Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Hybrid Machine-Learning Based Security Algorithm for Improved Detection of Distributed-Denial-of-Services Attacks in Internet of Things Networks

Coster Baloyi

costbal@gmail.com

Tshwane University of Technology

Topside Ehleketani Mathonsi

Tshwane University of Technology

Tshimangadzo Mavin Tshilongamulenzhe

Tshwane University of Technology

Tonderai Muchenje

Tshwane University of Technology

Du Plessis Daniel

University of Johannesburg

Article

Keywords: Distributed Denial of Service (DDoS), Machine Learning-based Security (MLBS), Recurrent Neural Network (RNN), K-Nearest Neighbors (KNN), Internet of Things (IoT)

Posted Date: September 1st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10456190/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on October 26th, 2026. See the published version at https://doi.org/10.1038/s41598-026-77311-0.

A Hybrid Machine-Learning Based Security Algorithm for Improved Detection of Distributed-Denial-of-Services Attacks in Internet of Things Networks

Coster Baloyi [*,], Topside Mathonsi [*,**], Tshimangadzo Tshilongamulenzhe [*,**], Tonderai Muchenje [*,**], and Deon du Plessis**

Tshwane University of Technology
2 Aubrey Matlakala St, Soshanguve,
Pretoria, 0001
South Africa
costbal@gmail.com, mathonsite@tut.ac.za, *tshilongamulenzhetm@tut.ac.za,* muchenjet@tut.ac.za, danieldp@uj.ac.za,
Funding: The manuscript was funded by Tshwane University of Technology.

Abstract - The Internet of Things (IoT) has emerged as a critical enabler of intelligent and autonomous systems, supporting applications across sectors such as agriculture, healthcare, education, and smart infrastructure. However, the increasing connectivity of IoT devices has significantly expanded the attack surface, making these networks particularly vulnerable to Distributed Denial of Service (DDoS) attacks. Existing detection approaches, including Decision Tree (DT), Linear Regression (LR), and K-means, are often limited by suboptimal classification accuracy, high response latency, and poor adaptability to evolving and zero-day attack patterns, thereby compromising their effectiveness in dynamic IoT environments. To address these limitations, this study proposes a Machine Learning-Based Security (MLBS) framework that integrates K-Nearest Neighbors (KNN) and Recurrent Neural Networks (RNN) to enhance the detection and mitigation of DDoS attacks in IoT networks. The proposed hybrid model leverages the temporal learning capability of RNNs and the instance-based classification strength of KNN to improve both detection accuracy and generalization. The framework was evaluated using the MATLAB simulation environment, where it achieved an accuracy of 98.72%, precision of 97.95%, recall of 98.94%, and an F1-score of 98.39%, outperforming conventional baseline models. Beyond performance improvements, the proposed MLBS framework demonstrates strong robustness under noisy data conditions, class imbalance, and previously unseen traffic patterns, indicating its suitability for real-world deployment. The key contribution of this work lies in providing a scalable, adaptive, and high-precision intrusion detection solution that enhances the resilience of IoT networks against both known and emerging DDoS threats.

Keywords - Distributed Denial of Service (DDoS), Machine Learning-based Security (MLBS), Recurrent Neural Network (RNN), K-Nearest Neighbors (KNN), Internet of Things (IoT)

1. Introduction

The Internet of Things (IoT) has emerged as a foundational technology of the Fourth Industrial Revolution, alongside big data and Artificial Intelligence (AI), enabling pervasive connectivity and intelligent data exchange across diverse physical and digital environments. IoT refers to a network of heterogeneous physical and virtual devices embedded with sensors, communication capabilities, and unique identifiers, which autonomously collect, transmit, and process data without requiring direct human intervention [1, 2, 3]. The rapid proliferation of IoT technologies has led to an unprecedented growth in interconnected devices, with estimates suggesting that over 127 devices are connected to the Internet every second, and projections exceeding 27 to 50 billion devices globally by 2025 to 2030 [1, 4]. This exponential expansion is expected to significantly influence global economic development, with IoT technologies contributing substantially to global gross domestic products through increased automation, efficiency, and data-driven decision-making.

IoT applications have permeated multiple sectors, including smart agriculture, healthcare, transportation, smart cities, and industrial automation, thereby transforming traditional systems into intelligent, data-driven ecosystems. In agriculture, IoT supports precision farming by enabling real-time monitoring of environmental conditions, livestock, and resource utilization, which enhances productivity, optimizes resource usage, and promotes sustainability [5]. Similarly, in healthcare, IoT facilitates remote patient monitoring, early diagnosis, and improved healthcare service delivery. However, despite these transformative benefits, the rapid expansion and heterogeneity of IoT ecosystems introduce significant security and privacy challenges. IoT environments are inherently characterized by resource-constrained devices, diverse communication protocols, and open wireless communication channels, which collectively increase their exposure to cyber threats and attacks.

As a result, IoT networks have become attractive targets for a wide range of cyber-attacks, including Distributed Denial of Service (DDoS) attacks, spoofing, botnet attacks, and injection-based threats [5, 6]. Among these, DDoS attacks pose a particularly critical threat due to their ability to overwhelm network resources, disrupt service availability, and degrade system performance. The impact of such attacks is amplified in IoT environments, where compromised devices can be exploited to form large-scale botnets capable of launching coordinated and high-volume attacks. Furthermore, IoT devices often serve as entry points for attackers, enabling lateral movement across interconnected systems such as cloud platforms, databases, and critical infrastructure, thereby increasing the potential severity of security breaches.

The rapid expansion of IoT networks has significantly increased the attack surface for cyber threats, particularly DDoS attacks, which disrupt service availability and degrade network performance. Although numerous machine learning-based security algorithms, including Support Vector Machines (SVM), Random Forest (RF), and K-Nearest Neighbours (KNN), have been proposed to detect and prevent DDoS attacks in IoT environments, these approaches exhibit several critical limitations [32, 33]. In particular, many existing

models suffer from limited generalization capability, resulting in poor performance when applied to large-scale and highly dynamic IoT traffic patterns. Additionally, high false positive rates and low precision reduce the reliability of detection systems, while increased latency negatively impacts real-time response and overall network efficiency

Below are the contributions of the study:

1) A novel hybrid Machine Learning-Based Security (MLBS) framework is proposed, integrating K-Nearest Neighbors (KNN) and Recurrent Neural Networks (RNN) to enhance the detection of both known and zero-day DDoS attacks in IoT environments. The hybrid design leverages the complementary strengths of instance-based learning and temporal pattern recognition to improve classification robustness.

2) An adaptive detection mechanism is developed that effectively addresses key limitations of existing approaches, including misclassification, high false positive rates, and poor generalization under noisy and imbalanced data conditions.

3) Comprehensive experimental validation demonstrates the effectiveness of the proposed model, achieving high performance metrics (98.72% accuracy, 97.95% precision, 98.94% recall, and 98.39% F1-score) and outperforming conventional baseline algorithms in terms of detection accuracy and reliability.

4) Enhanced robustness and real-world applicability are demonstrated through the model's ability to maintain stable performance under challenging scenarios, including noisy data, class imbalance, and previously unseen attack patterns, making it suitable for deployment in dynamic IoT environments.

The remainder of the paper is organized as follows: In section two, this paper presents a review of literature on numerous forms of DDoS attacks available in the IoT network. In section three, this paper focuses on system architecture and design, giving details of how the paper was carried out to achieve the desired results. In section four this paper presented the proposed MLBS algorithm. In section five, this paper presented the system implementation in detail. In section six the paper presents performance evaluation and simulation setup. In section seven the paper discusses the results. In section eight the paper presents the baseline comparison of the algorithms and lastly in section nine the paper presents conclusions and the future work.

2. Related Works

This section critically reviews existing ML and deep learning approaches that have been applied to the detection and mitigation of DDoS attacks within IoT networks. Given the dynamic, large-scale, and resource-constrained nature of IoT environments, traditional security mechanisms have proven insufficient, thereby necessitating the adoption of intelligent, data-driven detection techniques.

The section examines a range of previously proposed models, analysing their underlying algorithms, performance outcomes, and applicability to real-world IoT scenarios. Particularly attention is given to their strengths in handling high-volume and evolving attack patterns, as well as their limitations in areas such as real-time detection, scalability,

computational efficiency, and accuracy. By synthesizing findings from recent studies, this section identifies key research gaps that motivate the development of the proposed approach in this study.

In most cases intrusion detection approaches have multiple challenges in IoT environment due to the increasing traffic volumes, dynamic traffic patterns, heterogeneous systems and evolving DDoS attacks. For instance, [7] proposed a model that optimized automation detection of intrusions into IoT using self-learning. This model divided modelling into two layers in order to enhance the detection of intrusions attacks to learning algorithms by integrating algorithmic approaches and reduction of learning costs. The Cost-Sensitive Learning algorithm was used for cost discrimination, and the Meta-Learning was dedicated to metadata management. [7] achieved the classification quality rate of approximate 93.66%, consistency rate of 0.883, the false positives 0.2 and 4.2%. The sensitivity was between 77.8 and 97.3%, The specification was between 95.8% and 99.8%, the precision was 0.951. Similarly, [8] incorporated DL and ML algorithms to increase reliance of the model to improve challenges of missing out more complex and unique attacks. Conventional intrusion detection uses predefined criteria, and detection signatures increase the risk of missing out on the increasing complexity. Hence the proposed hybrid ML algorithm to improve accuracy rate and precision rate.

The recent IoT studies show multiple ML techniques to detect DDoS and malware attacks in IoT, however there is a constant rise in DDoS attacks due to elements of security weakness, distributed botnet and encryption challenges in IoT environment. These challenges have created bottlenecks on the current approaches, for instance, [9] proposed an IoT malware traffic evaluation technique to improve the safeguard of IoT devices on gateway level by detecting unknown malware. Similarly, [10] proposed a ML model to improve accuracy rate in IoT network by mitigation anomalous network traffic patterns. These approaches are challenged by an increased complexity and robustness of the attack surface. The technique achieved the accuracy rate of 91.32%, the precision rate of 91.67% and the recall rate of 91.03% showing a room to improve the model to handle the dynamic network traffic patterns using hybrid model and real-time datasets with diverse modern IoT attacks.

The increasing complexity and robustness of DDoS attacks have impacted most ML based techniques since IoT network devices operate in extremely vulnerable environments with unique attacks and zero-day attacks emerging frequency. For instance, [11] proposed a typical approach to deal with unique threats and attacks in the IoT devices by using strong encryption techniques SHA-256 hashing function to protect data at rest and data in transit. Their technique included authentication techniques namely identity or device authentication, procedural authentication and token based that verify the identity of an entity accessing the network service. Similarly, [12] proposed a ML-driven technique used for intelligent, adaptive and robust IoT

network environment with the aim to building the next generation technique capable of detecting unknown attacks in IoT ecosystem. The approach provides. The proposed solutions are lacking when it comes to increasing complexity and not cost effective. Furthermore, there remains a concern regarding security of data at rest particularly as the adoption of IoT network continues to grow. Hence the need to introduce the hybrid ML model to improve DDoS attacks accuracy rate.

Although the Machine-learning based models have achieved high precision rate and strong classification performance, many of these approaches rely on single learning models which may have limitations on robustness and adaptability. For instance, [13], proposed a new intrusion detection optimization (NIDO) model that endorsed selecting among machine learning methods namely supervised, reinforcement and unsupervised to effectively detect intrusions in the IoT devices. The optimal method choice for the model was based on the cost-sensitivity and the ability to minimize the learning impact of the attacks. The results showed classification rate of 93,66%, Precision rate 96%, False Positive 0,3%. The approach focused on the latest DDoS attacks dataset called CICIDS2018 with up-to-date real-world data which may introduce complexity issues when network traffic increases hence the proposed framework is introducing a hybrid ML-based model to enhance the detection accuracy and robustness of DDoS detection mechanisms in IoT networks.

An SDN-based multi-layer intrusion detection and mitigation models have achieved high accuracy rate, however they rely on static datasets and do not account for the dynamic nature of IoT network traffic, for instance [14] proposed a four-model DDoS attack detection and mitigating (FMDADM) framework for the purpose of detecting DDoS attacks at elevated and minimal rates and classifying traffic attacks and flash crowds while safeguarding local and remote IoT nodes at the same time preventing infections from propagating to the ISP level. The approach achieved 99.79% accuracy, 99.43% precision, 99.77% F-measure, 99.79% recall, 99.95% specificity, 00.21% negative predictive value, 00.91% false positive rate, 00.23% false negative rate and 2.64 average detection time benchmarks. Similarly, [15] proposed a technique for detecting and mitigating DDoS attacks in SDN in real time. The proposed framework focuses primarily on SDN-enabled IoT environments and does not address detection of emerging attack patterns.

An IoT-focused cybersecurity architecture to strengthen real-time detection and adaptive mitigation DDoS attacks on the IoT network is used previously by [16]. This approach is not using hybrid approach to handle the growing complexity of attack landscape. [16] proposed a defense framework called learning-driven detection mitigation (LEDEM) that uses semi-supervised machine-learning algorithm to mitigate DDoS attack on IoT environment. This framework makes use of software-defined network (SDN) and cloud paradigms. The testing of LEDEM was done in the testbed and

emulated topology. The results achieved an accuracy rate of 96.28% in detecting DDoS attacks and were compared with the state-of-the-art solutions. This type of approach lacks adaptability as it only uses single classification method and is incapable of dealing with complex IoT network traffic patterns.

ML based models are proposed to detect unknown attacks in IoT environment, the increasing attack surface has expanded dynamically causing the need for a novel framework to intelligently detect malware attacks including zero-day attacks. [17] proposed a unique machine learning (ML) based security algorithm that automatically deals with developing security features related to IoT environment. This is a supervised learning algorithm that makes use of both Network Functions Virtualization (NFV) and Software Defined Network (SDN) enablers to mitigate DDoS attacks on the IoT environment. The dataset used was immutable. The results showed 99.71% accuracy detection. The limitation of this algorithm is that the cost of misclassification is higher as for the processing time.

Deep learning models are proposed to detect and classify malware attacks automatically and achieve high accuracy rate. This approach is effective on increasing complex IoT network pattens but lacks when it comes to zero-day attacks. For instance, [18] proposed a deep learning-based framework that obtained internal feature groupings from gated recurrent unit (GRU) layers to detect DDoS attacks and classify them into their attack classes. The results showed 99% accuracy with 99% macro precision, 99% macro recall, and 99% macro F1-Score. For attack classification, the proposed GRU feature fusion model showed 99% accuracy with 99% macro precision, 99% macro recall, and 99% macro F1-Score. The Kernel principal component analysis (Kernel PCA) was used to obtain the optimal features which are then combined together. Furthermore, DDoS attack detection and its categorization are done using the fully associated network. This framework can be used in real time to efficiently detect DDoS attacks in IoT environment and organize them into their respective categories. However, the datasets used were insufficient for analysing the behaviour of IoT network traffic.

ML based techniques are used to improve detection rate and classify DDoS attacks effectively. The challenge with these attacks is that the [19] proposed a reversed hour-glass-based layered network classifier for classification and a hybrid intelligent system for feature selection to improve accuracy in the detection of DDoS attacks. This framework utilised three types of datasets namely: KDD-Cup99 database, NSL-KDD Database and UNSW-NB15 database for identifying new and old DDoS attacks. The framework achieved the goal by using the reversed hour-glass-based network by sampling the data with the spread in the number of layers for effective training. The results of the proposed model showed 99.97% of NSL-KDD, 99.57% of KDD-CUP99, and 99.73% of UNSW NB15 in accuracy detection. The limitation to this model is that it can only handle one classification method.

[20] proposed a hybrid Intrusion Detection System that combines with Long Short-Term Memory Network (LSTM) and Convolution Neural Network (CNN) algorithms to secure the spatial and temporal features of the network traffic. The amalgamation of CNN and LSTM algorithms has improved the intrusion detection and achieved 96.32% accuracy. This approach uses two regularization models namely: dropout model and L2 Regularization (L2Reg) to resolve the issues of overfitting. The proposed model improved the detection performance of zero-day attacks. The limitation to this approach is that it cannot handle real time datasets.

[21] proposed a unique hybrid Deep Learning (DL) approach that is built on Convolutional Neural Network (CNN) that classifies traffic flow to establish whether the traffic is normal or malicious attacks in the SDN environment. The proposed framework merges CNN with ML algorithms such as KNN and SVM to improve the anomalies detections. The results showed 97.47% of accuracy for CNN-RF and 97,37% of classification. The framework uses a regularization model called SD-Reg that was based on weight matrix of normal deviation to resolve the problem of overfitting and enhanced capability of detection of unknown attacks. The framework used the latest SDN dataset to train and evaluate the performance of the proposed framework. This approach cannot handle increasing complex IoT network pattens.

Many existing studies in the literature have focused on general detection of DDoS attacks in SDN-based network environment which does not . However, [22] proposed a hybrid ML technique that does not only focus on general detection of DDoS attacks but expand the scope in to other areas of SDN such as SCADA systems, IoT network, mobile network, Vehicular ad-hoc network and 5G. This comes with the benefit of exploring the security architecture of different SDN network environment, comparatively evaluating solution approaches and current threats. The supervised, unsupervised and deep learning methods limitations and applicability are highlighted for recommended to researchers. These include the type of an algorithm used to detect DDoS attacks, the feature selection and structure of the dataset characteristics. The study recommended DL models as they offer more performance. The CICIDS2017, KDDCip99, NSL-KDD and UNSW-NB15 datasets are found to be mostly used the literature. However, it was noticed that the normal and the network attack traffic in the datasets are often distributed unevenly.

[23] proposed a study that used Deep Learning-based IDS called DeepNonLocalNN model to train the Supervisory Control and Data Acquisition (SCADA) new dataset called SCADANet. The study used of the TCP/Modbus protocol to generate a virtual SCADA network. The new dataset is tailored for SCADA in cyber security research field. The simulation was conducted on both the normal network traffic as well as typical and specific SCADA cyber security attacks. The Wireshark, TShark and JA4+ were used for capturing and

evaluation of data. The study achieved good results using the local and global network traffic patterns. The experimental model included testing of 19 distinct classification algorithms that consisted of both ML and DL techniques. The evaluation of SCADA dataset was based on 14 different classes representing normal attacks and 13 others that represented distinct attack types. There are still existing weaknesses in cybersecurity of SCADA systems

Table 1: Literature Review Summary

Ref	Approach / Model	Key Contribution	Performance Metrics	Strengths	Limitations
[7]	Cost-Sensitive + Meta-Learning	Two-layer intrusion detection with cost optimisation	Accuracy: 93.66%, Precision: 0.951, FP: 0.2–4.2%	Good classification and cost reduction	Limited adaptability to dynamic IoT traffic
[8]	Hybrid ML + DL	Combines ML and DL to improve detection of complex attacks	Improved accuracy & precision (not quantified)	Handles complex attacks better than traditional IDS	Lacks detailed real-time validation
[9]	IoT Malware Detection at Gateway	Detects unknown malware at gateway level	Not specified	Improves IoT device security	Struggles with large-scale attack complexity
[10]	ML-based Anomaly Detection	Detects anomalous traffic patterns	Accuracy: 91.32%, Precision: 91.67%, Recall: 91.03%	Balanced performance	Not robust for dynamic traffic patterns
[11]	Encryption-based Security	Secures IoT data via encryption and authentication (SHA-256)	Not applicable	Strong data protection	Does not address attack detection
[12]	Adaptive ML-based IDS	Intelligent detection of unknown attacks	Not specified	Adaptive learning capability	High complexity and cost
[13]	NIDO Framework	Selects optimal learning method based on cost sensitivity	Accuracy: 93.66%, Precision: 96%, FP: 0.3%	Flexible model selection	Performance degrades with high traffic
[14]	FMDADM (SDN-based)	Multi-layer DDoS detection and mitigation	Accuracy: 99.79%, Precision: 99.43%, Recall: 99.79%	Very high performance	Uses static datasets, lacks adaptability
[15]	SDN Real-Time Detection	Real-time DDoS mitigation in SDN	Not specified	Real-time capability	Limited handling of emerging attacks
[16]	LEDEM Framework	SDN + cloud-based detection and mitigation	Accuracy: 96.28%	Good detection performance	Uses single model, lacks adaptability
[17]	ML + NFV + SDN	Automated security feature generation	Accuracy: 99.71%	High detection accuracy	High computational cost
[18]	GRU-based DL Model	Feature extraction and attack classification	Accuracy: 99%, Precision: 99%, Recall: 99%	High classification performance	Limited dataset generalisation
[19]	Reversed Hourglass Model	Multi-dataset training for DDoS detection	Accuracy: up to 99.97%	High accuracy across datasets	Single classification method

Ref	Approach / Model	Key Contribution	Performance Metrics	Strengths	Limitations
[20]	CNN + LSTM Hybrid	Combines spatial & temporal learning	Accuracy: 96.32%	Detects zero-day attacks	Not suitable for real-time datasets
[21]	CNN + ML Hybrid	Combines CNN with KNN/SVM	Accuracy: 97.47%	Improved anomaly detection	Cannot handle complex IoT patterns
[22]	Hybrid	DL models as they offer more performance	Not Specified	Does not only focus on general detection of DDoS attacks	datasets are often distributed unevenly
[23]	Hybrid	study achieved good results	Not Specified	dataset tailored for SCADA in cyber security research field	Existing weaknesses in cybersecurity of SCADA systems

The reviewed studies demonstrate significant progress in applying machine learning and deep learning techniques for detecting DDoS attacks in IoT environments. However, a critical limitation persists most existing approaches rely on static or offline datasets and are not designed to operate effectively under real-time, high-volume, and dynamic network conditions typical of IoT ecosystems. This limitation reduces their practical applicability, particularly in scenarios where rapid detection and response are essential to prevent service disruption.

Furthermore, many models struggle to balance detection accuracy, false positive rates, and computational efficiency, especially when deployed in resource-constrained IoT environments. These gaps highlight the need for a more robust and adaptive solution capable of handling real-time traffic while maintaining high detection performance.

To address these challenges, this study proposes an MLBS algorithm designed for real-time DDoS detection in IoT networks. The proposed approach focuses on achieving high detection accuracy, low false positive rates, and efficient processing of large-scale network traffic, thereby enhancing the reliability and responsiveness of IoT security systems.

3. System Modelling and Design

This paper presents the system modelling and design of the MLBS algorithm to detect DDoS attacks in IoT network. The proposed MLBS algorithm is designed by integrating KNN and RNN algorithms to form an advanced technique that will be effective in the detection of DDoS attacks including unknown attacks in IoT network. The MLBS algorithm ensures accuracy in the classification of DDoS attacks including unknown zero-day attacks in the IoT network. The system architecture of IoT Wireless Sensor Network (WSN) that depicts the layered structure and components that enable data transmission, sensing and processing in IoT network is discussed.

3.1 IoT System Architecture

This section presents the proposed architecture of the MBLS algorithm used to detect DDoS attacks in IoT networks and how the components collect data through interacting with one another. An initial step in designing IoT network devices is

discovering how it will interact with other parts of the world. The communication technology for IoT network devices has an impact on the device hardware requirements and costs.

Many interconnected actuators and sensors may be deployed in a wide area in a plant wirelessly called wireless IoT sensor network. This IoT is a network of sensors which enables every sensor to send data from node-to-node in the network data flow. The transmission of data from the sensors to the processing unit is offered in the network layer through communication protocols such as Wi-Fi and Bluetooth.

IoT wireless sensor network nodes are known to be cost effective technology and therefore can be scaled in large quantities. Furthermore, these wireless sensor devices are capable of operating on batteries and do not rely on high power requirements as they even employ technologies such as energy harvesting for low power usage. The IoT wireless sensor network architecture is presented in Figure 1.

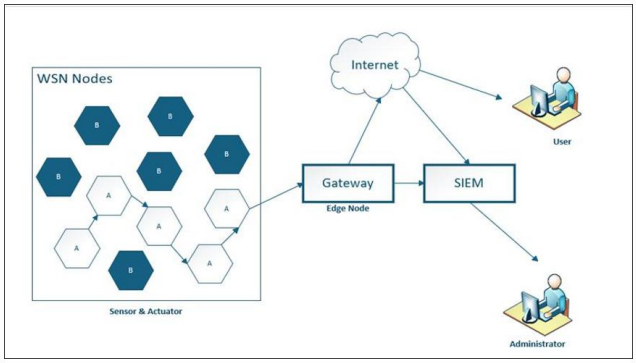


Figure 1: IoT wireless sensor network architecture [24]

Figure 1. provides a high-level overview of IoT wireless sensor network that connects multiple sensor nodes to collect data from WSN. Between the sensor network and the internet is an Edge node which operates as the Gateway. The Edge node is configured between the sensor network and the internet, in addition, Edge node has user interfaces, local processing and local storage capabilities. The Gateway node is a bridge between the internet and the senor node, and its role is to aggregate data communications from multiple Edge nodes. WSN is vulnerable to DDoS attacks due to their distributed nature and are likely to be attacked by the network layer DDoS attacks such as Sinkhole Attack, Sybil Attack, Hello flood and wormhole attacks [25].

The IoT network devices operate from everywhere, therefore it requires Wi-Fi connectivity. A security incident and event management is installed to monitor traffic from the internet gateway and send all the anomalies alerts to the administrator. An attack can be orchestrated to attack the network node that further flood to all other nodes as data is sent. This type of attack can result in DoS attacks and worse becoming DDoS attacks. This is accomplished by doubling an attack node from a single attack to many attacks that flood the IoT sensor network nodes, stopping the system completely. The traffic then sends an alarm to the WSN controller node.

4. The Overview of MLBS Algorithm

This section presents the design of the MLBS algorithm through the integration of the KNN and RNN algorithms. The step-by-step process of how the two algorithms were integrated is presented below:

4.1 K-Nearest Neighbors Algorithm (KNN)

KNN is a supervised learning algorithm that labels datasets of the training samples described as (p, q) in order to make prediction possible between p and q [26]. This research study utilizes the KNN algorithm due to its expertise of high accuracy in a wide variety of prediction-type problems. The three main parameters for KNN are positive integer (K), a hidden observation $h(p)$ and a distance d.

The KNN algorithm is an effective laid-back learning algorithm that has been established effectively in several studies. KNN algorithm helps MLBS algorithm to improve detection and prevention of DDoS attacks in IoT while integrated with RNN algorithm. KNN algorithm relies on the most used Euclidean Metric (EM) as its source. The EM computes the distance between two points in three-dimensional space [27, 28].

This research study follows the real-time DDoS attacks detection approach by analyzing the training dataset $\boldsymbol{X} = \{\boldsymbol{X_1}, \dots, \boldsymbol{X_n}\}$. For each mark G in the dataset X, the model identifies the KNN algorithm among the training marks, where the distance to the KNN is $\boldsymbol{s(X_1)}$, where $\boldsymbol{i} = \boldsymbol{1}$,…,n. In order to conclude training the $(\boldsymbol{1} - \boldsymbol{a})$ th percentile $\boldsymbol{sf}\big(\boldsymbol{G}(\boldsymbol{X})\big)$ of $\boldsymbol{sf}\big(\boldsymbol{G}(\boldsymbol{X})\big)\,\boldsymbol{of}\,\{\boldsymbol{sf}(\boldsymbol{X_1}), \dots, \boldsymbol{sf}(\boldsymbol{X_n}$. Where mark $\boldsymbol{G} =$ round $[(\boldsymbol{1} - \boldsymbol{a})\boldsymbol{N}]$. The EM computes KNN algorithm distance $\boldsymbol{sf}(\boldsymbol{X_t})$ for the training mark G. For real-time, attack evidence $\boldsymbol{\omega t}$ is calculated as shown in Equation (1) below:

$$\boldsymbol{\omega t} = \boldsymbol{r}[\log \boldsymbol{sf}(\boldsymbol{X_t}) - \log \boldsymbol{sf}(\boldsymbol{X}(\boldsymbol{G}))] \quad (1)$$

Where r represents the data dimensions. $\boldsymbol{\omega t}$ represent the algorithm performances under different approaches regardless of the increase in the observations of attacks packets. In other scenarios the Equation (2) might be used to improve detection accuracy and is given as decision statistic $\boldsymbol{P_t}$ as shown below:

$$P_t = \max\{P_t - 1 + \omega t, 0\} P0 = 0 \quad (2)$$

When the Equation (2) triggers and alert at the first instance $\boldsymbol{P_t}$ reaches the threshold at a time $\boldsymbol{T}$, as shown in Equation (3) below:

$$\boldsymbol{T} = \mathbf{min}\{\boldsymbol{t}: \boldsymbol{P_t} \geq \boldsymbol{hr}\} \quad (3)$$

Where it represents time obtained at the first time. $\boldsymbol{P_t}$ represents the sequence function P that depends on time while $\boldsymbol{hr}$ is the target threshold that requires the sequence function to be greater or equals to the threshold.

The real-time attack detection KNN algorithm is presented in Algorithm 1

Algorithm 1: The real-time attack detection KNN algorithm

1. ***Input:***
2. Number of Layers L // Initialization
3. Real-time input data: $X_{realtime}$
4. Training dataset $X = \{X_1, X_2, \dots, XM\}$ // Examine traffic pattens
5. **Set:** $s = 0, t = 0$ // initialization
6. **For** $n = 1$ to N **do**
7. Partition training patterns into $X_n\, P_1 and\, X_n\, P_2$
8. Determine $Ln(a)$
9. **While** $P_t < hr$ **do**

10. $t = t + 1$
11. Get new data $\{Xnt\}$ and compute $\{Xnt\}$
12. $Pnt = \max\{Pnt + Xnt, 0\}$
13. $P_t = \sum Nn = 1\ Pnt$
14. **If** $P_t >= hr\ then$
15. DeclareAttack($T = t$)
16. **End if**
17. **End while**
18. **End For**
19. ***Classification Algorithm for Offline Attack***
20. **For** $1 \in [1, L]$ **do**:
21. **Set** parameters: // Initialization
22. Weight matrix: $W1 = 0$
23. Bias vector: $b1 = 0$
24. Weight matrix for decryption: $W1 = 0$
25. Bias vector for decryption: $b1 = 0$
26. ***Encryption layer:***
27. Calculate encoding or hidden representation using equation (1): // Instantaneous attack evidence
28. $hn = f\theta(xn) = \sigma(Wxn + b)$
29. $h1 = S\ (W1xl1 + b1)$
30. ***Decryption layer***
31. **While** not loss = stopping criteria do:
32. Compute yi using equation (2): // triggers an alarm when threshold is met at first time T
33. $yi = g\theta(hn) = \sigma(Whn + b)$
34. Calculate the loss function: binary cross-entropy
35. Update layer parameters $\theta = \{W, b\}$
36. **End while**
37. **End for**
38. ***Classifier***
39. Fully connected neural network with soft-max activation at the output layer
40. Set $(Wl + 1, bl + 1)$ at the supervised layer
41. Execute the labels for each sample xn of the training dataset X
42. Apply dropout and normalization batch for speeding up the calculation
43. Complete back-propagation in a supervised manner for parameters tuning of the layers, loss function categorical cross entropy
44. Specify the predicted attack class and detection status specifying whether it is normal or an attack)
45. ***Output***
46. Class labels for the training data
47. Attack class and detection for real-time input data

4.2 Recurrent Neural Network Algorithm (RNN)

The RNN algorithm was integrated with KNN algorithm to achieve high accuracy and precision in the proposed MLBS algorithm. Furthermore, the RNN algorithm offers a sequential approach using a recurrent connection in the hidden layers which will strengthen the MLBS algorithm. The RNN algorithm is a deep neural network that is capable of being trained on a large volume of databases and still performs well on different classification problems. Moreover, the RNN algorithm is capable of duplicating hidden layers by applying the same biases and weights to the inputs each time without being interrupted [29]. RNN algorithm builds its technique by training data appropriately hence it is utilized in the integration with KNN algorithm to form MLBS algorithm to enable updating and rebuilding every time data passes.

The RNN algorithm has database created that assists the MLBS algorithm to store previous history of hidden state in hidden layers that are used to predict the output of the current iteration for accurate prediction results. The RNN algorithm is key in this research study due to its special capability of using

the previous output as input while hidden states. The RNN algorithm is easy to understand and works quite well with large datasets and it requires minimum data preparations for both categorical and numerical [29]. Furthermore, the RNN algorithm validate techniques making use of statistical tests [29, 30]. The Equation (4) and (5) below are utilized to determine the current state in the RNN.

$$\boldsymbol{q_t = f_r(q_{t-1}, h_t)} \quad (4)$$

Where, $\boldsymbol{q_t}$ represent the present state. The $\boldsymbol{f_r}$ is the integrate the present and previous state for a new state. The $\boldsymbol{q_{t-1}}$ represent input state includes the recurrence.

$$\boldsymbol{q_t = tanh(W_{hhq_{t-1}} W_{xhq_t})} \quad (5)$$

Where $\boldsymbol{W_{hhq_{t-1}}}$ represent weight at the recurrent neuron and $\boldsymbol{W_{xhq_t}}$ represent weight at input neuron.

The RNN Algorithm is presented in Algorithm 2 below:

Algorithm 2: The RNN Algorithm

1. ***Require***: Input classification $x_1, x_2, \ldots, x_T$
2. Initial hidden state h_0 // this is the hidden state
3. RNN parameters ($W_{xh1}, W_{h1h1}, W_{h1h2}, W_{xh2}, W_{h2h2}, W_{h2y}, b_{h1}, b_{h2}, b_y$) // initialization the weight matrix and the bias vector with its input respectively
4. Beginning function BeFu // the activation function
5. ***Decision Making Techniques***:
6. Attack: = 1& No Attack: =0, T: =0 // decision making equation for attack detection
7. **For all** $t = 1$ **to** T **do** // attack detection loop
8. hidden states
9. $h1_0$ & $h2_0$ ← initialized hidden state h_0 // first and second RNN layer hidden state
10. **End for**
11. ***Execute the beginning of the first RNN layer***
12. **If** $a1_t = W_{xh1}.x_t + W_{h1h1}.h1_{t-1} + b_{h1}$ then // the first RNN layer activation function
13. Apply the beginning function BeFu to the beginning
14. Get $a1_t$ ← attain the hidden state $h1_t =$ BeFu $(a1_t)$
15. Execute the beginning of the second RNN layer
16. **Else if** $a2_t = W_{h1h1}.h1_t + W_{xh2}.x_t + W_{h2h2}.\ h2_{t-1} + b_{h2}$ then // the second RNN layer activation function
17. Apply the beginning function BeFu to the beginning
18. Get $a2_t$ ← attain the hidden state $h2_t =$ BeFu $(a2_t)$
19. **End if**
20. **End if**
21. **Return** Output of the RNN at time step t $y_t = W_{h2y}.h2_t + b_y$ to predict whether incoming traffic is a DDoS attack or not

4.3 The design of MLBS algorithm

The MLBS algorithm is developed by integrating KNN and RNN algorithms to improve detection and prevention of DDoS attacks in IoT networks. This combination is applied in a sequential pipeline to benefit from the temporal and static characteristics of the IoT network traffic. At first the KNN is utilized to measure the similarities between known attacks and incoming network traffic patterns in the

IoT network environment. In this case the KNN acts as the feature enhancement layer that provides distance-based results and class labels output. These outputs are integrated with the original network traffic features to create strong and more informative indicators that improve potential DDoS characteristics.

The output of the IoT network traffic is fed into the RNN which focuses on learning the temporal and sequential behaviour and analyses how the network traffic is evolving over time. This integration significantly provides network traffic resembles through the KNN as well as the network traffic behave over time through RNN.

The MLBS algorithm steps are presented in Algorithm 3 are shown below:

Algorithm 3: MLBS Algorithm

1. ***Input:***
2. Number of Layers L // Initialization
3. Real-time input data: $X_{realtime}$
4. Input classification $x_1, x_2, \dots, x_T$
5. Initial hidden state h_0 // this is the hidden state
6. RNN parameters ($W_{xh1}, W_{h1h1}, W_{h1h2}, W_{xh2}, W_{h2h2}, W_{h2y}, b_{h1}, b_{h2}, b_y$) // initialization the weight matrix and the bias vector with its input respectively
7. Beginning function BeFu // the activation function
8. Training dataset $X = \{X_1, X_2, \dots, XM\}$ // Examine traffic pattens
9. **Set**: $s = 0, t = 0$ // Initialization
10. **For** $n = 1\ to\ N$ **do**
11. Partition training set into $X_n\ P_1 and\ X_n\ P_2$
12. Determine $Ln(a)$
13. **While** $P_t < hr$ **do**
14. $t = t + 1$
15. Get new data $\{Xnt\}$ and compute $\{Xnt\}$
16. $Pnt = \max\{Pnt + Xnt, 0\}$
17. $P_t = \sum Nn = 1\ Pnt$
18. **If** $P_t >= hr\ then$
19. DeclareAttack($T = t$)
20. **End if**
21. **End while**
22. **End For**
23. ***Classification Algorithm for Offline Attack***
24. **For** $1 \in [1, L]$ **do**:
25. Set parameters: // Initialization
26. Weight matrix: $W1 = 0$
27. Bias vector: $b1 = 0$
28. Weight matrix for decryption: $W1 = 0$
29. Bias vector for decryption: $b1 = 0$
30. ***Encryption layer:***
31. Calculate encoding or hidden representation using equation (2): //Instantaneous attack evidence
32. $hn = f\theta(xn) = \sigma(Wxn + b)$
33. $h1 = S\ (W1xl1 + b1)$
34. ***Decryption layer***
35. **While** not loss = stopping criteria **do**:
36. Compute yi using equation (3): // triggers an alarm when threshold is met at first time T
37. $yi = g\theta(hn) = \sigma(Whn + b)$
38. Calculate the loss function: binary cross-entropy
39. Update layer parameters $\theta = \{W, b\}$
40. **End while**
41. **End for**
42. ***Decision Making Techniques:***
43. Attack: = 1 & No Attack: = 0, $T := 0$ decision making equation for attack detection
44. **For all** $t = 1$ to T do // attack detection loop
45. hidden states
46. $h1_0$ & $h2_0$ ← initialized hidden state h_0 // first and second RNN layer hidden state
47. **End for**
48. Execute the beginning of the first RNN layer

49. **If** $a1_t = W_{xh1}.x_t + W_{h1h1}.h1_{t-1} + b_{h1}$ then // the first RNN layer activation function
50. Apply the beginning function BeFu to the beginning
51. Get $a1_t \leftarrow$ attain the hidden state $h1_t$ = BeFu$(a1_t)$
52. **End if**
53. ***Classifier***
54. Fully connected neural network with soft-max activation at the output layer
55. Set $(Wl+1, bl+1)$at the supervised layer
56. Execute the labels for each sample xn of the training dataset X
57. Apply dropout and normalization batch for speeding up the calculation
58. Complete back-propagation in a supervised manner for parameters tuning of the layers, loss function categorical cross entropy
59. Specify the predicted attack class and detection status specifying whether it is normal or an attack)
60. $\text{DetectAttacks}_I \leftarrow \text{ApplyCombinedModel}$ ($(Mi, MLBSalgorithm, Threshold)$
61. Compute performance metrics (accuracy, recall, precision, F1-score,)
62. ***Output***
63. Class labels for the training data
64. Attack class and detection for real-time input data
65. The trained MLBS algorithm
66. **End**

4.4 MLBS Algorithm Flowchart

The research study integrated the KNN and RNN algorithms to optimize performance. The performance metrics such as accuracy, precision, recall, F1-score and confusion matrix are used to test the effectiveness of MLBS algorithm. Below is the flowchart of how the MLBS algorithm works in Figure 2.

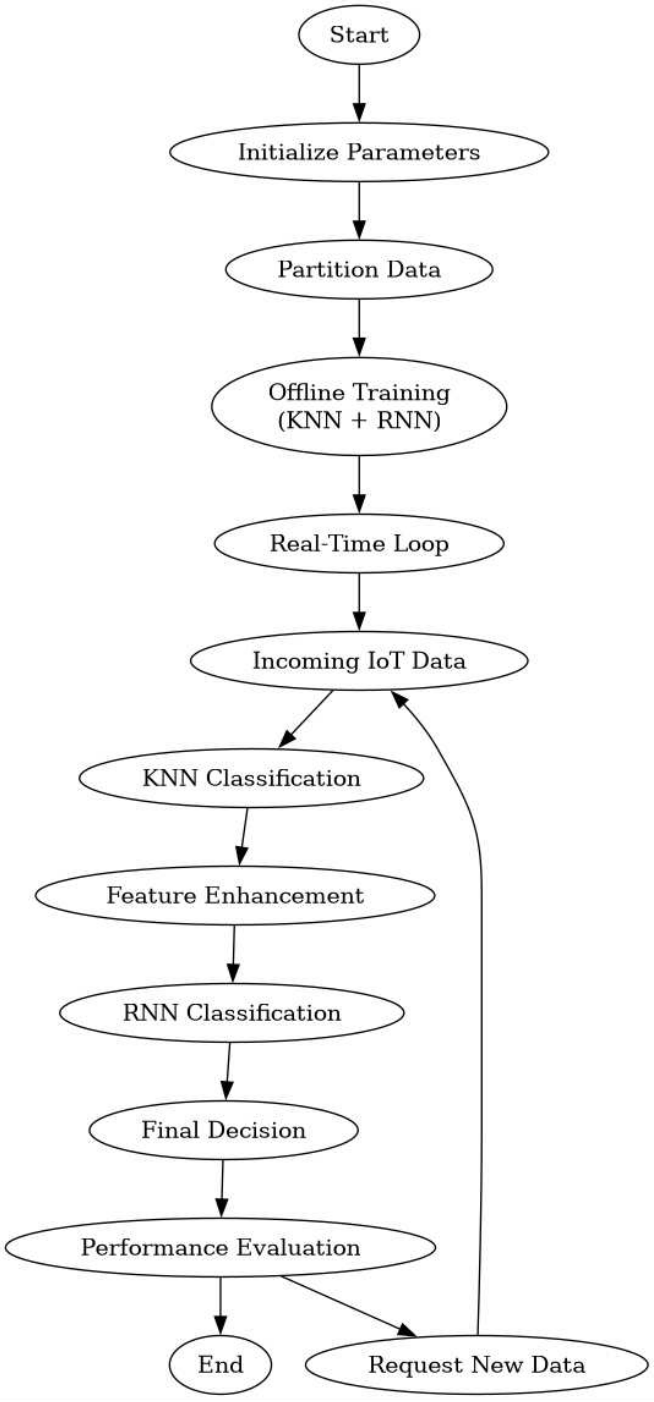


Figure 2: MLBS algorithm

The above MLBS algorithm flow chart depicts the integration of the KNN and RNN algorithms and how they work together. In the beginning of the

flowchart, we initialize parameters in the first process flow. The second process flow shows data partition setup. Then the research study presented a decision loop for the RNN and KNN algorithms. The KNN algorithm is used for the detection of quick detections and RNN algorithm will be used for deeper analysis and offline attacks. An alert will be triggered when the condition T is met. The algorithm then computes for loss function and lastly, the performance evaluation that includes Accuracy, Accuracy, Precision, Recall, F1-score, Confusion Matrix, then end loop.

5. System Implementation

This study designed the MLBS algorithm in order to detect DDoS attacks in real-time in IoT network. When malicious traffic is detected, it is stopped immediately and an alarm signal of such attack is sent to the administrator. MATLAB network simulation tool was selected to assess the performance of MLBS algorithm for different reasons. MATLAB offers solutions for high-level programming environment [31]. MATLAB R2025a version is the latest version released in May 2025 and was used in this study. MATLAB is frequently used for algorithm development, modelling and data analysis in science research and engineering. This is installed in an Intel Core i5 processor running 32GB of RAM with 500 GB hard disk.

6. Performance Evaluation and Simulation Setup

To demonstrate that the designed MLBS algorithm can improve accuracy and precision in the detection of DDoS attacks in IoT, a simulation is performed over an average of 10 independent simulation runs, ensuring consistency and reliability. The MLBS algorithm uses the IoT network traffic intrusion and public dataset called TON-IoT, to compare the results. Datasets are selected for their relevance in the intrusion detection system and complexity of attack types. Performance evaluation is conducted using metrics namely, the accuracy, precision, recall and F1-score.

6.1 Dataset Pre Processing

In the simulation process, the TON-IoT dataset undergoes pre-processing for the purpose of standardizing and improving data formats to ensure compatibility with the MLBS algorithm as well as achieving high efficiency and accuracy in detection of DDoS attacks [32]. Data cleaning was conducted for TON-IoT dataset to identify incomplete traffic records and removal of duplicate records, missing attributes as well as unwanted values. Eventually, aligned numerical format is established across all dataset attributes.

Initially, the dataset contained above 80 attributes and after the removal of fields that contained missing and zero values, the number of attributed decreased to 70 on the dataset. We used the filter called “Randomize” to randomly shuffle the dataset in order of instances, this is to ensure that when testing techniques an output accuracy is obtained. Data quality was achieved by employing “RobustScaler” to handle columns that are with numerical type of data, this helps in preventing the use of unwanted

attributes.

6.2 Feature Selection

After pre-processing and dataset conversion to the readable format, feature selection was conducted to eliminate irrelevant attributes since the selected dataset contains large number of features and up to 70 attributes. In order to refine these features, a comprehensive feature selection was conducted to reduce complexity of modelling and training time. This was conducted using the Principal Component Analysis (PCA) to enable model performance more effectively through the removal of unwanted variables without losing the quality of information.

For the MLBS algorithm to be able to differentiate between DDoS attack traffic and normal IoT traffic, PCA was used to reduce the feature to 19 from 70 attributes and different features that were initially selected. The features that were reduced were due to that they were not relevant and that they do not contribute into the proposed model's detection capabilities. The selected features are shown in Table 2.

Furthermore, Pearson's correlation approach was employed to eliminate redundant features containing nearly zero correlation to the variable target [33]. This ensures that only relevant features are used for classification by reducing the dimensionality of the space feature as well as increasing efficiency in training, improving generalization performance and reduce computational time [33]. The coefficients correlation values range from negative one (-1) to positive one (+1). This means, a value near zero indicates a weak negative correlation, no correlation is indicated by a value of 0 and a strong positive correlation is indicated by the value of 1.

Table 2: Machine Learning Features [23].

Feature	**Description**	**Correlation**
IP_Source	Device source IP address	0.127560
Port_Source	Address sent	0.779831
IP_Destination	Device destination IP address	0.148871
Port_Destination	Packets receiving address	0.348760
Protocol	Data transmission, UDP or DNS	0.162641
Duration_Flow	Duration between two packets	-1.00000
Packets_Fwd_Total	Total number of packets forward	0.888404
Packets_Backward_Total	Total number of packets backward	0.931222
Packets_Fwd_Len_Total	Total forward packets length	0.880166
Packets_Bwd_Len_Total	Total backward packets length	0.787430
Length_Fwd_Header	Forward header length	0.812806
Length_Bwd_Header	Backward header length	0.879744
Packets_Fwd_per_sec	Packets forward per	0.166998

	second	
Packets_Bwd_per_sec	Packets backward per second	0.309762
Count_SYN_Flag	Synchronous flag count	0.723789
Count_ACK_Flag	Acknowledged flag count	0.548743
Packets_Size_Average	Packet average size	0.652298
Segment_Size_Fwd_Avg	Segment average forward	0.723781
Segment_Size_Bwd_Avg	Segment average backward	0.888498

The benign and malicious are the two types of traffic labels used by the dataset. The attack prediction is conducted using the 19 features in Table 1.

6.3 Training and Testing Split

In order for the proposed model to function accordingly, training and testing split is necessary. The dataset was divided into training and testing sets using an 70/30 split employing the learn_scikit library. Where 70% was used for MLBS algorithm training and 30% used for testing. Feature normalization and dimensionality reduction techniques, such as PCA were applied before model training.

6.4 Cross Validation

In addition, cross validation was conducted to enhance reliability of the MLBS algorithm. The $K = 5$ cross validation was employed [26]. In each iteration, the first fold was used as the validation set while training is conducted in the remaining folds. This step is repeated until all other folds have served as the validation set only once. The proposed MLBS algorithm is compared against baseline models including RNN, KNN, RF, SVM, and LSTM. The performance of the MLBS algorithm will be evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion metrics.

6.5 Performance Evaluation

The performance evaluation metrics used in this paper study are accuracy, precision, recall, and F1-score. We split our dataset into train and test and used cross validation to validate our results. We ensure that our proposed MLBS model is not overfitted with training data and that our end results are not generalized. This is achieved using the k-fold cross validation. Furthermore, to ensure a fair and rigorous evaluation of the proposed MLBS model, this study explicitly considers baselining algorithms namely: RNN, KNN, RF, SVM, and LSTM that will be used as comparative benchmarking.

Although the baselining algorithms were initially introduced in the problem statement and further elaborated in the literature review, they are formally employed in this research study as the baseline classifiers against which the proposed MLBS model is evaluated. These algorithms were selected due to their frequent use in the intrusion detection research studies and known effectiveness on IoT networks security dataset [20, 21, 34, 35].

7. Evaluation Results

The performance of the MLBS algorithm was

evaluated using the accuracy, precision, recall and F1-score performance metrics. The results of the evaluation of the MLBS algorithm training for detection of DDoS attacks in IoT has shown that the proposed algorithm is reliable and working well.

Accuracy was used to determine the effectiveness of the proposed MLBS algorithm by evaluating the proportion of the classified instance's creativeness across all samples while the precision focuses on the algorithm's ability to identify DDoS attacks effectively with less false positives alarms. The recall focuses on evaluating the MLBS algorithm's capability to detect DDoS attacks traffic instances correctly while the F1-score provides the balance performance evaluation of the MLBS algorithm.

To validate the effectiveness of the proposed MLBS algorithm, comparative performance analysis was considered to compare with several baseline deep learning and ML algorithms normally used for intrusion detection in IoT environments. The baseline algorithms include KNN, RNN, RF, SVM, and LSTM. To obtain a true reflection and consistent comparison analysis the baseline algorithms were evaluated using the same TON-IoT dataset, training and testing split, feature selection and pre-processing.

The recall performance evaluation was conducted over an average of 10 independent simulation runs. The MLBS algorithm achieved the highest recall performance by achieving 98.94% recall, demonstrating its capability to detect DDoS attacks while reducing the false negative rate. The KNN and SVM algorithms showed a lower performance of 92.67% and 91.78% recall, respectively, indicating that a significant number of DDoS attacks are missing due to the inability to show reliance on complex patterns. The RF and the LSTM algorithms achieved 95.34% and 96.89% recall, respectively, showing notable performance due to improved learning capability. The MLBS algorithm still came first due to its superior detection capability, as shown in Table 3.

7.1 Accuracy

Accuracy was evaluated to demonstrate consistent improvement in the detection of DDoS attacks in an IoT environment. The MLBS algorithm achieved the highest accuracy of 98.72% in the detection of DDoS attacks in an IoT environment over the traditional learning model, namely: KNN, which achieved 94.17%, and SVM, which achieved 93.46%, demonstrating the inability to deal with complex DDoS attack patterns in IoT. The RF that came third by achieving 96.67%, and the LSTM that came second, indicating the strong performance through its ability to model dependencies in the DDoS network traffic, and achieving 97.56 behind the MLBS algorithm. The performance of the MLBS algorithm in this performance metric shows that it is more efficient and effective at classifying a large proportion of DDoS attacks in IoT networks, as shown in Figure 3.

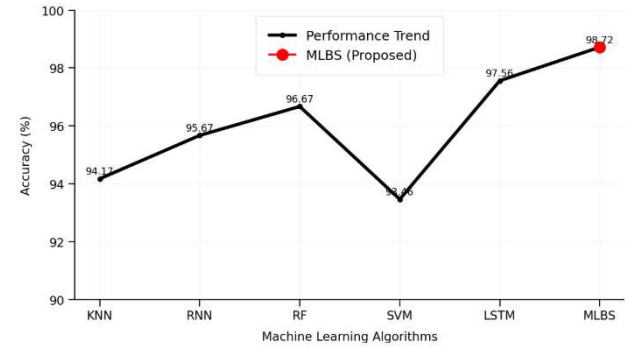


Figure 3: MLBS algorithm accuracy performance

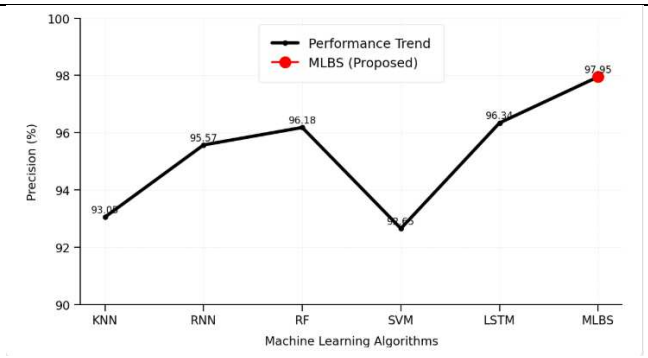


Figure 4: MLBS algorithm precision performance

7.2 Precision

To demonstrate the model's ability to correctly detect DDoS attacks while ensuring minimum false positives, precision performance was evaluated. The MLBS algorithm demonstrated the ability to detect with correctness the DDoS attacks in IoT networks by achieving the highest precision of 97.95% while ensuring the minimized false positives. The KNN and the SVM achieved 93.05% and 92.65% precision respectively showing the higher rate of incorrectly flagged DDoS attacks by benign traffic. The RF and the LSTM algorithms came closer by achieving 96.18% and 96.34% precision respectively showing better decision boundaries and feature handling, however the MLBS algorithm demonstrated its ability to accurately detect correct attack instances with minimum false positives due to its enhanced features that make it robust to reduce false positives as shown in Figure 4

7.3 Recall

The recall performance evaluation was conducted over an average of 10 independent simulation runs. The MLBS algorithm achieved the highest recall performance by achieving 98.94% recall, demonstrating its capability to detect DDoS attacks while reducing the false negative rate. The KNN and SVM algorithms showed a lower performance of 92.67% and 91.78% recall, respectively, indicating that a significant number of DDoS attacks are missing due to the inability to show reliance on complex patterns. The RF and the LSTM algorithms achieved 95.34% and 96.89% recall, respectively, showing notable performance due to improved learning capability. The MLBS algorithm still came first due to its superior detection capability, as shown in Figure 5.

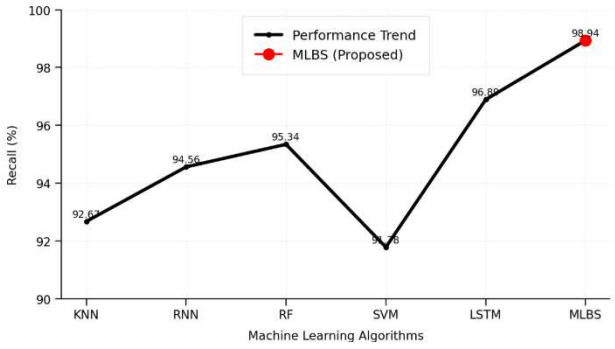


Figure 5: MLBS algorithm recall performance

7.4 F1-score

The F1-score performance was conducted over an average of 10 independent simulation runs ensuring consistency across all models by balancing precision and recall providing comprehensive view of overall effectiveness in detection. The MLBS algorithms obtained the highest performance of 98.39% F1-score, indicating the balance between DDoS attacks detection accuracy and minimized false alarms. The KNN and the SVM achieved lower F1-score of 92.04% and 92.09% respectively due to their uneven precision and recall. The RF and the LSTM algorithms achieved 94.67% and 95.75% F1-score although their balance remains inferior over that of the MLBS algorithm as shown in Figure 6.

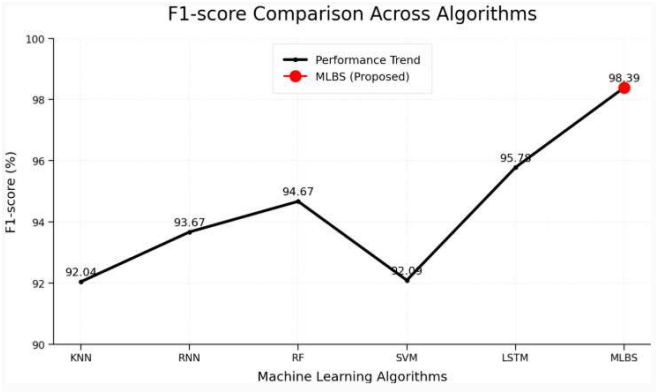


Figure 6: MLBS algorithm F1-score performance

The overall performance of the MLBS algorithm demonstrates strong performance across all metrics, which indicates that it is reliable and effective in detecting DDoS attacks within the IoT environment. The MLBS algorithm achieved 98.72% accuracy, 97.95% precision, 98.94% recall, and 98.39% F1- score which confirms it as the most reliable model. This then confirms that the integration of KNN and RNN contributes to the enhanced detection capability and improved classification accuracy.

8. Confusion matrix

The representation of the confusion matrix for the MLBS algorithm with its baseline models is shown in in Figure 7. The MLBS algorithm shows strong performance with minimal misclassification, indicating its relevance in separating between malicious IoT networks traffic and benign [36]. The hybrid MLBS algorithm low FP affirms its applicability for real-world applications due to its ability to optimize training process by diversifying feature representations as shown in Figure 7

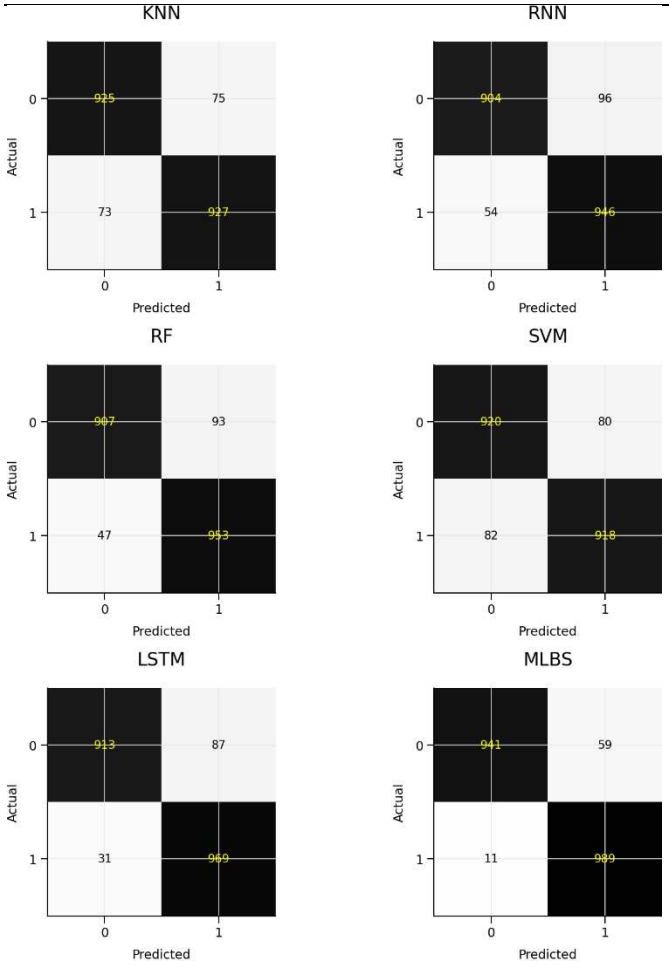


Figure 7 MLBS algorithm confusion matrix performance with baseline algorithms

a) KNN

The KNN in Figure 7 has demonstrated higher FP and FN, this is due to the complexity of the TON-IoT dataset that consists of multiple features that causes high-dimensional space because of increased FP and FN. KNN is challenged by its strong reliance to distance while the IoT network detection requires learning patterns and handling of noisy data.

b) RNN

Although RNN performed better than KNN and SVM from Figure 7 above, it still shows higher FN and FP compared to MLBS, RF, and LSTM. This is due to long-term learning dependency for earlier patterns in IoT network traffic sequences that are still missed.

c) RF

Similarly, the RF performed well but not as compared to MLBS and LSTM from Figure 7. RF shows FN and FP due to the time-dependent patterns requirement for IoT network detection. RF is not capable of capturing temporal and sequential IoT network traffic patterns as effectively as MLBS, leading of increased classification errors.

d) SVM

The SVM in Figure 7 shows a weaker performance of the FP and FN as compared to MLBS, LSTM and RF. This is due to the complexity of the samples and feature of the IoT network traffic including computational costs of large datasets, leading to a reduced classification performance.

e) LSTM

Although LSTM performed very strongly and second to MLBS in Figure 7, it still shows slightly higher FP and FN compared to MLBS. This is due to that LSTM learns temporal dependencies well but struggle with increasing variability and evolving IoT network attacks.

f) MLBS

The MLBS algorithm offers superior classification performance compared to baseline models in Figure 7 as it achieves the lowest False Positive (FP) of 3% and False Negative (FN) of 5% rates, while maintaining the highest true positive of 97% and

true negative of 98.68% counts. This results in improved the MLBS algorithm accuracy to 98.98% and F1-score (95.9%), indicating a better balance between detection capability and false alarm reduction. The confusion matrix patterns confirm that MLBS algorithm provides the most optimal balance between detection accuracy and false alarm reduction, making it the most suitable model for IoT network detection environments.

9. ROC curves

Figure 8 below demonstrates the ROC analysis of the hybrid MLBS algorithm compared to the baseline models

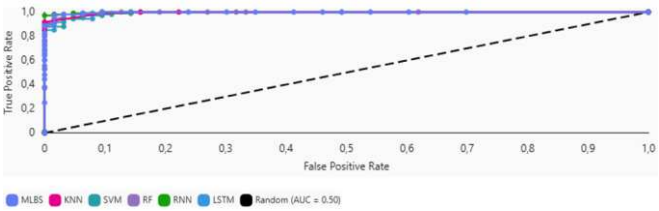


Figure 8 ROC curves: MBLS vs Baseline Models

The ROC analysis in Figure 8 is conducted to compare the performance of the hybrid MLBS algorithm against the baseline models. The MLBS algorithm achieved an AUC score of 0.999, outperforming the baseline models which obtained the following AUC scores: RF is 0.997, KNN is 0.995, LSTM is 0.993, RNN is 0.998 and SVM is 0.994. The ROC curve of the proposed hybrid MLBS algorithm shows consistent domination across all baselines demonstrating its relevance in the detection of DDoS attacks in IoT networks devices.

10. AUC scores

Figure 9 below shows the AUC score of the hybrid MLBS algorithm compared to the baseline models. The hybrid MLBS algorithm achieved the largest area under the curve and the highest ascent of 0.999 AUC score, outperforming the baseline models and conformance of it Superiores. The rest of the baseline models obtained: RF is 0.992, coming second from the hybrid MLBS model, the LSTM is 0.989, the RNN is 0.982 and the KNN is 0.970.

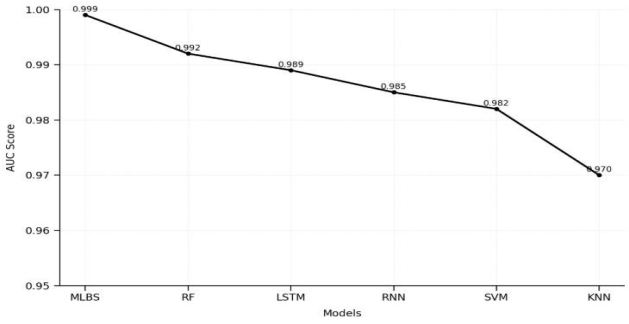


Figure 9 AUC scores: MBLS vs Baseline Models

11. Precision-Recall curves

Figure 10 below demonstrates the precision-recall analysis of the hybrid MLBS algorithm compared to the baseline models

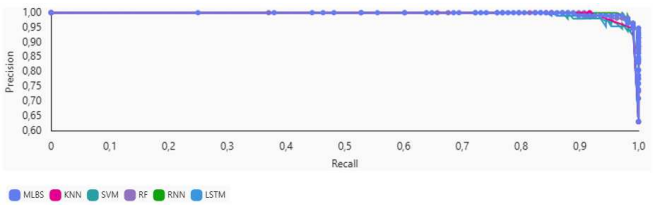


Figure 10 Precision-Recall curves: MBLS vs Baseline Models

The precision-Recall analysis in Figure 10 shows that the proposed hybrid MLBS algorithms

achieved the highest PR-AUC of about 0.997 while the baseline models achieved the lower PR-AUC, RNN achieved 0.991, RF is 0.995, RNN is 0.991, SVM is 0.988 and KNN is the lowest with 0.975. This demonstrates that the MLBS algorithm is capable of maintaining high precision across increasingly recall levels.

12. Baseline Algorithms Summary

Table 3 shows the summary of performance results of the baseline algorithms. The MLBS algorithm outperforms all other baseline algorithms across all performance metrics. The KNN obtains the lowest performance, specifically on the recall, due to its sensitivity to noisy and its inability to store temporary dependencies associated with the TON-IoT dataset.

The RNN improves detection of DDoS attacks in IoT through its quality of modelling sequential patterns, however, limited on performance if not complemented by feature-based classification.

The RF ensembles learning capability which helps in achieving strong results but fall short on temporal awareness that is required when identifying evolving DDoS attacks traffic.

The SVM has scalability issues and struggles with complex non-linear interactions in large network traffic hence moderate performance. The LSTM performs better than the RF by achieving high accuracy and precision, however, at an increased computational complexity.

The MLBS algorithm which is a result of the integration of the RNN and KNN has resolved temporal pattern learning and complemented by feature-base classification achieved 98.72% accuracy, 97.95% precision, 98.94% recall and 98.39% F1- score which is the highest across all performance metrics, proving to be robust and reliable solution for detection of DDoS attacks in IoT environment.

Table 3: Baseline Comparison summary

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	94.17	93.05	92.67	92.04
RNN	95.67	95.57	94.56	93.67
RF	96.67	96.18	95.34	94.67
SVM	93.46	92.65	91.78	92.09
LSTM	97.56	96.34	96.89	95.78
MLBS	98.72	97.95	98.94	98.39

A key observation from the results is that the proposed MLBS model achieves consistently high performance across all evaluation metrics, unlike traditional models that exhibit trade-offs between precision and recall. For example, while RF and LSTM achieve high accuracy, their performance varies slightly across precision and recall, indicating potential weaknesses in handling certain attack patterns.

The MLBS model, however, maintains uniform performance across all metrics, suggesting that the hybrid integration effectively mitigates the limitations of individual algorithms. This consistency is a strong indicator of model robustness and generalization capability.

The superior performance of the MLBS framework can be attributed to its hybrid architecture,

which integrates KNN's instance-based learning, enabling effective classification based on local data similarity. RNN's temporal modelling capability, allowing the model to capture sequential dependencies in network traffic. This combination enables the model to detect both static and time-dependent attack behaviours, which are characteristic of DDoS attacks in IoT environments. Furthermore, the hybrid approach enhances the model's ability to handle noisy data, class imbalance, and dynamic traffic patterns, which are common challenges in real-world IoT systems.

In addition, the findings indicate that the proposed MLBS model is not only accurate but also practically viable for real-world deployment. Its high precision reduces false alarms, while its high recall ensures that critical threats are not missed. This makes the model particularly suitable for real-time intrusion detection systems in resource-constrained IoT environments, where both efficiency and reliability are essential.

13. MLBS algorithm performance

The technical performance of the MLBS algorithm is described in this section. The integration of RNN and KNN algorithms enhance interpretation ability and generalization [37]. The RNN are effective at modelling data sequentially and uses this capability to contribute to the performance of MLBS algorithms to detect temporal learning [37]. Furthermore, RNN algorithm uses it capability to record temporal learning and contextual data in the MLBS algorithm thereby assisting with high precision and enhanced sequence classification.

The KNN algorithm contributes to the performance of the MLBS algorithm by accurately selection of K number and optimizing distance parameter to detect any forms of malicious IoT network attacks and efficiently differentiating DDoS attacks and normal network traffic. The KNN contributed to MLBS algorithm instance-based classification by using Euclidean distance to calculate the distances between a new instance, in our case, is the network traffic instance and the rest of the instances in the training datasets [38]. To obtain the best performance of the MLBS algorithm, the KNN number of neighbors is set to 10, Euclidean distance and uniform weighting. The hybridization of the RNN record temporally learning while the Euclidean distance is set to 10 neighbors to improved classification.

Furthermore, the MLBS algorithm improved the F1-score. F1-score is the balanced instrument of precision and recall and is useful in class imbalance. In addition, the MLBS algorithm achieved a high F1-score meaning that the model has high recall and high precision confirming classification of malicious network traffic attacks correctly while reducing false positive alarms outperforming the LSTM and RF models. The baseline comparison presented in Table 3 confirms the statistical significance of the proposed MLBS algorithm. Finally, the consistent performance of the MLBS algorithm across difference network attack classes in TON-IoT datasets suggests potential for generalization, although validation on external datasets is

necessary.

14. Computational Complexity Analysis

Although the hybrid MLBS framework improves detection accuracy, the integration of KNN and RNN increases computational overhead during training due to their sequential integration. However, the model maintains acceptable inference latency suitable for near real-time IoT environments. KNN is a lazy learner and is compute distance between a source point and all instances stored, resulting in complexity making KNN less effective as IoT network traffic increase. Since KNN retains all training data, it requires high memory which can be an issue with resource-constrained environment and large-scale.

RNN contains most training complexity due to its backpropagation through time and sequential processing. In RNN the computational cost increases with hidden layer and the length of the sequence thereby making causing resource intensive. However, once the training is started RNN inference is effective and suitable of capturing temporal patterns in IoT network traffic.

The hybrid MLBS can handle both KNN distance examination and RNN forward passes which can strain scalability in increasing datasets and large IoT network environments. Nonetheless, the MLBS algorithm can still achieve real-time performance through optimization, enabling the framework to remain practical for large-scale IoT network deployments. Table 4 is the summary of the complexity analysis.

Table 4: Complexity analysis summary

Model	**Training Cost**	**Inference Cost**	**Memory Usage**	**Temporal Learning**	**Scalability**
KNN	Low	High	High	No	Limited
RNN	High	Moderate	Moderate	Yes	Moderate
Proposed MLBS	High	Moderate	High	Yes	Manageable

15. Ablation Study

This section presents the achieved outcomes of the hybrid MLBS algorithm across all performance metrics, this process will be applied to the baseline algorithms used in this study namely: KNN, RNN, RF, SVM and LSTM. From Table 3, it is evident that the proposed hybrid MLBS algorithm outperforms the baseline algorithms in the classification of attacks in IoT network devices, demonstrating outstanding results across all performance metrics, in particular accuracy, precision, recall and F1 score. Additionally, the dismantling of one component like RNN or KNN of the proposed model displays a decline in accuracy indicating the significancy of all components in contributing to the overall performance of MLBS algorithm [39]. The results demonstrate the necessity of hybrid model.

From the simulation results obtained, it is observed that the hybrid MLBS algorithm offered good results in a complex and evolving IoT traffic attacks classification. This is primarily due

simulation results of the implementation of the proposed hybrid MLBS algorithm through the RNN that classifies attacks related features based on multiple variety of attributes and KNN that stores datasets with the assumption of the similarity between existing and new detections and stores the newly identified attack detection in the attributes that is more similar to the existing, when integrated they make classification seamless.

The RNN classifies attacks by learning temporal patterns and extracting useful features from network traffic, while the KNN utilize the features to create precise predictions based on similarity to known samples. This integration provides accurate instance decisions and strong pattern learning resulting in an accurate and more robust model.

16. Real Interpretation of IoT Behavior

The real-world IoT network behavior is characterized by predictable, low-volume communication patterns, unique temporal regularities, and traffic structures that consist of specific protocols. However, malicious activities in IoT network introduce deviations in timing, packet rates, and feature interactions. Furthermore, IoT network attacks are evolving from simple high-volume disruptions also referred to as DoS attacks, to sophisticated multiple attacks referred as DDoS attacks. Effective ML models for detection of DDoS attacks in IoT network environment therefore relies on elements such as capturing temporal packet behavior, modelling complex feature interactions, and identifying deviations from established behavioral patterns. The superior performance of the MLBS algorithm suggests its ability to learn these intricate relationships, making it more effective than traditional baseline models in distinguishing between normal and malicious IoT network traffic.

17. Comparison with Existing Literature

Performance comparison with existing literature is necessary to determine the efficacy of the proposed hybrid MLBS algorithm. Recently, DDoS attacks detection and mitigation studies in the IoT-SDN environment have utilised artificial and custom datasets including SDN controller to implement their techniques [40, 41, 42], thereby using the existing related challenges to perform direct comparison. However, some studies share similar approach with the proposed hybrid MLBS algorithm focusing on the IoT network environment and using publicly available datasets [42, 43, 44]. Their models are evaluated using similar performance metrics including accuracy, precision, recall and F1-score [37, 45, 46]. The implementation environment significantly differs from various studies while the related studies deploy detection model on a resource constrained switch. In this study we strategically compare the performance of our proposed MLBS algorithm with hybrid models using similar TON-IoT datasets and two that use the IoTID20 dataset. This is to confirm that our proposed model is significantly superior to that reported in Table 5.

Table 5: Comparison with Literature

Ref	**Model**	**Datasets**	**Epochs**	**Accuracy**	**Precision**	**F1-score**	**Recall**
[36]	Hybrid RF-IF	TON-IoT	10	97.8	97.5	97.7	98.0
[47]	Hybrid CNN-LSTM	IoTID20	10	91.2	90.81	91.75	92.7
[48]	Hybrid LSTM model	IoTID20	10	93.0	90.0	93.0	92.0
Our model	MLBS	TON-IoT	10	98.72	97.95	98.39	98.94

18. Comparative Literature Discussion

The superior performance achieved by the proposed MLBS framework demonstrates the effectiveness of combining instance-based similarity learning with temporal sequence modelling for DDoS detection within IoT environments. While traditional classifiers such as SVM and KNN exhibit acceptable performance under static conditions, their inability to effectively model sequential traffic behavior limits their adaptability to dynamic IoT attack patterns. The integration of RNN enables the proposed framework to capture temporal dependencies associated with evolving DDoS traffic bursts, thereby improving recall performance and reducing false negatives. Furthermore, the hybridization strategy improves robustness against noisy and heterogeneous IoT traffic, which remains a major limitation in existing single-model intrusion detection systems. These findings align with recent studies by [49] and [18], which reported improved intrusion detection performance through deep sequential learning architectures. However, the proposed MLBS framework further extends existing work by demonstrating improved generalization capability and balanced performance across all evaluation metrics within the TON-IoT dataset.

The results in Table 3 highlight our proposed MLBS algorithm consistently outperforming the baseline models across all performance metrics. The MLBS algorithm achieved accuracy of 98.72, precision of 97.95, recall of 98.94 and F1-score of 98.39. These results surpass all baseline models including LSTM and RF which achieved the accuracy value of 97.56 and 96.67 respectively. The hybrid MLBS algorithm demonstrates a high significant value of p-value = 0.004, that further highlights its superior performance compared to the baseline models [50]. This is through a paired t-test conducted on 10-fold cross validation results showing accuracy of $t = 3.87$ and $p = 0.004$, F1-score $t = 3.21$,

p-value = 0.009), and AUC t = 4.12, p = 0.003). The 97% confidence intervals for the mean differences exclude zero highlighting its robustness of the improvements.

19. Threats to Validity

Several threats to validity apply to this study. Firstly, although the datasets used in this study are widely used on ML intrusion detection in IoT network attacks, the study is only limited to several attributes after the removal of fields containing zero attributes. In addition, the reliance to TON-IoT dataset may limit generalizability due to its predefined attacks, limited device diversity and controlled environment. Secondly, dataset representation remains a concern as the real-world IoT network attacks exhibit significantly more complexity and variability. Thirdly, the superior performance of the MLBS algorithm raises potential overfitting concerns, particularly given the possibility of learning dataset containing changing patterns. Finaly, the limited diversity of attack types further constrains the evaluation, as emerging and sophisticated attack vectors are not represented. These factors collectively suggest that while the proposed hybrid MLBS algorithm demonstrates promising results, further validation using diverse, real-world datasets is necessary to confirm its practical applicability and robustness.

20. Conclusion and Future Work

This study proposed a MLBS framework for the detection of DDoS attacks in IoT environments. The proposed hybrid model integrates KNN and RNN to enhance detection capability by combining instance-based learning with temporal pattern recognition. The performance of the MLBS framework was evaluated through a comprehensive simulation using the MATLAB environment and benchmarked against widely used machine learning and deep learning models, including KNN, RNN, RF, SVM, and LSTM. The experimental results demonstrate that the proposed model achieves superior detection performance, with an accuracy of 98.72%, precision of 97.95%, recall of 98.94%, and an F1-score of 98.39%. These results confirm that the MLBS framework is highly effective, robust, and reliable in identifying both known and previously unseen DDoS attack patterns.

Beyond classification performance, the proposed model demonstrates strong adaptability to noisy data, class imbalance, and dynamic traffic conditions, which are common challenges in real-world IoT environments. This highlights its potential as a scalable and practical intrusion detection solution, capable of enhancing the resilience and security of IoT networks.

Despite these promising results, several limitations provide opportunities for future research. First, the current study relies on a simulation-based evaluation, and future work should focus on validating the proposed framework using real-world IoT datasets and live network environments. Second, although the model achieves high detection accuracy, further investigation is required to optimize computational efficiency and resource consumption, particularly for deployment in resource-constrained IoT devices. Third, future studies should explore

the integration of lightweight deep learning models and edge computing frameworks to support real-time, distributed intrusion detection. Finally, extending the model to address a broader range of cyber threats, including multi-vector and adaptive attacks, would further enhance its applicability in evolving threat landscapes.

Overall, this study contributes to the advancement of intelligent intrusion detection systems by providing a high-performance, adaptive, and scalable solution for securing IoT networks against DDoS attacks, thereby supporting the reliable and sustainable deployment of IoT technologies.

21. References

[1] Zewdie, T.G. & Girma, A., 2020. IOT Security and the Role of AI/ML to Combat Emerging Cyber Threats in Cloud Computing Environment. Issues in Information Systems, Vol 21, No. 4, pp. 253-263, https://doi.org/10.48009/4_iis_2020_253-263

[2] Chen, Y., 2020. IoT, Cloud, Big data and AI in Interdisciplinary Domains, Simulation Modelling Practice and Theory, Vol. 102,

[3] Bobrovnikova, K. Lysenko, S. Hurman, I. Kwiecien, A., 2022. Machine Learning Based Techniques for Cyberattacks Detection in the Internet of Things Infrastructure.

[4] Alsamiri, J. & Alsubhi, K., 2019. Internet of Things Cyber Attacks Detection using Machine Learning. International Journal of Advanced Computer Science and Applications (IJICT), Vol. 10, No. 12.

[5] Mishra, N. & Pandya, S., 2021. Internet of Things Applications, Security Challenges, Attacks, Intrusion Detection, and Future Visions. IEEE Access, 9: pp59353-59377.

[6] Abhishek, G.S, 2023. Review on DDoS Attacks on IOT Devices. International Journal of Research Publication and Reviews, Vol. 4, No. 12, pp.4771–4776. https://ijrpr.com/uploads/V4ISSUE12/IJRPR20776.pdf

[7] Seyed, C., Roux, J., Ngo, B., & Kebe, M., 2023. Proposal of a Machine Learning-based Model to Optimize the Detection of Cyber-attacks in the Internet of Things. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 14, No.11, pp. 965-970, DOI:10.14569/IJACSA.2023.0141198

[8] Chandrakanth, R.B., Ramya, V.R., Zabiha, K., & Srinivas, C., 2025. Optimized Intrusion Detection for IoT Networks Using Machine Learning and Feature Selection with RTGBO-ELM Integration.

[9] Shire, R., Shiaeles, S., Bendiab, K.,Suma, V., 2021. Internet-of-Things (IoT) based Smart Agriculture in India - An Overview. Journal of ISMAC., Vol. 03, No. 1, pp1–5, DOI:10.36548/jismac.2021.1.001

[10] Sarwar, N., Alharthi, S.R., Aljohani, M., & Elhosseini, M.A., 2026. Securing IoT networks: a machine learning approach for detecting unusual traffic patterns. Sci Rep 16, 3397. https://doi.org/10.1038/s41598-025-33447-z

[11] Williams, P., Dutta, I.K., Daoud, H. & Bayoumi, M., 2022. A Survey on Security in Internet of Things with a Focus on the Impact of Emerging Technologies. Journal Homepage: www.sciencedirect.com/journal/internet-of-things

[12] Alfahaid, A., Alalwany, E., Almars, A.M., Alharbi, F., Atlam, E., & Mahgoub, I., 2025. Machine Learning-Based Security Solutions for IoT Networks: A Comprehensive Survey, 25, 3341. https://doi.org/10.3390/s25113341. International

[13] Seyed, C., Kebe, M., Arby, M.M, Mohmoud, M.B, & Seyidi, C.M.C., 2023. Cybersecurity Mechanism for Automatic Detection of IoT Intrusions Using Machine Learning. Journal of Computer Science, Vol.1, No. 20.

[14] Khedr, W.I, Gouda, A.E, & Mohamed, R.E., 2023. FMDADM: A Multi-Layer DDoS Attack Detection and Mitigation Framework Using Machine Learning for Stateful SDN-Based IoT Networks, Vol.11, No. 20.

[15] Garba, U.H., Toosi, A.N., Pasha, M.F., & Khan, S., 2024. SDN-based detection and mitigation of DDoS attacks on smart homes. Computer Communications, Vol.221, pp29–41.

[16] Ravi, N., & Shalinie, S.M., 2020. Learning-Driven Detection and Mitigation of DDoS attack in IoT via SDN-Cloud Architecture. IEEE Internet Things J., vol. 7, no. 4, pp3559–3570,doi: 10.1109/JIOT.2020.2973176.

[17] Bagaa, M., Taleb, T., Bernabe, J.B., & Skarmeta, A., 2020. A Machine Learning Security Framework for IoT systems,'' IEEE Access, vol. 8, pp114066–114077, 2020, doi: 10.1109/ACCESS.2020.2996214

[18] Ravi, V., Chaganti, R., & Alazab, M. (2022). Deep learning feature fusion approach for intrusion detection system in SDN-based IoT networks. IEEE Internet of Things Magazine, Vol. 5, No. 4, pp166-170

[19] Kumar, N. & Sharma, S., 2023. A Hybrid Modified Deep Learning Architecture for Intrusion Detection System with Optimal Feature Selection. Electronics 2023, Vol. 12, No. 19, pp4050. https://doi.org/10.3390/electronics12194050

[20] Abdallah, M., An Le Khae, N., Jahromi, H., & Jurcut, A.D., 2021. A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs. The 16th International Conference on Availability, Reliability and Security,

https://doi.org/10.1145/3465481.3469190

[21] ElSayed, S.M., Le-Khac, N.A., Albahar, M.A., & Jurcut, A., 2021. A novel hybrid model for intrusion detection systems in SDNs based on CNN and a new regularization technique. Journal of Network and Computer Applications, https://doi.org/10.1016/j.jnca.2021.103160

[22] Polat, O., Durmus, O., Dogan, F., Turkoglu, M., Seker, H., Atasoy, F., & Algiil, E., 2026. Supervised and deep learning techniques for DDoS detection in software-defined network architecture: a systematic review. Engineering Science and Technology, an International Journal, DOI: 10.1016/j.jestch.2026.102290

[23] Algus, E., Dogan, F., Ahmad, A.A., & Polat, O., 2026. SCADANet: A novel dataset for SCADA cybersecurity and intrusion detection, Computer Networks, Vol.278, No. 112087, https://doi.org/10.1016/j.comnet.2026.112087

[24] Abed, A., 2016. Internet of Things (IoT): Architecture and design. Conference Paper, DOI: 10.1109/AIC-MITCSA.2016.7759958.

[25] Stomeo, C., Ciancarini, P., Presutti, V., & Alam, M., 2018. Applying Machine Learning to Cyber Security. School of Science Second degree in computer science, Vol., 14. Issue No. 11.

[26] Mohammed, S., 2021. A Machine Learning-Based Intrusion Detection of DDoS Attack on IoT Devices. International Journal of Advanced Trends in Computer Science and Engineering, Vol, 10. No. 04.

[27] Samantaray, M., Barik, R.C., & Biswal, A.K., 2024. A comparative assessment of machine learning algorithms in the IoT-based network intrusion detection systems. Decision Analytics Journal Vol.11, No. 4, pp315-335. https://doi.org/10.1016/j.dajour.2024.100478

[28] Jiang, Y., 2024. oT Device Identification Based on IP Traffic. Journal of Computing and Electronic Information Management Vol 15. No. 3, pp315-335.

[29] Yadav, S., Hashmi, H., Vekariya, D., Khan, Z.A.N., & Fidelis, V.J., 2024. Mitigation of attacks via improved network security in IOT network environment using RNN. journal homepage: www.sciencedirect.com/journal/measurement-sensors, Vol. 32, No. 101046.

[30] Hariprasad, S., Deepa, T., & Bharathiraja, N., 2021. Detection of DDoS Attack in IoT Networks Using Sample Selected RNN-ELM. Intelligent Automation & Soft Computing, DOI:10.32604/iasc.2022.022856.

[31] Chukwunweike, JN., Eze, C.C., Izekor, L.O., Abubakar, I., & Adeniran, I., 2024. Integrating deep

learning, MATLAB, and advanced CAD for predictive root cause analysis in PLC systems: A multi-tool approach to enhancing industrial automation and reliability. World Journal of Advanced Research and Reviews. Vol 23, No. 02. Pp2538-2557

[32] Hussain, M.Z., Hasan, M.Z., Maqbool, K.Q., Nazir, A., & Fard, H.M.F., 2025. DDoS Attack Detection System Using Machine Learning Techniques. IRACST-International UCP Journal of Engineering and Technology. Vol.02, No 2.

[33] Sakr, H.A., Fouda, M.M., Ashour, A.F., Abdelhafeez, A., El-Afifi, M.I., & Abdeler, M.R., 2024. Machine learning-based detection of DDoS attacks on IoT devices on multi-energy systems. Egyptian Informatics Journal 28,

[34] Jarjis, A.H., Alzubaidi, N.Y.S., & Pehlivanoglu, M.K., 2023. Cyber Attacks Classification on Enriching IoT Datasets. EAI Endorsed Transactions on Internet of Things, Vol. 9, No. 3, e2.

[35] Rafique, S.H., Abdallah, A., Musa, N.S., & Murugan, T., 2024. Machine Learning and Deep Learning Techniques for Internet of Things Network Anomaly Detection - Current Research Trends, Vol. 24, pp1968, https://doi.org/10.3390/s24061968

[36] Beatrice, N., David, M., & Anyembe, A., 2025. An Enhanced Hybrid Machine Learning Model for Detecting DoS Attacks in IoT Network. Indian Journal of Computer Science and Technology, Vol. 7, No. 4, pp. 3559–3570, https://www.doi.org/10.59256/indjcst.20250403024.

[37] Ali, M., Shahroz, M., Mushtaq, M.F., Alfarhood, S., Safran, M., & Ashraf, I., 2024. Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment, IEEE Access, Vol. 12, pp. 40682–40699. DOI: 10.1109/ACCESS.2024.3376400

[38] Sama, N.U., Ullah, S., Kazmi, S.M.A., & Mazzara, M., 2024. Cutting-Edge Intrusion Detection in IoT Networks: A Focus on Ensemble Models, IEEE Access., Vol. 13, No. 16, pp. 8375–8392.

[39] Binbusayyis, A., 2025. Innovative Defense: Deep Learning-Powered Intrusion Detection for IoT Networks. IEEE Access, Vol.13, pp. 31105–31119.

[40] Vishnu V.M., & Manjunath, P., 2023. ProSD-EdgeIoT: Protected cluster assisted SDWSN for tetrad edge-IoT by collaborative DDoS detection and mitigation, Cyber-Phys. Syst, Vol. 9, No. 2, pp. 144–173

[41] Yang, Y., Wang, J., Zhai, B., & J. Liu, J., 2019. IoT-based DDoS attack detection and mitigation using the edge of SDN, in Proc. 11th Int. Symp.

Cyberspace Saf. Secur, Guangzhou, China. Cham, Switzerland: Springer, pp. 3–17

[42] Chauhan, P., & Atulkar, M., 2023. A framework for DDoS attack detection in SDN-based IoT using hybrid classifier, in Proc. 3rd Int. Conf. Mach. Learn., Image Process., Netw. Secur. Data Sci. Cham, Switzerland: Springer, pp. 889–900.

[43] Cheng, H., Liu, J., Xu, T., Ren, Mao, B.J., & Zhang, W., 2020. Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks. Int. J. Sensor Netw, Vol. 34, No. 1, p. 56

[44] Zang, M., Zaballa, E.O., & Dittmann, L., 2022. SDN-based in-band DDoS detection using ensemble learning algorithm on IoT edge, in Proc. 25th Conf. Innov. Clouds, Internet Netw. ICIN, pp. 111–115.

[45] Alzahrani, M.Y., & Bamhdi, A.M, 2022. Hybrid deep-learning model to detect botnet attacks over Internet of Things environments, Soft Comput., Vol. 26, No. 16, pp. 7721–7735.

[46] Belachew, H.M., Beyene, M.Y., Desta, A.B., Alemu, B.T., Musa, S.S, & Muhammed, A.J., 2024. Design a Robust DDoS Attack Detection and Mitigation Scheme in SDN-Edge-IoT by Leveraging Machine Learning. IEEE Access. Vol. 13, No. 16, pp. 10194–10214.

[47] Sahu, A.k., Sharma, S., Tanveer., & Raja, R., 2021. Internet of Things attack detection using hybrid Deep Learning Model, Computer Communications, Vol 176, pp. 146–154.

[48] Yadav T., & Kumar, D., 2025. Efficient IDS System Using Hybrid Machine Learning Mechanism in IoT, Proceedings of the 6th International Conference on Deep Learning, Artificial Intelligence and Robotics (ICDLAIR 2024), Advances in Intelligent Systems Research 193, https://doi.org/10.2991/978-94-6463-740-3_20

[49] Abdallah, M., An Le Khae, N., Jahromi, H., & Jurcut, A.D., 2021. A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs. The 16th International Conference on Availability, Reliability and Security, https://doi.org/10.1145/3465481.3469190

[50] Bensaoud, A, & Kalita, J., 2025. Optimized detection of cyber-attacks on IoT networks via hybrid deep learning models. Ad Hoc Networks, Vol 170. No. 103770,