basic packet-level statistics and signaling behavior. ReLU activation functions, like CNN at that time is used to add nonlinearity as well as to alleviate the vanishing gradient issue during backpropagation.

The third and fourth thick layers with 64 and 32 neurons, respectively, are where the more complicated feature interrelationships are learned. Such layers extract intermediate representations like correlations among packet timing, burst patterns of traffic and protocol-level behaviors, which are typical of botnet activities. The gradual growth of the number of neurons in the model enables its decision space to grow a step at a time and permits greater abstraction in early stages without other parameters growing exponentially. In these layers, the same ReLU activation is employed to guarantee the flow of gradients and faster convergence. There is dropout regularization after each dense layer to decrease co-adaptation of neurons and to increase generalization, which is especially relevant when faced with imbalanced or noisy NB-IoT traffic data.

The 128 neurons in the fourth thick layer is the high-level feature integration layer. This is the step where the network summarizes the representations learned at the time of the prior layers to differentiate between subtle and sophisticated botnet activities and real NB-IoT traffic. This layer is essential in the capturing of complicated nonlinear correlations that can occur as a result of coordinated botnet activities, command and control communication structures, as well as covert attack techniques. A sort of Batch normalization is applied to stabilize the learning process by decreasing the internal covariate shift, which enhances efficacy and hardiness of training.

The single neuron with the sigmoid function is found in the output layer to facilitate the classification of benign and botnet traffic. The sigmoid is a probability calculator that gets the network output and maps it to the range [0, 1], which allows making a decision based on a threshold. Binary cross-entropy loss is used to train the model, and it is used to estimate the difference between the predicted probabilities and the ground-truth labels. Adam optimizer in the proposed DNN changes the learning rates in an adaptive fashion so that convergence across all layers is guaranteed. It is commonly high-dimensional, sparse, and non-stationary statistical. Adam is a combination of Momentum and RMSProp that also keeps both first-order (mean) and second-order (variance) gradient averages in a decaying way that can adaptively adjust the learning rate of each parameter. Optimizer changes network parameters (following equations):

$$m_j = \partial_1 * m_{j-1} + (1 - \partial_1) * \boldsymbol{g}_j \quad (18)$$

$$v_j = \partial_2 * v_{j-1} + (1 - \partial_2) * \boldsymbol{g}_j^{\,2} \quad (19)$$

where $\boldsymbol{g}_j$ is the gradient variables at iteration j. The variables∂_1 and ∂_2 are decay rates that are controlling momentum and variance accumulation. With this under consideration, bias-corrected estimates are further computed as:

$$\widehat{m_j} = \frac{m_j}{1-\partial_1} \text{ and } \hat{v}_j = \frac{v_j}{1-\partial_2} \quad (20)$$

The final parameter update rule uses equation 17 for the above estimates by replacing $\hat{\chi}_t$ with $\widehat{m}_1$ in it for all iterationsi.

The mini-batch gradient descent algorithm applied with a sufficiently chosen batch size and early stopping using the validation loss are used to implement the model training phase in order to reduce overfitting. The sequential arrangement of the layers comprising 16, 32, 64, and 128 neurons allows the deep neural network to provide an ideal trade-off between the detection accuracy and computational resources, making it highly applicable to NB-IoT settings where the resource limitations are a critical issue, and the high detection rate is obligatory. This architecture shows a high level of capability in detecting known and previously unknown patterns of botnet attacks, which improves the overall safety of NB -IoT networks.

Model Training and Validation

To guarantee reliable and stable assessment, all models in this research are trained and validated in a similar experimental environment fixed after an extensive analysis study. The NB-IoT traffic dataset is cleaned initially to obtain the incomplete records, noisy records, and duplicate records, which may negatively affect the learning process. Next, the feature values are normalized by minmax normalization to ensure that all attributes have a similar number range, resulting in a more stable and convergent training. The dataset is further split into three mutually exclusive subsets: a training set used to learn the model parameters, a validation set used to optimize the hyperparameters and keep track of the performance, and a test set that is used in the final evaluation in the ratio of 60:20:20.

First, the SVM model design was implemented with a nonlinear kernel to successfully differentiate the normal and botnet traffic patterns. The regularization factor and kernel width are optimized during training, depending on how well the validation was done. The validation-based tuning can help the SVM to attain a trade-off between classification accuracy and generalization ability. Since the optimization of the SVM is a convex optimization, it will always converge, and hence it provides a reliable baseline when compared to deep learning models.

In the case of the deep learning models, such as CNN, LSTM, and the developed six-layer DNN, the training was performed with mini-batch learning to be able to work with a large amount of NB-IoT traffic data. The Adam optimization algorithm is used to optimize network weights because it has an adaptive learning rate mechanism and is resistant to deep architectures. The objective function is binary cross-entropy, which enables the models to acquire probabilistic differences between benign and malicious traffic. Hidden layers in the proposed models apply ReLU activation to help in efficient gradient flow, and the output layer applies a sigmoid to produce probabilities of classes.

Every deep learning model is trained through various epochs, and the performance on the validation set is constantly maintained to prevent overfitting. Early stopping is used in case the validation loss does not improve anymore, meaning that the learned models provide good generalization to unseen data. The dropout and batch normalization methods are adopted to make it more robust and less vulnerable to noise and data imbalance. The most desired weights of each model are retained according to the accuracy of the validation.

Validation is critical in evaluating how stable the model is, as well as learning behavior during the training procedure. Testing on a separate validation set allows monitoring of the generalization behavior continuously, such that what is observed to be performance is due to learning but not to overfitting to the training data. After the training stage, a final evaluation on an independent test set is done on the models. This is a separate set of data that is not considered during the training and validation step, but which can be used as an objective measure of the actual performance of the models. Through testing on unseen data, we can properly determine the extent to which the models extrapolate outside the training environment, so that the reported findings mirror their application in the real world, as opposed to the training samples overfitting the models. The systematic training and validation methodology makes it possible to give a meaningful and fair comparison of all models and makes sure that botnet attacks in NB-IoT networks is reliably detected.

Evaluation and Comparison

The efficiency of the suggested botnet detection framework is evaluated with the help of a thorough evaluation and comparative analysis of all the implemented models. Once the training phase is done, every model is evaluated on a separate dataset not used during the training process or the validation process. Such rigorous isolation guarantees that an accurate representation of the models in the real-world detection in NB-IoT conditions is represented by the reported results. The models in question are measured by using well-known

classification metrics, that is, accuracy, precision, recall, and F1-score; these measures combine to define the detection performance and the ability of the models to withstand false alarms.

The overall correctness of each model is measured using the accuracy, and precision and recall give more information about the reliability of botnet detection. The comparative analysis of the numerical results that were realized during the experiment analysis of both algorithms is provided by the following figure (Fig. 5).

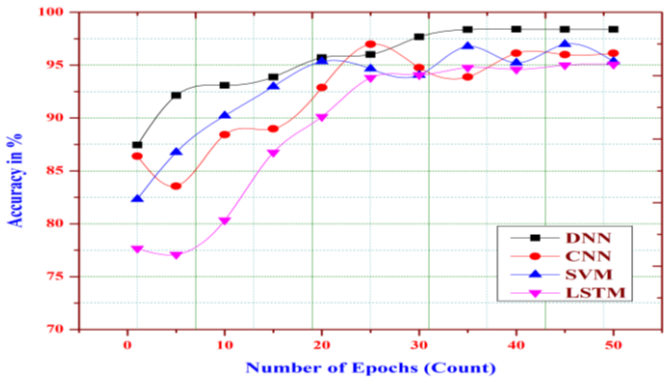


Fig. 5 Accuracy Convergence of SVM, CNN, LSTM, and DNN Models Across Training Epochs for NB-IoT Botnet Attack Detection

The diagram shows how four various learning models, which include SVM, CNN, LSTM, and DNN, change their accuracy in terms of classification over time during the training process in the scenario of NB-IoT botnet attack detection. The horizontal line shows the training epochs, and the vertical one shows the detection accuracy in the form of a percentage values. This visualization indicates that both of the models exhibit convergence behavior as well as a stable learning process over time.

All the models show relatively low accuracy at the very beginning of epochs, which is because of the first learning stage when the model parameters are being tuned. It has a consistent and steady improvement with the SVM model, and with a relatively small number of epochs, it attains a competitive accuracy, which implies that it has a high generalization ability with limited training. The CNN model also exhibits a moderate variation in the initial stages but as it acquires the relationships between the spatial features in the traffic, it becomes stable. The LSTM model begins with a low level of accuracy compared with other models because it is based on learning pattern over time, however, it can improve following the extra epochs, with long-term dependencies successfully learned.

The DNN model also performs better than the other methods throughout the majority of the epochs and has the highest and most consistent values of the accuracy at the later stages of the training. This action is indicative of the benefit of its richer architecture and more and more complex neuron setup, which allows learning of more complex nonlinear features interactions present in NB-IoT traffic better. This empirical study proves that despite all models enjoying longer training cycles, deep learning models, specifically DNN, exhibit statistically significant convergence, and have better detection capability when detecting botnet attacks in NB -IoT conditions.

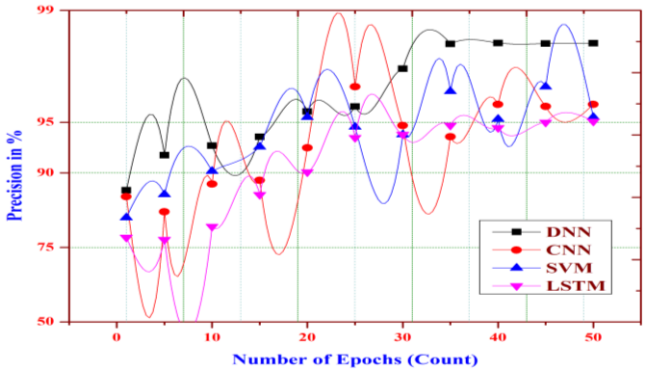


Fig. 6 Precision Variation of SVM, CNN, LSTM, and DNN Models Over Training Epochs for NB-IoT Botnet Attack Detection

The figure conducts a comparative study of the precision performance of SVM, CNN, LSTM, and DNN models with respect to training epochs in the case of the NB-IoT botnet detection. The horizontal axis represents the number of training epochs, whereas the vertical axis is the percentage of precision each of the models achieved, which is the capacity to classify botnet traffic correctly and the likelihood to reduce false alarms.In the early stages of training, one would expect to see observable variations in accuracy in all models, which is normal in early phases of training when model parameters are still settling. The SVM model demonstrates quite a constant improvement in the sense that similar values of precision are obtained with the increase in the training, which implies the successful learning of boundaries between benign and malicious traffic.

The CNN model has more variability in precision at the initial stages and mid-training stages, implying that the model is sensitive to spatial feature learning prior to achieving convergence. Conversely, the LSTM model in the analysis initiates with relatively low accuracy, and it progressively increases as temporal relationships in the trends of NB-IoT traffic are better represented with each epoch. The DNN model exhibits better and more consistent accuracy in most epochs and especially on the later training phases. This consistency in the findings shows high efficiency of its more sophisticated architecture and hierarchical layer structure in lowering false positive detection. The results emphasize the fact that the suggested DNN model, in comparison with others, provides a higher precision at all times, which makes it more reliable and efficient when it comes to detecting botnets in NB-IoT systems.

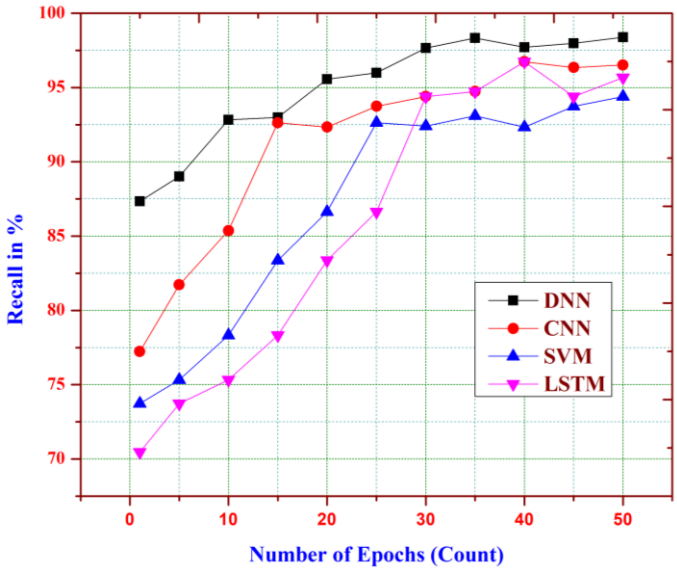


Fig. 7 Recall Convergence of SVM, CNN, LSTM, and DNN Models Across Training Epochs for NB-IoT Botnet Attack Detection

The number depicts the performance change of SVM, CNN, LSTM, and DNN models at each of the training epochs in detecting botnet attacks in an NB-IoT setup. The horizontal line displays the number of training epochs, and the vertical line displays the recall percentage, which shows the ability of the model to detect the malicious instances of the traffic correctly without missing the actual attacks.

All models initially record comparatively lower recall values in the initial phases of training, hence negligible detection ability at initial stages of the training. In this case, the SVM and LSTM models begin with relatively lower recall because they have more training iterations before they can successfully learn the decision boundaries and the temporal behavior, respectively. The CNN model exhibits greater improvement in the early epochs due to its spatial feature learning, but the DNN model exhibits greater recall even at the initial stages of training,which is a strong feature representation ability of the model.

The higher the epochs, the better the recall of all models, which means that more botnet activities are detected as the models keep learning. The best and most consistent recall results are produced by the DNN model at the majority of epochs, which indicates its efficiency at reducing the amount of false negatives by more profound and expressive learned nonlinear features. The CNN model achieves a competitive recall result, although it takes slight variations at the intermittent stages of training. The LSTM model does improve substantially following further epochs, implying that the model is powerful in grasping the temporal relationships when sufficient

training has been reached. The SVM model is seen to increase gradually and co-levels off at a slightly lower recall than the deep learning methods.

On the whole, the figure proves that the DNN is always the best model when it comes to recall, despite the fact that all of them are characterized by a long training period. It means that it is more effective in identifying larger fraction of botnet attacks in NB-IoT networks and therefore can be a good option when the security requirements of an application are critical and the reduction of missed detections is a priority.

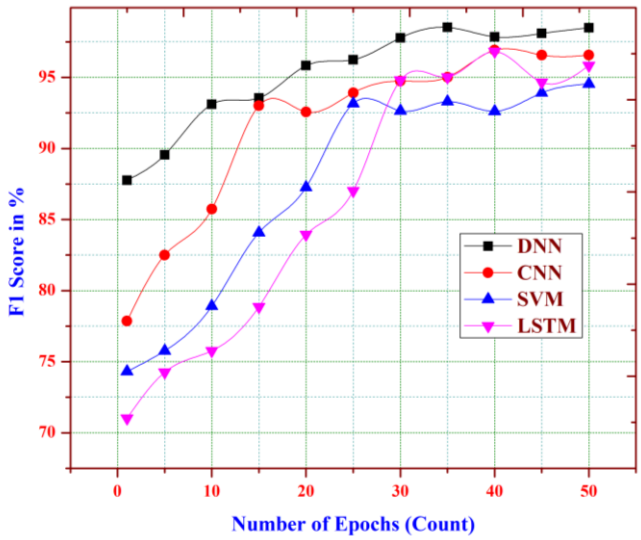


Fig. 8 F1Score Convergence of SVM, CNN, LSTM, and DNN Models Across Training Epochs for NB-IoT Botnet Attack Detection

The figure shows the changes in the F1-score of four distinct classification models including SVM, CNN, LSTM and DNN during training through multiple epochs in an NB-IoT botnet detection system. The above figure plots the number of training epochs versus the x-axis, and the F1-score as a percentage versus the y-axis, which is a single measure of detection performance by averaging precision and recall.

At the initial phases, the models show rather low F1-score values, and it is natural at the first stage of learning when the models are yet to become acquainted with the underlying traffic patterns. These SVM and LSTM models exhibit slower initial learning, which is indicative of the time cost to determine effective decision boundaries and infer sequential behaviour, respectively. The CNN model also has a faster increase in F1-score in initial training on the basis of its capability to acquire spatial feature associations in the data. On the contrary, the DNN shows better performance at the initial stages, which means that it has the potential to produce meaningful feature representations despite the few training iterations.

As the training is further progressed, the F1-scores of each of the models go up steadily, which is an indication of a better trade-off between correctly classifying botnet traffic and false classification. The DNN shows better learning efficiency and stronger robustness as it has the highest and most consistent F1-score on later epochs. Though CNN and LSTM models tend to achieve similar performance at increased epochs, they have a slight disparity during the convergence. The SVM model reaches a low F1-score compared to the deep learning methods, indicating its weaknesses in recognizing in-depth nonlinear attack patterns.

The figure shows that deep learning models are more balanced and reliable in detection performance as compared with conventional machine learning methods. The DNN also has the highest level of consistency in F1-score, which highlights the usefulness of the algorithm in the detection of botnet attacks in NB-IoT networks without sacrificing the strong trade-off between precision and recall.

Out of all the evaluated methods, the proposed six-layer DNN is the one that the evaluation metrics show performs better on most. The dense architecture of it that is gradually getting larger allows it to acquire hierarchical and nonlinear interactions between features, and thus enhance the generalization to novel botnet behaviors. Although the DNN has a greater computational cost than SVM, its superior detection performance and resilience offset the cost, more so in centralized NB-IoT security monitoring systems.

On the whole, the comparative analysis shows that the deep learning models are much more effective than conventional machine learning algorithms at identifying sophisticated botnet attacks in the NB-IoT networks. These findings establish that a more precise botnet detection is achieved through the integration of spatial, temporal, and nonlinear features learning mechanisms, and the proposed framework is appropriate in terms of real-world NB-IoT security systems.

Conclusion And Future Scope

The findings of this paper revealed that the methods based on artificial intelligence can be effectively used to promote the security of NB-IoT networks in relation to attacks by botnets. The comparative analysis of SVM, CNN, LSTM, and DNN models revealed that deep learning methods had a strong benefit in acquiring multifaceted and heterogeneous traffic habits. The DNN model was always the superior-detecting model, shown to have better detection capabilities and a faster convergence and reliability in recognizing malicious traffic streams and infected NB-IoT devices. Contrary to the classical form of security solutions, AI-based detection models demonstrated the ability to follow dynamically changing attack patterns, as well as help with near real-time detection, and maintain the requirements of the NB-IoT setting, including the strict energy and bandwidth requirements. In spite of the good results of the suggested framework in terms of detection, more enhancement methods may be investigated to increase its use in practice. Among the key directions to be narrowed down to in the future, one might highlight the development of lightweight and energy-conscious AI models that ensure high detection rates and low overhead computation and memory consumption.

References

1. Koshy, A. S., & Alankar, B. (2023, August). NB-IoT: Review, Comprehensive Study & Open Issues. In 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) (pp. 1486-1490). IEEE.

2. Borges, L., Ferreira, F. K., & Prade, L. R. (2025, April). NB-IoT in the Real World: A Systematic Review of Practical Applications. In 2025 IEEE Latin Conference on IoT (LCIoT) (pp. 133-136). IEEE.

3. Khond, S., Chythanya, K. R., Munawar, S., Mandava, M., & CG, K. (2023, October). Narrow Band Internet of Things (NB-IoT) for Wireless Communication and Network Optimization. In 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT) (pp. 1-6). IEEE.

4. Balan, R. S., Gouri, M. S., Senthilnathan, T., Gondkar, S. R., Gondar, R. R., Zeema, J. L., & Jothikumar, R. (2023). Development of smart energy monitoring using NB-IOT and cloud. Measurement: Sensors, 29, 100884.

5. Labdaoui, N., Nouvel, F., & Dutertre, S. (2023, July). Energy-efficient IoT Communications: A comparative study of Long-Term Evolution for Machines (LTE-M) and narrowband internet of things (NB-IoT) technologies. In 2023 IEEE Symposium on Computers and Communications (ISCC) (pp. 823-830). IEEE Computer Society.

6. Iiyambo, L., Hancke, G., & Abu-Mahfouz, A. M. (2024). A survey on NB-IoT random access: Approaches for uplink radio access network congestion management. IEEE Access, 12, 95487-95506.

7. Qasim, N. H., Salman, A. J., Salman, H. M., AbdelRahman, A. A., & Kondakova, A. (2024, April). Evaluating NB-IoT within LTE networks for enhanced IoT connectivity. In 2024 35th Conference of Open Innovations Association (FRUCT) (pp. 552-559). IEEE.

8. Kousias, K., Rajiullah, M., Caso, G., Ali, U., Alay, O., Brunstrom, A., ... & Di Benedetto, M. G. (2023). A large-scale dataset of 4G, NB-IoT, and 5G non-standalone network measurements. IEEE Communications Magazine, 62(5), 44-49.

9. Muteba, K. F., Djouani, K., & Olwal, T. (2022). 5G NB-IoT: Design, considerations, solutions and challenges. Procedia Computer Science, 198, 86-93.

10. Reshma, R., Mohanasundaram, N., & Santhosh, R. (2025). Design strategy for narrowband internet of things with its scope and challenges of security solutions. Smart Factories for Industry 5.0 Transformation, 187-227.

11. Attaullah, H. M., Ali, S., Aslam, M. I., Ahmed, I., & Pathan, A. S. K. (2025). An Efficient Risk Impact and Vulnerability Assessment Framework for NB-IoT Networks: Risk Impact and Vulnerability Assessment Framework for NB-IoT Networks. Sukkur IBA Journal of Computing and Mathematical Sciences, 9(02), 13-27.

12. Mohan, V. S., Sankaran, S., Nanda, P., & Achuthan, K. (2023). Enabling secure lightweight mobile Narrowband Internet of Things (NB-IoT) applications using blockchain. Journal of Network and Computer Applications, 219, 103723.

13. Muteba, K. F., Djouani, K., & Olwal, T. (2022). 5G NB-IoT: Design, considerations, solutions and challenges. Procedia Computer Science, 198, 86-93.

14. Du, X., & Lai, M. (2023, November). Research on NB-IOT Device Access Security Solutions. In 2023 3rd International Conference on Electronic Information Engineering and Computer (EIECT) (pp. 419-424). IEEE.

15. Qadir, J., Cabus, J. E. U., Butun, I., Lagerström, R., Gastaldo, P., & Caviglia, D. D. (2023). Analysis of LPWAN: Cyber-security vulnerabilities and privacy issues in LoRaWAN, Sigfox, and NB-IoT. In Low-power wide-area networks: opportunities, challenges, risks and threats (pp. 139-170). Cham: Springer International Publishing.

16. Tengshe, R., & Akanksha, E. (2023, April). Security in LP-wan technologies: challenges and solutions. In 2023 IEEE 8th International Conference for Convergence in Technology (I2CT) (pp. 1-4). IEEE.

17. Menon, U. V., Kumaravelu, V. B., Kumar, C. V., Rammohan, A., Chinnadurai, S., Venkatesan, R., ... & Selvaprabhu, P. (2025). AI-powered IoT: A survey on integrating artificial intelligence with IoT for enhanced security, efficiency, and smart applications. IEEE Access.

18. Dhanalakshmi, M., Kumar, G. N. K., Himabindu, G., Gosavi, V. R., Ganesh, C. S., & Premanand, R. (2025). Advanced Machine Learning Innovations in Embedded Systems and Narrowband Internet of Things (NB-IoT) Devices. In Integrating Artificial Intelligence Into the Energy Sector (pp. 331-356). IGI Global Scientific Publishing.

19. Salendra, S., Chaitanya, J., Krishna, B., Veeru, B., & Parvez, F. A. (2024, November). Artificial Intelligence and Automation on Tackle the 5G Network's Complexity with Edge Computing Integration. In 2024 International Conference on Advances in Computing, Communication and Materials (ICACCM) (pp. 1-6). IEEE.

20. Ashrif, F. F., Sundararajan, E. A., Hasan, M. K., Ahmad, R., Hashim, A. H. A., & Talib, A. A. (2024). Provably secured and lightweight authenticated encryption protocol in machine-to-machine communication in industry 4.0. Computer Communications, 218, 263-275.

21. Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. IEEE Communications Surveys & Tutorials, 21(3), 2671-2701.

22. Du Feng, J. H., & Hongli, J. (2024). Research on communication network security based on narrowband Internet of Things and encryption algorithms. Journal of Big Data and Computing (ISSN: 2959-0590), 2(2), 113-116.

23. Lata, S., Alrowais, F., & Urooj, S. (2024, December). ADCS-IoT: Anomaly Detection with Cryptographic Security Framework for the Internet of Things. In 2024 17th International Conference on Sensing Technology (ICST) (pp. 1-6). IEEE.

24. Zhao, G., Chen, H., & Wang, J. (2023). A lightweight block encryption algorithm for narrowband internet of things. Peer-to-Peer Networking and Applications, 16(6), 2775-2793.

25. Zhan, G., Jiang, Y., Yin, X., & Li, S. (2021, March). Research on security of NB-IoT based on cryptography. In 2021 4th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE) (pp. 1207-1211). IEEE.

26. Ahmed, K. R., Chowdhury, M. A. R., Karim, M. R., Khan, M. S., Hosien, M. A., Nesar, S. T., & Bazan-Antequera, R. (2025, July). Enhancing Signature-Based Intrusive Detection System (IDS) for IoT Networks Using Machine Learning Algorithm. In 2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN) (pp. 1-6). IEEE.

27. Rehman, F., Mushtaq, F., & Zaman, H. (2024, October). A Host-based Intrusion Detection: Using Signature-based and AI-driven Anomaly Detection for Enhanced Cybersecurity. In 2024 4th International Conference on Digital Futures and Transformative Technologies (ICoDT2) (pp. 1-7). IEEE.

28. Sheikh, T. U., Rahman, H., Al-Qahtani, H. S., Hazra, T. K., & Sheikh, N. U. (2019, October). Countermeasure of attack vectors using signature-based IDS in IoT environments. In 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 1130-1136). IEEE.

29. Sheikh, N. U., Rahman, H., Vikram, S., & AlQahtani, H. (2018). A lightweight signature-based IDS for IoT environment. arXiv preprint arXiv:1811.04582.

30. Nasir, M. H., Arshad, J., & Khan, M. M. (2023). Collaborative device-level botnet detection for internet of things. Computers & Security, 129, 103172.

31. Farooq, M., & Khan, M. H. (2022). Signature-Based Intrusion Detection System in Wireless 6G IoT Networks. Journal on Internet of Things, 4(3).

32. Chakraborty, S., Pandey, S. K., Maity, S., & Dey, L. (2024). Detection and classification of novel attacks and anomaly in IoT network using rule based deep learning model. SN Computer Science, 5(8), 1056.

33. Awotunde, J. B., Chakraborty, C., & Adeniyi, A. E. (2021). Intrusion detection in industrial internet of things network-based on deep learning model with rule-based feature selection. Wireless communications and mobile computing, 2021(1), 7154587.

34. Sadikin, F., & Wiranda, N. (2023). Anomaly Detection In Zigbee-Based Iot Using Secure And Efficient Data Collection. Jurnal Teknologi Informasi Universitas Lambung Mangkurat (JTIULM), 8(1), 7-16.

35. Bhavsar, M., Roy, K., Kelly, J., & Olusola, O. (2023). Anomaly-based intrusion detection system for IoT application. Discover Internet of things, 3(1), 5.

36. Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. Computers and Electrical Engineering, 99, 107810.

37. Walling, S., & Lodh, S. (2024). Enhancing IoT intrusion detection through machine learning with AN-SFS: a novel approach to high performing adaptive feature selection. Discover Internet of Things, 4(1), 16.

38. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3), 12-22.

39. Otoum, Y., Liu, D., & Nayak, A. (2022). DL-IDS: a deep learning–based intrusion detection framework for securing IoT. Transactions on Emerging Telecommunications Technologies, 33(3), e3803.

40. Sedjelmaci, H., Senouci, S. M., & Taleb, T. (2017). An accurate security game for low-resource IoT devices. IEEE Transactions on Vehicular Technology, 66(10), 9381-9393.

41. Jose, J., & Jose, D. V. (2023). AS-CL IDS: Anomaly and signature-based CNN-LSTM intrusion detection system for internet of things. Int. J. Adv. Technol. Eng. Explor, 10(109).

42. AR, S., & Katiravan, J. (2025). Enhancing anomaly detection and prevention in Internet of Things (IoT) using deep neural networks and blockchain based cyber security. Scientific Reports, 15(1), 22369.

43. Savic, M., Lukic, M., Danilovic, D., Bodroski, Z., Bajović, D., Mezei, I., ... & Jakovetić, D. (2021). Deep learning anomaly detection for cellular IoT with applications in smart logistics. IEEE Access, 9, 59406-59419.

44. Salva-Garcia, P., Alcaraz-Calero, J. M., Wang, Q., Bernabe, J. B., & Skarmeta, A. (2018). 5G NB-IoT: Efficient Network Traffic Filtering for Multitenant IoT Cellular Networks. Security and Communication Networks, 2018(1), 9291506.

45. Alanazi, R., & Aljuhani, A. (2023). Anomaly Detection for Industrial Internet of Things Cyberattacks. Computer Systems Science & Engineering, 44(3).

46. Mutambik, I. (2024). An efficient flow-based anomaly detection system for enhanced security in IoT networks. Sensors, 24(22), 7408.

47. Alsoufi, M. A., Razak, S., Siraj, M. M., Nafea, I., Ghaleb, F. A., Saeed, F., & Nasser, M. (2021). Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review. Applied sciences, 11(18), 8383.

48. Hassan, J., Abid, M. K., Ghulam, A., Fakhar, M. S., & Asif, M. (2023). A survey on blockchain-based intrusion detection systems for iot. VAWKUM Transactions on Computer Sciences, 11(1), 138-151.

49. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Breitenbacher, D., & Shabtai, A. (2018). Detection of IoT botnet attacks (N-BaIoT). UCI Machine Learning Repository.