Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.065260

ARTICLE

Renovated Random Attribute-Based Fennec Fox Optimized Deep Learning Framework in Low-Rate DoS Attack Detection in IoT

Prasanalakshmi Balaji[1,2], Sangita Babu[3], Maode Ma[4], Zhaoxi Fang[2], Syarifah Bahiyah Rahayu[5,6,*], Mariyam Aysha Bivi[1] and Mahaveerakannan Renganathan[7]

[1]Department of Computer Science, College of Computer Science, King Khalid University, Abha, 61421, Saudi Arabia
[2]Institute of Artificial Intelligence, Shaoxing University, Shaoxing, 312000, China
[3]Department of Computer Science, King Khalid University, Rijal Alma, 61421, Saudi Arabia
[4]KINDI Center for Computing Research, College of Engineering, Qatar University, Doha, 122104, Qatar
[5]Faculty of Defence Science and Technology, National Defence University of Malaysia (UPNM), Kuala Lumpur, 57000, Malaysia
[6]Cyber Security & Digital Industrial Revolution Centre, National Defense University of Malaysia (UPNM), Kuala Lumpur, 57000, Malaysia
[7]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, 602117, India

*Corresponding Author: Syarifah Bahiyah Rahayu. Email: syarifahbahiyah@upnm.edu.my

Received: 08 March 2025; Accepted: 24 June 2025; Published: 30 July 2025

ABSTRACT: The rapid progression of the Internet of Things (IoT) technology enables its application across various sectors. However, IoT devices typically acquire inadequate computing power and user interfaces, making them susceptible to security threats. One significant risk to cloud networks is Distributed Denial-of-Service (DoS) attacks, where attackers aim to overcome a target system with excessive data and requests. Among these, low-rate DoS (LR-DoS) attacks present a particular challenge to detection. By sending bursts of attacks at irregular intervals, LR-DoS significantly degrades the targeted system's Quality of Service (QoS). The low-rate nature of these attacks confuses their detection, as they frequently trigger congestion control mechanisms, leading to significant instability in IoT systems. Therefore, to detect the LR-DoS attack, an innovative deep-learning model has been developed for this research work. The standard dataset is utilized to collect the required data. Further, the deep feature extraction process is executed using the Residual Autoencoder with Sparse Attention (ResAE-SA), which helps derive the significant feature required for detection. Ultimately, the Adaptive Dense Recurrent Neural Network (ADRNN) is implemented to detect LR-DoS effectively. To enhance the detection process, the parameters present in the ADRNN are optimized using the Renovated Random Attribute-based Fennec Fox Optimization (RRA-FFA). The proposed optimization reduces the False Discovery Rate and False Positive Rate, maximizing the Matthews Correlation Coefficient from 23, 70.8, 76.2, 84.28 in Dataset 1 and 70.28, 73.8, 74.1, 82.6 in Dataset 2 on EPC-ADRNN, DPO-ADRNN, GTO-ADRNN, FFA-ADRNN respectively to 95.8 on Dataset 1 and 91.7 on Dataset 2 in proposed model. At batch size 4, the accuracy of the designed RRA-FFA-ADRNN model progressed by 9.2% to GTO-ADRNN, 11.6% to EFC-ADRNN, 10.9% to DPO-ADRNN, and 4% to FFA-ADRNN for Dataset 1. The accuracy of the proposed RRA-FFA-ADRNN is boosted by 12.9%, 9.09%, 11.6%, and 10.9% over FFCNN, SVM, RNN, and DRNN, using Dataset 2, showing a better improvement in accuracy with that of the proposed RRA-FFA-ADRNN model with 95.7% using Dataset 1 and 94.1% with Dataset 2, which is better than the existing baseline models.

KEYWORDS: Detecting low-rate DoS attacks; adaptive dense recurrent neural network; residual autoencoder with sparse attention; renovated random attribute-based fennec fox optimization

1 Introduction

The use of IoT devices is increasing, and network attacks are becoming more frequent. IoT technology offers solutions that operate autonomously, enabling the development of intelligent systems capable of monitoring real-time applications [1]. IoT devices consist of multiple layers, and interconnecting these layers presents a significant challenge [2]. However, IoT devices remain susceptible to security threats, particularly DoS attacks, which disrupt interaction among IoT users [3]. DoS attacks on a network lead to excessive traffic, thereby interrupting its normal operations. This results in legitimate users being unable to access services, which causes substantial damage, especially in cloud systems where multiple services are interdependent [4]. Detecting LR-DoS attacks in cloud computing settings is particularly challenging due to its dynamic nature, the diverse range of potential attack vectors, and the necessity to maintain accuracy during detection while minimizing the impact on legitimate traffic [5]. LR-DoS attacks lead to invalid synchronous data, ultimately exhausting their available resources and hindering their ability to respond to legitimate requests.

Conventional detection methods for DoS attacks often struggle to identify LR-DoS attacks due to their low-volume and inconspicuous nature [6]. Consequently, there is an urgent need for sophisticated detection frameworks capable of effectively recognizing and mitigating these threats in real time. LR-DoS attacks frequently duplicate traffic patterns, which complicates the ability of traditional detection systems during DoS attack detection [7]. Numerous IoT devices and networks function with limited computational and memory resources, which diminishes the detection accuracy [8]. The dynamic and diverse nature of IoT networks causes fluctuations in traffic patterns, making it difficult to establish baseline behaviors necessary for actual anomaly detection [9]. Conventional techniques struggle to recognize new or unknown attack patterns, as they are mainly designed to identify only the known threats [10]. By linking the utilization of deep learning, the proposed framework seeks to boost the accuracy and efficiency of LR-DoS attack detection, thereby sustaining the overall security of IoT networks [11].

Deep learning models can identify intricate patterns within data, facilitating more precise detection of LR-DoS attacks, even when these attacks closely resemble authentic traffic [12]. These models also adapt to changing attack schemes and emerging threats, making them particularly effective in dynamic environments such as IoT networks [13]. Deep learning models enhance the accuracy of attack detection, leading to a lessening of false positives and minimizing interruptions to legitimate traffic [14]. Furthermore, deep learning-based detection systems are integrated with other security measures, bolstering the overall security framework and providing a multi-layered defense against potential attacks [15]. Additionally, many deep learning systems are designed for constant learning, allowing them to refine their detection capabilities over time as they encounter new traffic patterns and attack scenarios [16]. This framework highlights the dynamic network traffic analysis while incorporating feature extraction methods to extract critical patterns associated with LR-DoS attacks [17]. The proposed model effectively learns and adjusts to changing attack strategies by utilizing the temporal dependencies present in network traffic data. This cutting-edge approach aims to deliver a robust solution for protecting the IoT environment against the rising threat of LR-DoS attacks, confirming the integrity and availability of essential services. Dong et al. [18] introduced deep reinforcement learning on abnormal traffic flow detection.

Addressing the overall challenges from existing systems, the need for an innovative deep learning-based detection framework plays a pivot role:

- Stronger security measures are required because IoT devices are growing across businesses, making them a prime target for hackers.
- LR-DoS attacks are dynamic, have low traffic volumes, and can imitate legal traffic, and they are difficult for traditional DoS detection techniques to detect.

- Existing security solutions are less effective because many IoT devices have low memory and processing capability.
- IoT networks are so dynamic that it is essential to create security systems constantly learning and adjusting to new and changing attack techniques.
- Deep learning improves attack detection accuracy while reducing false positives and makes it possible to identify intricate traffic patterns.
- IoT security can be improved by integrating a deep learning-based detection system with multi-layered protection tactics.
- Maintaining continuous access to vital services in IoT contexts requires defense against LR-DoS assaults, especially for cloud-based applications where service interruptions may have far-reaching effects.

The main contribution of this proposed scheme is provided in the below points.

- An improved deep learning-based detection model is designed to achieve high accuracy in LR-DoS attack detection. This proposed model aims to reduce false negatives and ensure reliable detection of LR-DoS attacks by employing advanced methods and feature extraction. Additionally, it has been developed to enable the real-time detection of LR-DoS attacks, facilitating quick responses and mitigation strategies. This capability is essential for preserving the integrity and availability of services within IoT networks.
- The proposed model utilizes ResAE-SA for feature extraction. This method transforms raw data into significant features, thereby boosting its capability to detect patterns related to LR-DoS attacks. By incorporating sparse attention, this model highlights essential features while minimizing noise, resulting in a more efficient and compelling illustration of the data.
- RRA-FFA strategy, enables dynamic adjustments within the optimization process. This flexibility enhances the model's responsiveness to data variations, improving its accuracy in classifying LR-DoS attacks. Additionally, the optimization process facilitates more efficient utilization of computational resources by fine-tuning the ADRNN parameters. As a result, this leads to quicker processing times and reduced resource consumption.
- ADRNN, which was specifically designed to process sequential data, makes it highly effective for analyzing time-series data like network traffic. This feature enables the model to recognize temporal dependencies and patterns vital for detecting LR-DoS attacks. By tuning the attributes of ADRNN, the proposed model attains high classification accuracy.

The overall layout of this proposed model is described in the points below. A brief outline of LR-DoS attacks in IoT is provided in Section 1. Reviews of existing methods for LR-DoS attack detection are provided in Section 2. Section 3 provides the description of datasets and the proposed model's architecture. The explanations of the feature extraction process using attention mechanisms are provided in Section 4. The objective function for the training process to improve detection accuracy is provided in Section 5. The experimental results are arranged in Section 6. Finally, the key contributions and future work are provided in Section 8.

2 Literature Survey

Low-Rate Denial-of-Service (LR-DoS) attacks are growing and are becoming a nightmare despite equal contributions to their detection. Modified versions of deep learning architectures, such as Feedforward–Convolutional Neural Networks (FCNN), Bidirectional Long Short-Term Memory (Bi-LSTM), Recurrent Neural Networks (RNN), and Deep Neural Networks (DNN), contributed to the improvement in performance metrics towards the detection of LR-DoS attack. Optimization techniques like attention mechanisms, federated learning, autoencoders, and dropout algorithms to enhance feature extraction, reduce false alarms,

and ensure real-time classification have greatly supported this aspect. Furthermore, hybrid models that maximize deep learning frameworks have shown encouraging results in various network settings, including cloud computing and the Internet of Things. Notwithstanding these developments, there are still issues, like high processing requirements, the requirement for sizable labeled datasets, and concerns about managing the time-domain nature of LR-DoS attacks. This section examines significant advancements in LR-DoS detection, emphasizing optimization strategies, deep learning architectures, and their effects on enhancing detection performance while filling current research gaps.

2.1 Related Works

In 2022, Ilango et al. [19] proposed a feedforward-convolutional neural Network (FCNN) that achieved remarkable detection accuracy. A high level of accuracy was essential for effectively differentiating between benign and malicious traffic. Notably, this model could detect LR-DoS attacks using the features extracted from the network. This efficiency was crucial for real-time monitoring and response in network security. Furthermore, the design of FFCNN focuses on achieving a very low false alarm rate, which is vital for preventing the misclassification of legitimate traffic as malicious, thus enhancing the overall reliability of the detection system.

In 2020, Díaz et al. [20] presented a Support Vector Machine (SVM) method to improve the detection capabilities and provide guidelines for implementing effective mitigation strategies against intrusions. This adaptability was essential for responding to emerging threats and incorporating new technologies. Additionally, this architecture was built to integrate different machine learning models and support deployment across various environments, including large-scale networks and data centers.

In 2023, Liu et al. [21] presented an innovative data preprocessing technique focused on optimizing data utilization, thereby improving feature extraction for detecting LDoS attacks. The local model utilized Bidirectional Long Short-Term Memory (Bi-LSTM) networks integrated with a mechanism of attention for the classification process. This model aimed to reduce the effects of noise in the data while preserving temporal dependencies during LDoS attack detection. This framework facilitates high classification accuracy while ensuring data remains decentralized and minimizing time complexity.

In 2023, Fayoumi et al. [22] introduced an intelligent lightweight detection scheme, specifically the Decision Tree Classifier (DTC) model. This scheme was designed to be practical for resource-constrained IoT devices, enhancing their security. This strategy seeks high detection accuracy without sacrificing efficiency, making it well-suited for environments with limited resources.

In 2023, Pasha et al. [23] developed an artificial intelligence-enabled LR-DoS attack detection framework designed explicitly for identifying LR-DoS attacks within cloud computing environments. This framework employed deep autoencoders and dropout methods to enhance detection capabilities and lessen the impact of these attacks on cloud services.

In 2022, Fu et al. [24] implemented a Deep Neural Network (DNN) for processing original traffic input to generate detection results, thereby improving the detection process by utilizing real network traffic as its foundation. In contrast to traditional detection methods that necessitated extensive feature extraction from numerous packets, the proposed model reduced resource consumption by focusing on fundamental statistical features, which enhanced its efficiency for real-time applications.

In 2020, Tang et al. [25] investigated the detection method for LR-DoS attacks integrating two-step cluster analysis. This innovative approach successfully identifies clusters of network traffic affected by LR-DoS attacks by utilizing the distinct features of TCP traffic alongside the stability of traffic during periods of congestion. The findings revealed that the proposed detection method accurately identifies LR-DoS

attacks while maintaining a low false positive rate, highlighting their potential for real-world applications in network security.

In 2024, Yuvaraja et al. [26] suggested an innovative technique for identifying the LR-DoS and DoS attacks, which were achieved using Recurrent Neural Networks (RNN). This integrated approach facilitated prompt categorization of attacks in near real-time, thereby significantly reducing the potential impact on affected systems. The features and challenges highlighted in the existing deep-learning based LR-DOS detection models are summarized in Table 1.

Table 1: Features and challenges of existing deep learning-based LR-DoS detection model

Authors	Methodology	Features	Challenges
Ilango et al. [19]	FFCNN	It only utilizes the seven-network flow feature to achieve accurate results in the IoT environment. Identifying the low attack takes a minimum of time.	It requires a large, labeled dataset, which is more expensive to collect. Evaluating the long-term features is also tricky.
Díaz et al. [20]	SVM	It helps to select the explicit pattern by learning the corresponding traffic pattern for the detection process.	It is unsuitable for small-scale deployment, and no timely results have been found.
Liu et al. [21]	Bi-LSTM	It minimizes the overall communication rounds. It achieves high performance based on the preprocessing and federated learning process.	It needs more memory and computational resources.
Fayoumi et al. [22]	DTC	It can predict individual communication traffic. It enhances the accuracy of performance in detecting the attack.	It is more sensitive to minute variations in input data.
Pasha et al. [23]	HA-LRDD	It can handle tasks such as normalization, feature extraction, and classification. Training the model takes minimum time.	It has interpretability challenges. It takes more training time.
Fu et al. [24]	Deep neural network	It effectively analyses the potential time-frequency domain. It automatically extracts the data feature.	More training samples are required to complete the modeling.
Tang et al. [25]	BIRCH	It helps to compress the data size effectively.	It includes more arbitrary decisions.
Yuvaraja et al. [26]	RNN	It enables effective mitigation and timely responses. It provides a remarkable improvement in network security.	It has an exploding issue. It is complex to train the RNN model for challenging tasks.

(Continued)

Table 1 (continued)

Authors	Methodology	Features	Challenges
Rostami et al. [27]	Transformers & LSTM	The results highlight a significant improvement in estimation accuracy and system performance, validating the robustness of the proposed method.	With the development of the EKF and UKF estimation theory, nonlinear conditions should be considered to model the DC microgrid process more accurately.

2.2 Problem Statement

IoT network nodes are generally subjected to multiple attacks, deeply affecting integrity, availability, and confidentiality. LR-DoS is a complex DoS attack type that affects the computing resources on the server. It is a kind of attack with heavy time-domain characteristics in IoT. Various approaches have been developed to detect LR-DoS in existing literature. Still, some complex issues need to be solved, which are listed below.

- Although the existing hybrid model performs better, it requires all the network flow features to detect the attack. Therefore, better feature extraction is needed to compute the feature before the flow is benign or malicious. Hence, this work implements the attention mechanism-based feature extraction to solve the complexity.
- The existing models face various challenges, such as the network device's requirement to be placed inside the network traffic model to determine the entropy value effectively. To solve this issue, this work utilizes the adaptive deep learning model.
- The existing detection and mitigation model is not more efficient because of the limited resource usage in IoT devices. Hence, an efficient and effective protocol is required to detect the LR-DoS attack in the IoT environment. Therefore, the proposed model utilized a detection scheme based on the deep network with the added heuristic mechanism.
- Because of the inadequate training samples, the existing machine learning models are ineffective for attack detection. To address these challenges, this work implements feature extraction and the adaptive detection model to identify the LR-DoS attack with an effective solution.
- The traditional detection model for LR-DoS involves a rule-based system, statistical analysis, and anomaly detection. However, because of their prolonged nature, these mechanisms often face challenges in accurately detecting and mitigating minimum-rate attacks. This work implements the new adaptive deep learning framework to solve this issue.

3 Proposed Model and Description

LR-DoS attacks are typically subtle and persist over long durations, which are difficult to detect with conventional methods. Their low-rate nature enables them to produce legitimate traffic, complicating the identification process. Traditional detection systems, including statistical analysis and rule-based approaches, often struggle to accurately recognize LR-DoS attacks due to the attackers' ability to spread malicious traffic over time, thereby evading detection mechanisms. Additionally, the growing complexity of network environments, particularly in IoT contexts, introduces further challenges. IoT traffic's varied and dynamic characteristics obscure attack patterns, making detection even more challenging. LR-DoS attacks are designed to gradually deplete system resources, resulting in significant service degradation without

triggering immediate alerts, extending the attack's impact. To overcome such drawbacks, an effective LR-DoS attack detection model is implemented. The diagrammatic specification of the proposed LR-DoS attack detection scheme is provided in Fig. 1.

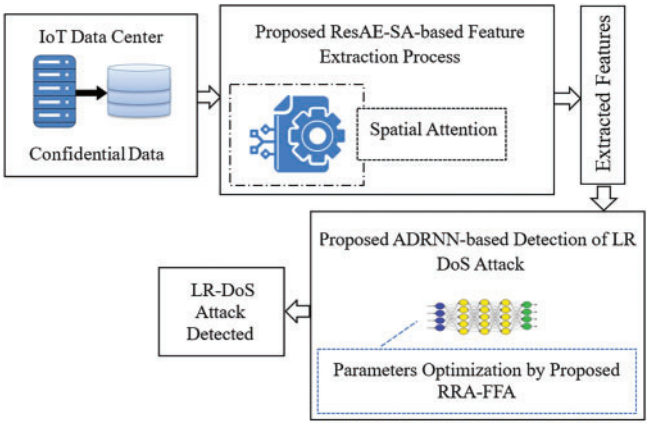

Figure 1: Diagrammatic representation of proposed LR-DoS attack detection model

The proposed LR-DoS attack detection model is developed to detect LR-DoS in an IoT environment, enhancing the reliability of data transmission through a detection scheme that focuses on fundamental statistical features, thereby reducing resource consumption for real-time applications. The data needed for this model is collected from the relevant source, which provides various attack scenarios and typical traffic patterns. Extracting the relevant features starts with the raw data, which is processed using a ResAE. This network architecture is enhanced with sparse attention mechanisms to confine the most pertinent features from the input data successfully. The residual Autoencoder allows efficient learning of the fundamental patterns in the data, while sparse attention focuses on the most considerable features, improving the quality of the extracted information. Once the features are extracted, they are fed into an ADRNN for classification. Sequential data are effectively handled using this method, making it suitable for detecting LR-DoS attacks. The efficiency of the proposed detection network is enhanced by optimizing the attributes from DRNN with the support of RRA-FFA. The objective function of the optimization process is to classify the input features using the ADRNN and evaluate the performance by comparing the target labels with the classified outcomes, ensuring that the ADRNN effectively identifies LR-DoS attacks. Furthermore, the integration of the RRA-FFA facilitates dynamic adjustments within the optimization process. This adaptability enables the model to respond more effectively to data variations, improving its accuracy in classifying LR-DoS attacks.

Dataset 1 (LR-DoS dataset): The LR-DoS dataset generally comprises network traffic data replicating different low-rate DoS attacks. It features benign and attack traffic, enabling researchers to train effectively and assess detection models. The relevant data are collected from the link https://data.mendeley.com/datasets/bzf9jcvhx4/1 (accessed on 12 September 2024). This dataset is usually gathered from controlled environments where network traffic is generated and observed, often utilized to simulate attack scenarios while capturing the resulting traffic patterns.

Dataset 2 (Cyber Security Dataset): This proposed model utilizes a cyber security dataset. This dataset provides data that helps identify common vulnerabilities in the system. The relevant data are collected from the link www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot (accessed on 12 September 2025). This dataset helps train the deep learning model effectively. The entire

collected data is represented as LI_X, where the term X is the count of total data collected from the dataset. Datasets 1 and 2 are well summarized on their size, Class distribution and Attack types in Table 2.

Table 2: Dataset description

Feature/Characteristic	Low rate DDoS (MQTT) dataset	Edge-IIoTset dataset
Size	Balanced classes of Train and Test comprising 160,000 on train and 39,994 on test	Unbalanced classes (later balanced on preprocessing) of Train and Test comprising 79,999 data on train and 20,000 data on test.
Class distribution	Normal, DDoS 99,995—Normal 99,999—DDoS	Normal, DDoS (HTTP, ICMP, TCP, UDP) Normal—24,302 DDoS—49,396
Attack types	Low-Rate DDoS attacks over MQTT protocol	14 attack types categorized into 5 threats: DoS/DDoS attacks, Information Gathering, Man-in-the-Middle attacks, Injection attacks, and Malware attacks but only 4 types of DDoS attacks were considered for study.

4 Attention-Aided Deep Learning-Based Feature Extraction for Improving Classification Processes

A key element of LR-DoS attack detection is efficient feature extraction, which guarantees that unprocessed network traffic data is converted into classification-relevant representations. High-dimensional data and noise are common problems for traditional deep learning techniques, which results in ineffective detection. To overcome these obstacles, this section presents an attention-aid deep Learning-Based Feature Extraction framework that combines sparse attention methods with residual autoencoders to improve classification accuracy. The Residual Autoencoder (ResAE) uses skip connections to enable deeper network training, improve learning efficiency, and decrease overfitting. Furthermore, the model may selectively concentrate on the most pertinent features while lessening the influence of redundant or noisy input thanks to the addition of sparse attention (SA). By integrating these methods, the suggested model successfully identifies key patterns in the data, leading to better feature extraction and more accurate LR-DoS attack categorization.

Furthermore, the model may selectively concentrate on the most pertinent features while lessening the influence of redundant or noisy input thanks to the addition of sparse attention (SA). By integrating these methods, the suggested model successfully identifies key patterns in the data, leading to better feature extraction and more accurate LR-DoS attack categorization. The following subsections detail the architecture, optimization techniques, and benefits of the ResAE-SA-based feature extraction module in enhancing classification performance.

4.1 Residual Autoencoder with Sparse Attention

Residual autoencoder [28] is a neural network architecture that integrates the function of autoencoders with residual learning. It comprises both an encoder and a decoder component. The encoder reduces the input data into a lower-dimensional representation, and the decoder reconstructs the original data. Residual learning incorporates skip connections, enabling the input to pass one or more layers within the network.

This approach reduces the overfitting problems and facilitates the training of deeper networks. In a residual autoencoder, the encoder's output is combined with the input to produce the network's final output.

Network Architecture: Residual autoencoder is composed of multiple layers, including convolutional and pooling layers. Its primary function is to lessen the dimensionality of raw data.

Convolutional layer: This layer is designed to extract features from the input data by performing convolution operations using learnable filters. It is mathematically expressed in Eq. (1).

$$R_{(p,l)} = \sum_{b=0}^{B-1} \sum_{h=0}^{H-1} S \cdot \gamma_b + J_p \tag{1}$$

here, the input term is represented as S, and the bias term is indicated as J_p. The convolution operation is specified as $S \cdot \gamma_b$. The output feature map is indicated $R_{(p,l)}$ at the position $R_{(p,l)}$. The dimensions of the filter are indicated as b and h, respectively.

Pooling layer: It reduces the facial feature dimension and performs a down-sampling process to decrease the computational complexity of data. The mathematical expression of the pooling layer is indicated in Eq. (2).

$$Ep\,(p,l) = \max\left(R_{(p,l)}\,(pD\,(p+1)\,D)\right) \tag{2}$$

here, the width of the pooling layer is indicated as D.

Residual block: It enables the input to bypass specific layers, which enhances gradient flow during training. The residual block Br is mathematically expressed in Eq. (3).

$$Br = G\,(D,p) + J_p \tag{3}$$

here, the term $G\,(D,p)$ is the layer's output within the residual block. The residual autoencoder is designed to reconstruct the input data. This capability enhances feature extraction and denoising, as the model concentrates on capturing the essential characteristics of the data by reducing noise.

Incorporating residual blocks makes the design and tuning process more difficult. Integrating residual blocks into the autoencoder framework requires careful attention to layer configurations and hyperparameters. As the model becomes more complex with adding layers and parameters, the risk of overfitting increases, mainly when the training dataset is small. To address this challenge, sparse attention is included in this proposed feature extraction model.

4.2 Feature Extraction Module

The collected input data LI_X is applied to the feature extraction phase. Feature extraction is vital in LR-DoS attack detection, where raw data is converted into representative features. The process starts with raw data. The raw data frequently includes noise and irrelevant information, making it essential to extract meaningful features that capture the underlying patterns. The main part of the feature extraction process is the residual Autoencoder, which consists of the encoder and the decoder, which ensure the accuracy of the features extracted during this module. This ensures that the features extracted by the encoder accurately represent the input data. To further improve the feature extraction process, sparse attention mechanisms are incorporated into the residual Autoencoder. It enables the model to focus on the valuable features of the input data by removing the less significant information. This is especially beneficial in high-dimensional datasets.

The attention mechanism assigns weights to different input data segments based on relevance. In the context of ResAE-SA, this allows the model to selectively highlight certain features while down-weighting others, resulting in a more efficient and effective representation. By enforcing sparsity in the attention weights, the ResAE-SA model is encouraged to focus on a limited number of key features, which helps to reduce noise and boost the interpretability of the extracted features. The output of the ResAE-SA network is a collection of features that capture the essential information from the raw data. These features are generally lower-dimensional and more informative than the original input, making them well-suited for classifying LR-DoS attacks. Finally, the extracted features are represented as D_f^{ResAE}. The diagrammatic demonstration of the ResAE-SA-based feature extraction model is given in Fig. 2.

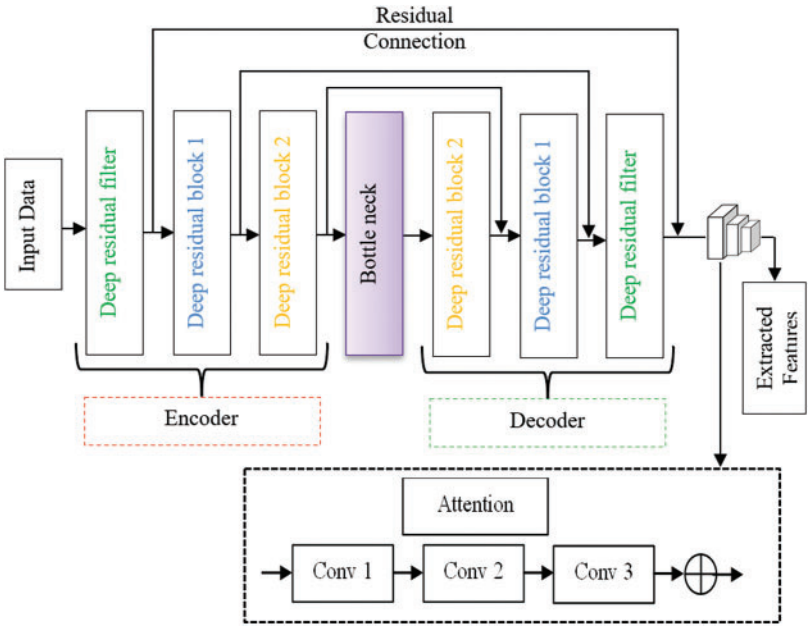

Figure 2: Diagrammatic representation of ResAE-SA-based feature extraction model

5 Adaptive Deep Learning Network for Classification of Attacks and Its Objective Function

To accurately classify Low-Rate Denial-of-Service (LR-DoS) attacks in Internet of Things (IoT) networks, sophisticated deep learning models that can effectively process sequential data are required. This section presents an Adaptive Deep Learning Network that is intended to improve the classification process by dynamically modifying its architecture and processing techniques. The Dense Recurrent Neural Network (DRNN) is the foundation of this model, which combines dense layers with recurrent connections to capture complex temporal dependencies in network traffic data. However, traditional DRNNs have issues with overfitting, high computational demands, and imbalanced datasets. To address these issues, this work suggests an Adaptive DRNN (ADRNN) optimized using the RRA-FFA (Resilient and Resource-Aware Fennec Fox Algorithm), which improves feature selection and dimensionality reduction. This section describes the DRNN framework, the adaptive learning mechanism, and the optimization strategy, emphasizing its efficacy in enhancing LR-DoS attack detection in resource-constrained IoT environments. The optimization process intends to minimize False Discovery Rate (FDR) and False Positive Rate (FPR) while maximizing the Matthews Correlation Coefficient (MCC), guaranteeing robust and reliable attack classification. Dense Recurrent Neural Network Description.

DRNN [29] is a specific type of architecture based on the function of RNN. It integrates the concepts of dense (fully connected) layers with a recurrent function, enabling the network to process sequential data effectively. DRNNs are particularly well-suited for tasks where the sequence of inputs is crucial, such as time series analysis. The recurrent connections in a DRNN enable preserving a hidden state that generates information from prior time steps, allowing the model to learn temporal dependencies effectively.

In a DRNN, every neuron in a layer is connected to all preceding layer neurons. This dense connectivity helps the network learn intricate input data representations. Dense layers capture complex patterns among the data, thereby improving the model's predictive accuracy. The input to a DRNN consists of sequences of feature vectors and is mathematically expressed in Eq. (4).

$$S = \{s_1, s_2, \ldots, s_V\} \tag{4}$$

here, the term V indicates the sequences' overall length. The network can collect earlier information in the sequence by updating the hidden state at each step, relying on the prior hidden state and the current input. A nonlinear activation function is typically used to determine the hidden state, giving the model non-linearity and permitting it to learn more intricate functions. It is statically expressed in Eq. (5).

$$Sh = \alpha\left(Ds_v + Fsh_{v-1} + J_p\right) \tag{5}$$

here, the hidden state is indicated as Sh, F is the weighted matrix of the hidden layer, α is the function of activation and D is the weight of the input layer. The output of the DRNN is derived from the hidden state at the current time step p. The output term is mathematically expressed in Eq. (6).

$$R(p) = \psi\left(YSh + J_p\right) \tag{6}$$

here, the softmax function is indicated as ψ, the weight matrix of the output layer is indicated as Y. The output layer typically uses an activation function like softmax for multi-class classification, converting the raw output scores into probabilities.

5.1 Low-Rate DoS Attack Detection in IoT

The process begins with a set of features D_f^{ResAE} extracted from the raw data. These features are necessary for identifying patterns indicative of LR-DoS attacks. Here, the classification is performed using ADRNN. This architecture is designed to handle sequential data effectively, making it appropriate for analyzing time-dependent network data.

In ADRNN, the extracted features are fed to recurrent connections that enable it to retain a hidden state across different time steps. This functionality lets the network capture past inputs needed to interpret intricate patterns. Furthermore, each neuron in one layer gets linked to every other neuron in the layer below because of the inclusion of thick layers. This dense connection increases the network's capability to collect complicated properties of the input sequences, making it easier for the network to discover detailed patterns and correlations within the data.

Yet, an imbalanced dataset delays the DRNN's learning process, as it may become biased towards the majority class and struggle to identify the minority class. Additionally, network traffic data is often high-dimensional, including numerous features representing different aspects of the traffic. This high dimensionality complicates the training process, making it difficult for the DRNN to recognize relevant patterns without risking overfitting. Furthermore, training DRNNs is resource-intensive, demanding considerable computational power and memory, which poses a challenge for deployment in resource-constrained

environments. To overcome these challenges, an adaptive nature of DRNN is designed with the aid of RRA-FFA in this proposed LR-DoS attack detection model.

The adaptive characteristics of the DRNN allow it to conduct feature selection and dimensionality reduction dynamically. By identifying and emphasizing the most pertinent features during training, RRA-FFA-ADRNN concentrates on the essential elements of the data that facilitate effective pattern recognition. This approach diminishes the risk of overfitting and improves the model's capacity. Additionally, the RRA-FFA-ADRNN adjusts its architecture and processing methods according to the available computational resources by optimizing the attributes of DRNN. This flexibility enables the RRA-FFA-ADRNN to function proficiently in resource-constrained settings. The objective functions of the proposed RRA-FFA-ADRNN-based LR-DoS attack detection are the minimization of the False Discovery Rate (FDR) and the false Positive Rate (FPR) along with the maximization of the Matthews Correlation Coefficient (MCC). The mathematical expression of the objective function is provided in Eq. (7).

$$F_{obj} = \underset{\left\{NH_n^{DRNN}, CE_v^{DRNt}, RL_h^{DRNN}\right\}}{\arg\min} \left(\left(\frac{1}{MCC}\right) + FDR + FPR\right) \tag{7}$$

here, the term NH_n^{DRNN} is the optimized hidden neuron count that varies in the range of $[5-225]$, CE_v^{DRNt} is the optimized neuron count that varies in the range of $[5-50]$, RL_h^{DRNN} is the optimized learning rate that varies in the range of $[0.01-0.99]$. The fitness function is a comprehensive measure that guides the training and evaluation of this proposed model, ensuring that it effectively identifies LR-DoS. The mathematical formulas used for calculating FDR, FPR, and MCC are provided in the equations below.

FDR: In this proposed model, minimum FDR reduces the number of instances where the model is incorrectly identified as usual or attacked. It is mathematically expressed in Eq. (8).

$$FDR = \frac{Rt}{Gt + Rt} \tag{8}$$

FPR: A lower FPR in this proposed model indicates that the model reduces misclassification. It is mathematically expressed in Eq. (9).

$$FPR = \frac{Rt}{Rt + Gj} \tag{9}$$

The MCC of the proposed model is calculated using Eq. (10).

$$MCC = \frac{(Gt * Gj) - (Rt * Rj)}{\sqrt{(Gt * Rt) * (Gt + Rj) * (Gj + Rt) * (Gj + Rj)}} \tag{10}$$

here, the valid positive and negative values are indicated as Gt and Gj, respectively. False positive and negative values are indicated as Rt and Rj, respectively. Fig. 3 shows the diagrammatic depiction of the suggested LR-DoS attack detection in the IoT system.

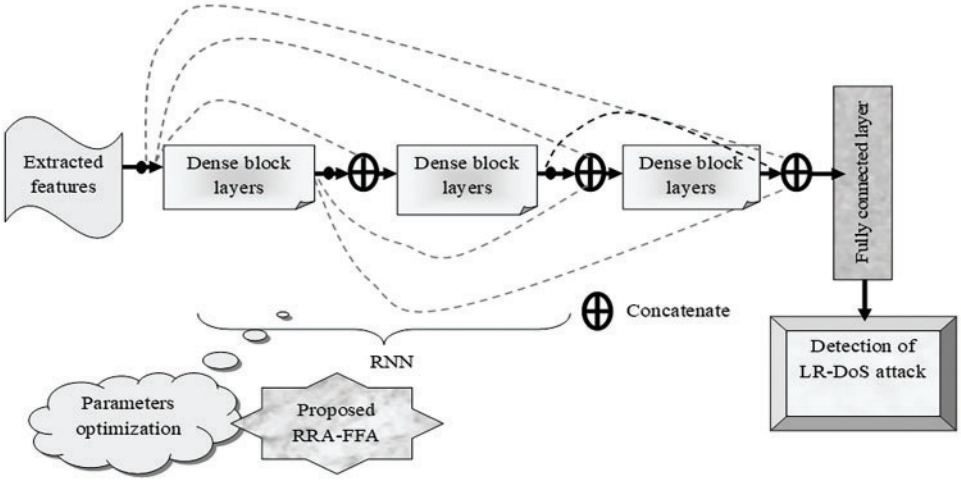

Figure 3: Diagrammatic representation of proposed LR-DoS attacks detection in IoT model

5.2 Presented RRA-FFA

FFA [30] is enthused by the behaviors and survival strategies of the fennec fox, especially its adaptability to harsh environments and its effective hunting methods. This algorithm's adaptive mechanisms allow it to adjust its search efforts based on the unique features of the optimization issue, enhancing its performance in various situations. Furthermore, its performance on benchmark functions shows that FFA has proven resilient in handling a variety of optimization issues. Based on these conditions, the position is updated using Eq. (11).

$$R_{k,l} = \left\{ \begin{array}{ll} r_{k,l} + U_r \cdot (r_{U,K,l} - H \cdot r_{k,l}) & B_{k,l} \triangleleft B_k \\ r_{k,l} + U_r \cdot (r_{U,K,l} - r_{k,l}) & else \end{array} \right\} \tag{11}$$

here, the term $r_{U,K,l}$ is the initial position, $R_{k,l}$ is the updated position, U_r is the random variable, H is the influence factor, and B is the fitness value. Yet, FFA converges more slowly than expected, especially in complex settings. This slower convergence impacts the efficiency of identifying optimal solutions. Additionally, preserving diversity within the population of solutions is essential for successful exploration. Therefore, FFA needs to incorporate mechanisms that prevent premature convergence by updating its random variable U_r.

N represents the population size (number of fennec foxes or candidate solutions in the optimization process). T represents the number of iterations (how often the algorithm runs to refine the solutions). Hence, the overall time complexity is O(N·T), which makes it scalable for large datasets. Since each iteration performs operations on all population members, the total number of operations grows proportionally to N × T, leading to an O(N·T) time complexity. As discussed in step 2, the decision logic includes Exploration and Exploitation, wherein the objective of the Exploration phase is to discover diverse and promising regions in the solution space.

The strategy includes: N represents the population size (number of fennec foxes or candidate solutions in the optimization process). T represents the number of iterations (how often the algorithm runs to refine the solutions). Hence, the overall time complexity is O(N·T), which makes it scalable for large datasets. Since each iteration performs operations on all population members, the total number of operations grows proportionally to N × T, leading to an O(N·T) time complexity. As discussed in step 2, the decision logic

includes Exploration and Exploitation, wherein the objective of the Exploration phase is to discover diverse and promising regions in the solution space.

- The random variable r in Eq. (12) is large, resulting in more significant jumps in the search space.
- This high randomness helps the Fennec fox solutions explore new potential optimal solutions instead of getting stuck in local optima.
- Encourages diversity in the population to avoid premature convergence.

Exploitation phase aims to fine-tune solutions and converge to the best possible value. The strategy includes:

- The random variable r is reduced in Eq. (12), making movements smaller and more refined.
- The focus shifts towards intensification, improving promising solutions rather than exploring new ones.
- This phase refines the best candidate solutions identified earlier, ensuring precise convergence.

The improved algorithm RRA-FFA focuses on the iterative process of updating the positions of the solution population. The mathematical form of the newly developed concept for updating the random variable is provided in Eq. (12).

$$U_r = \frac{Fct}{(Fwt + Fmt - Fbt + Fct)} \tag{12}$$

here, the best, worst, mean, and current fitness values are indicated as Fbt, Fwt, Fmt and Fct, respectively. The computation of U_r. Considering the current, worst, mean, and best fitness values enables a more dynamic evaluation of each solution's performance. By including the worst and mean fitness values in the calculation of U_r, the RRA-FFA accurately identifies underperforming solutions, promoting exploration of new regions within the solution space and potentially enhancing the performance in the LR-DoS attack detection phase. The pseudocode of the proposed RRA-FFA model is provided in Algorithm 1.

Algorithm 1: RRA-FFA algorithm

Input: *Attribute set for optimization*
Output: *Optimized attribute set*
Step 1: Initialization
1. Define Parameters:
Set population size N, maximum iterations T, and control parameters for optimization.
2. Initialize Population:
Randomly initialize the positions of fennec foxes (solutions).
Evaluate the fitness function for each fox to determine the initial quality of solutions.
Step 2: Iterative Optimization Process
3. For *t = 1 to T* ***do***
For *each fox i in N* ***do***
1. Modify Random Variable (Exploration-Exploitation Balance):
Adjust the ***random coefficient*** *using* **Eq. (11)** *to control search intensity.*
Early Phase iteration → Prioritize exploration (high randomness).
Later Phase iteration → Prioritize exploitation (low randomness).
2. Digging Phase (Exploration Mechanism):
Search for prey (better solutions) by modifying the position vector.

(Continued)

Algorithm 1 (continued)

If fitness ***improves*** *after digging → update the position.*
Else → retain the previous best position.

3. Status Update (Position Refinement):

Update the position using **Eq. (10)** *based on the best-found solution.*
If the new position provides a lower error value → accept the change.
Else → retain the prior position to prevent unnecessary perturbations.

4. Escaping Strategy (Avoiding Local Optima—Exploitation Phase):

Adjust the movement to escape predators and prevent premature convergence.
If fox ∈ ***suboptimal region*** *→ introduce a* ***large perturbation*** *for exploration.*
If fox is close to the ***global best*** *→ apply* ***small adjustments*** *to refine accuracy.*

5. Position Update:

Store the newly identified best solution for the fox.

End For

Global Best Selection:

Identify the ***best-performing fox*** *(solution) in the current iteration.*
If it is better than the previously stored ***global best*** *→ update it.*

4. End For

Step 3: Output Optimized Solution

5. *Return the final optimized attributes after convergence.*

6 Results and Discussion

This section thoroughly analyzes the suggested RRA-FFA-ADRNN model for LR-DoS attack detection, demonstrating how variations affect accuracy and computational efficiency. The experimental setup describes the model's implementation and training procedure, and then it is evaluated using standard performance metrics like accuracy, Critical Success Index (CSI), and False Omission Rate (FOR). A convergence analysis is carried out to validate the model's effectiveness further, looking at the model's stability and optimization during training. In conclusion, the accuracy analysis highlights the superior detection capability of the RRA-FFA-ADRNN model by comparing it to existing deep learning and optimization. The results confirm that the suggested model performs better than traditional methods, achieving higher accuracy and robustness in identifying LR-DoS attacks within IoT environments.

6.1 Experimental Setup

A practical model for LR-DoS attack detection was designed and implemented with Python language support. With a maximum iteration of 50 and a population size of 10, the obtained data is split into 75% for training and 25% for testing. The chromosomal length is set at 3. By comparing the suggested model to the current detection models, the success rate of the former was evaluated. The existing algorithms are Gorilla Troops Optimizer (GTO) [31], Emperor Penguins Colony (EPC) [32], Dolphin Pod Optimization (DPO) [33], and FFA [30]. Conventional techniques like FFCNN [19], SVM [20], RNN [24], and DRNN were also utilized to assess the validation of the proposed model.

6.2 Performance Metrics

The mathematical formulas of performance metrics utilized for the validation process are specified below.

Accuracy (Cy): The accuracy of the proposed model is calculated using the formula in Eq. (13).

$$Cy = \frac{Gt + Gj}{(Gt + Gj + Rt + Rj)} \tag{13}$$

Critical Success Index (CSI): This metric assesses correctly detected attacks' proportions. It is considered using the formula in Eq. (14).

$$CSI = \frac{Gt}{(Gt + Rt + Rj)} \tag{14}$$

The proposed model's false Omission Rate (FOR) is assessed using Eq. (15).

$$FOR = \frac{Rj}{Rj + Gt} \tag{15}$$

The proposed scheme's Bookmaker Informedness (BM) is dignified using the formula in Eq. (16).

$$BM = Specificity + Sensitiivity \tag{16}$$

The Markedness (MK) of the considered model is dignified using the formula in Eq. (17).

$$MK = \Pr + NPV \tag{17}$$

here, the term Pr is precision and it is dignified using the formula in Eq. (18).

$$\Pr = \frac{Gt}{Gt + Rj} \tag{18}$$

6.3 Convergence Analysis of the Proposed Model

The convergence analysis of the proposed model performance across different detection models is given in Fig. 4a. The optimal classification threshold value for the task is identified by examining the convergence curve. The proposed outcome is compared against various models. Convergences analysis assesses how quickly and reliably the proposed RRA-FFA-ADRNN model achieves a stable solution during training, which is vital for ensuring effective deployment in real-time environments. The given plots in Fig. 4a,b represent different algorithms' convergence behavior in minimizing the cost function over iterations on Datasets 1 and 2, respectively. The x-axis represents the number of iterations, while the y-axis represents the cost function value. Convergence curves on Dataset 1 include an interpretation wherein the initial phase of 0–10 iterations shows that the cost function starts at a relatively high value (~1.7–1.8) for almost all the comparative algorithms. A sharp decline is observed within the first few iterations, indicating rapid convergence. RRA-FFA-ADRNN and FFA-ADRNN shows the fastest reduction in cost, reaching around 1.2–1.3. The middle phase including 10–30 iterations shows that the curves begin to stabilize, with small fluctuations in cost values. GTO-ADRNN exhibits a sudden drop around iteration 40, suggesting delayed convergence compared to other algorithms. Most algorithms reach a plateau, meaning minimal further improvements by the end of 50 iterations. RRA-FFA-ADRNN achieves the lowest cost function value (~1.2), indicating superior performance. Similarly, convergence curves on the Dataset 2 interprets that in the initial stages of iterations the cost function starts at a higher value (~2.0 for some algorithms). The EPC-ADRNN has a drastic drop initially, but it fluctuates heavily before stabilizing. Followed by some fluctuations, but most algorithms start settling around 1.3–1.4. FFA-ADRNN and RRA-FFA-ADRNN consistently maintain

lower cost values, suggesting better stability. Finally, by the end of iterative performance RRA-FFA-ADRNN remains the most stable and achieves the lowest cost function, similar to the first dataset. The GTO-ADRNN and DPO-ADRNN models show minor improvements but do not reach the optimal convergence levels of RRA-FFA-ADRNN. Consolidating the interpretation:

- RRA-FFA-ADRNN consistently outperforms other methods, achieving the lowest final cost values across both datasets.
- FFA-ADRNN also shows strong convergence but is slightly less stable than RRA-FFA-ADRNN.
- GTO-ADRNN and EPC-ADRNN experience delays in convergence, with sudden improvements in later iterations.
- DPO-ADRNN performs reasonably well but does not reach the lowest cost values.
- Overall, RRA-FFA-ADRNN is the most effective approach, demonstrating faster convergence, lower cost function values, and improved stability across both datasets.

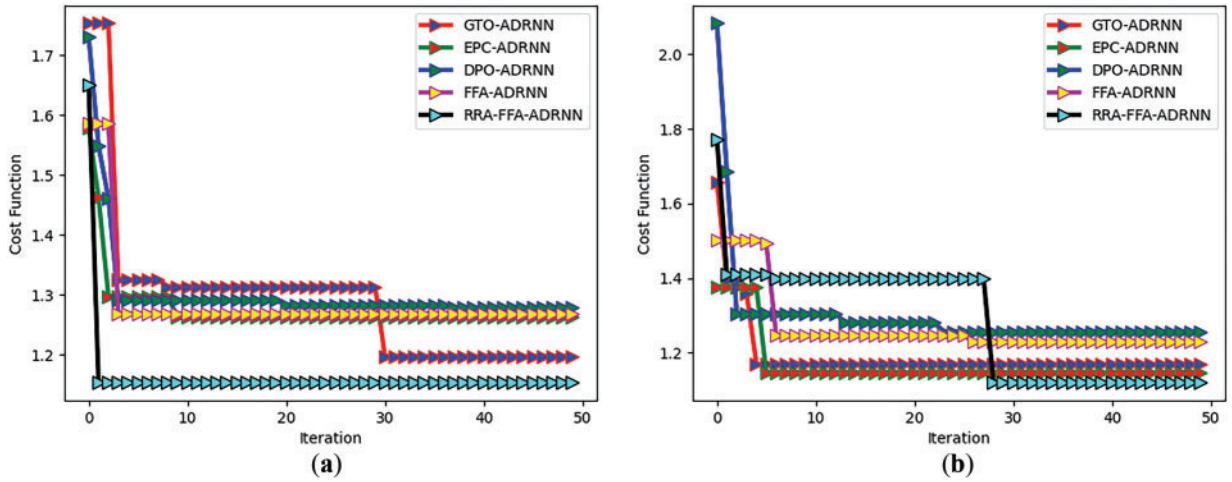

Figure 4: Convergence analysis of proposed LR-DoS attack detection model regarding (**a**) Dataset 1 and (**b**) Dataset 2

6.4 Performance Analysis Based on Batch Size

Fig. 5 specifies the graphical view of performance comparison of various optimization algorithms on Dataset 1 and Dataset 2, and Fig. 6 provides the graphical view of performance metrics on various deep learning methods using Dataset 1 and Dataset 2. Different batch sizes impact the speed of training. Analyzing accuracy helps to identify the detection efficiency that provides detection results based on training data. At batch size 4, the accuracy of the designed RRA-FFA-ADRNN model progressed by 9.2% compared to GTO-ADRNN, 11.6% compared to EFC-ADRNN, 10.9% compared to DPO-ADRNN, and 4% compared to FFA-ADRNN. The choice of batch size influences the model's ability. Higher batches with less noise in the gradient estimation help boost the proposed model's flexibility. Analyzing accuracy helps to assess how the RRA-FFA-ADRNN model performs on validation data with different batch sizes. Based on Fig. 6a, the accuracy of the proposed RRA-FFA-ADRNN is boosted by 12.9%, 9.09%, 11.6%, and 10.9% over FFCNN, SVM, RNN, and DRNN, using Dataset 2 at the batch size of 4. Analyzing accuracy with varying batch sizes, the proposed RRA-FFA-ADRNN detection performance is compared against other techniques, highlighting its strengths and weaknesses in different training scenarios. Based on these analyses, the proposed RRA-FFA-ADRNN model is more effective in LR-DoS attack detection.

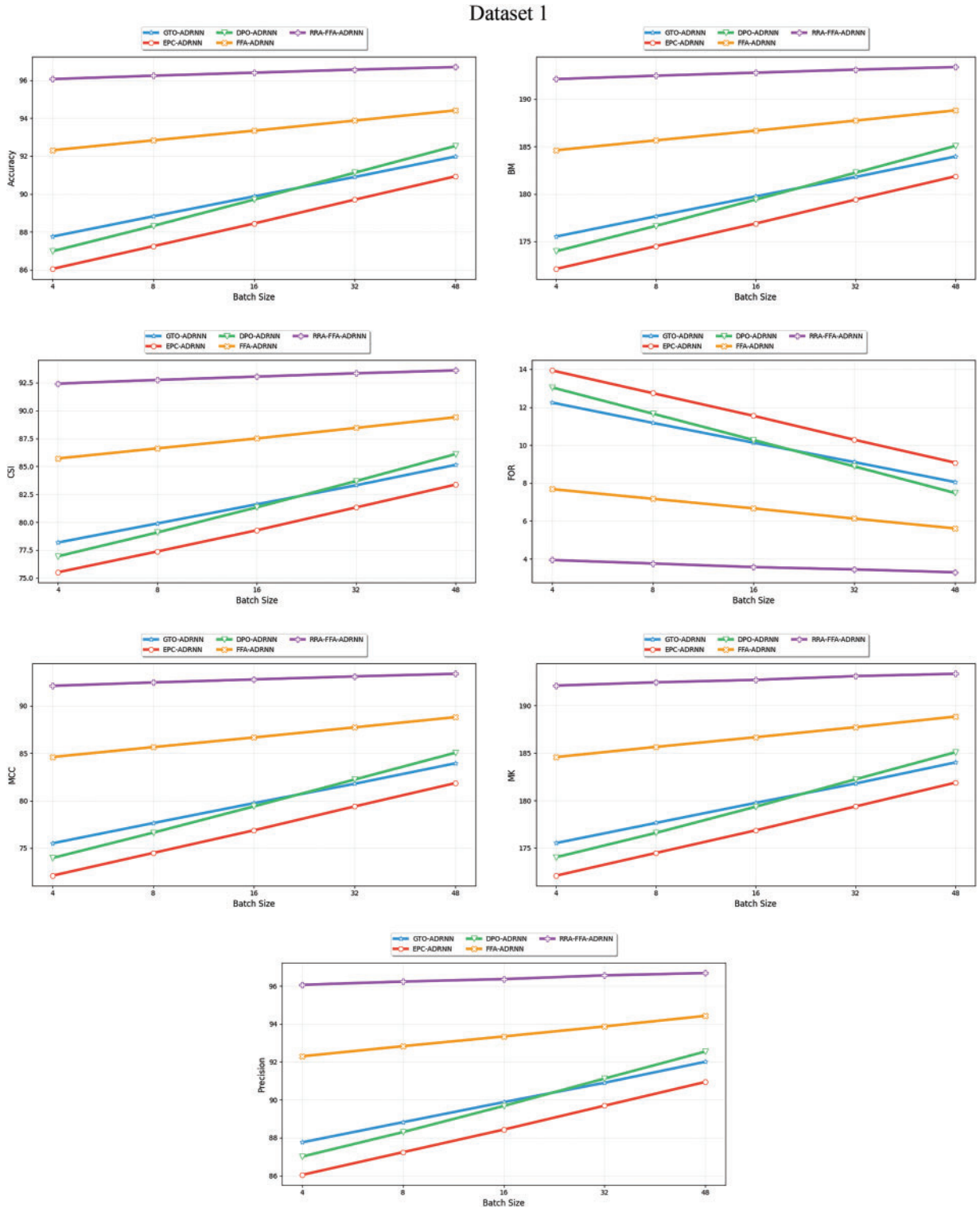

Figure 5: (Continued)

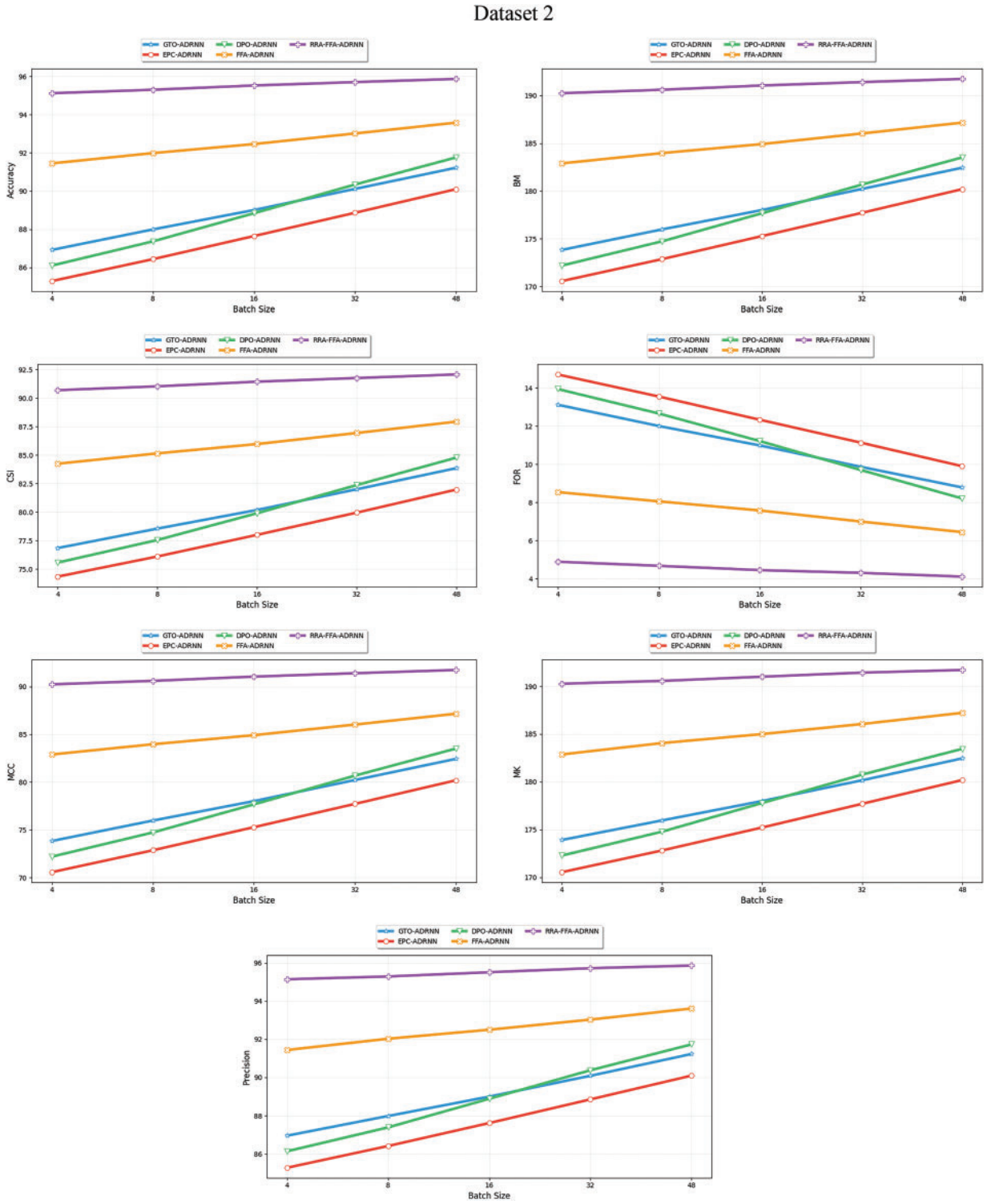

Figure 5: Performance metrics analysis of the proposed optimized LR-DoS attack detection model using Datasets 1 and 2

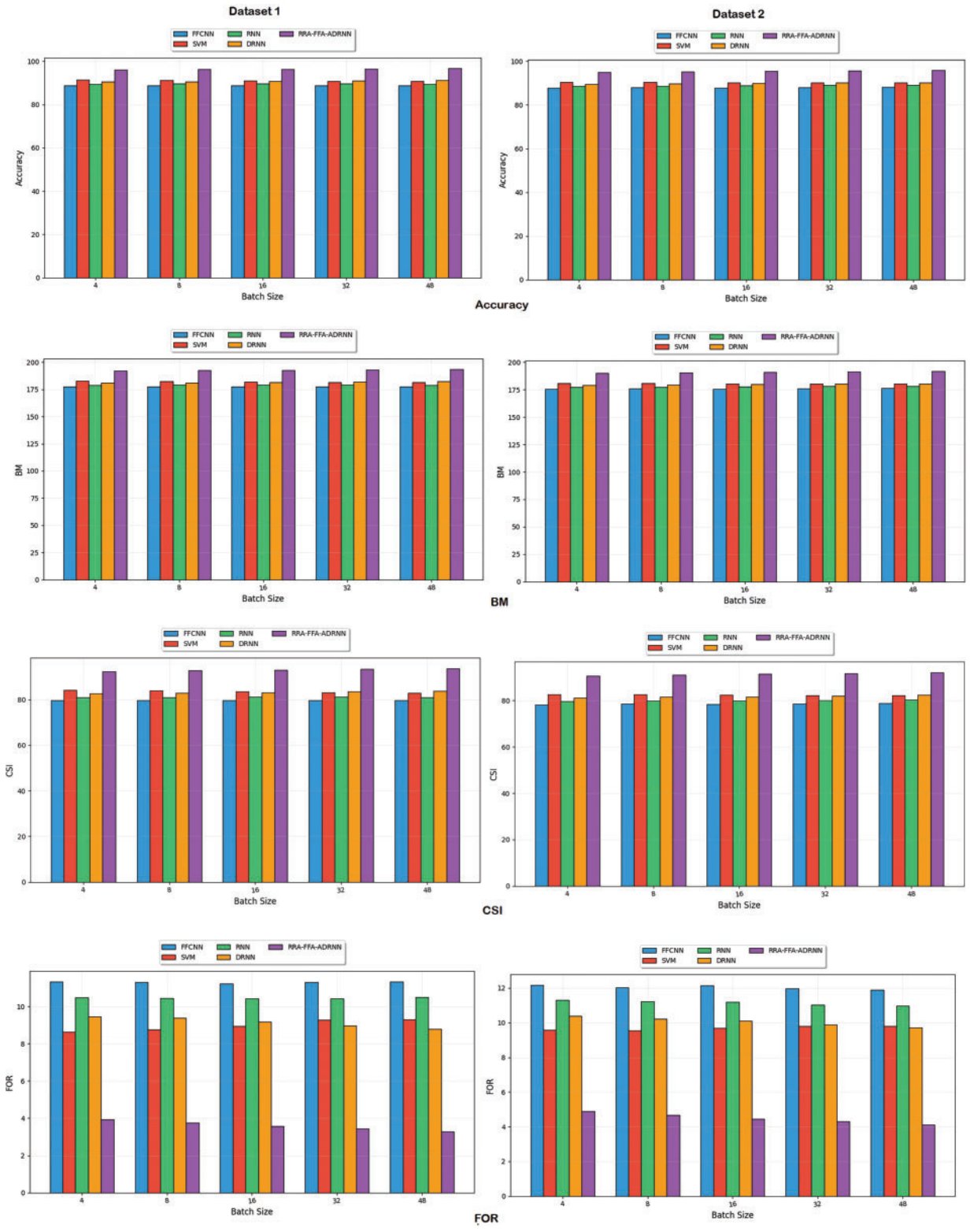

Figure 6: (Continued)

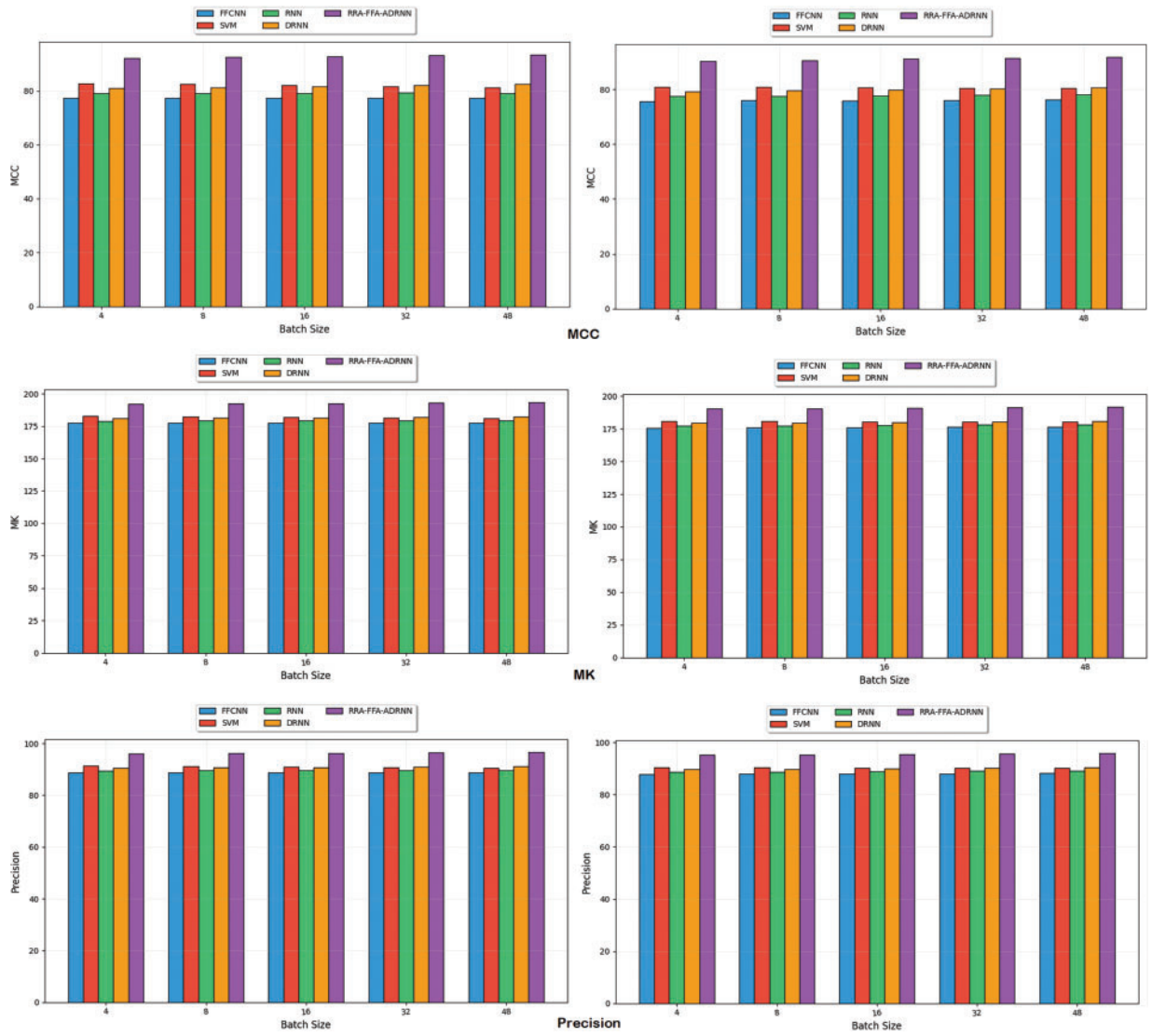

Figure 6: Performance metrics analysis of the deep learning models of attack detection using Datasets 1 and 2

Fig. 5 shows that accuracy increases with batch size for all models. RFA-FHA-ADRNN consistently achieves the highest accuracy, nearing 96% in both datasets. FHA-ADRNN follows closely, maintaining a significant lead over the other models. EPC-ADRNN has the lowest accuracy performance. BM values also improve as batch size increases. Similar to accuracy, RFA-FHA-ADRNN remains the best performer, followed by FHA-ADRNN.GTO-ADRNN and DPO-ADRNN (Green) show steady improvements but remain below FHA-ADRNN.EPC-ADRNN consistently performs the worst.

The CSI metric follows the same pattern of increasing with batch size. RFA-FHA-ADRNN consistently outperforms all others. FHA-ADRNN is the second-best performer. EPC-ADRNN has the lowest CSI scores. False Omission Rate (FOR) decreases as batch size increases. RFA-FHA-ADRNN maintains the lowest FOR at all batch sizes. EPC-ADRNN has the highest FOR, indicating weaker performance in avoiding false omissions. FHA-ADRNN consistently outperforms other models except RFA-FHA-ADRNN. Matthew's Correlation Coefficient (MCC) increases with batch size. RFA-FHA-ADRNN has the highest MCC, reaching

values around 96% at large batch sizes. FHA-ADRNN follows, with MCC above 90%. GTO-ADRNN and DPO-ADRNN show similar trends but at lower values. EPC-ADRNN performs the worst. MK Metric shows a positive correlation with batch size. RFA-FHA-ADRNN has the best results. FHA-ADRNN follows closely. Other models lag, with EPC-ADRNN being the weakest. Precision increases with batch size. RFA-FHA-ADRNN maintains the highest precision across all batch sizes (near 100%). FHA-ADRNN is the second-best performer. EPC-ADRNN has the lowest precision. Consolidating the performance metrics, RFA-FHA-ADRNN is the best-performing model across all metrics and datasets. FHA-ADRNN is the second-best in all metrics. EPC-ADRNN consistently underperforms compared to other models. Larger batch sizes improve all performance metrics, particularly reducing FOR and increasing MCC, MK, and Precision.

Accuracy increases with batch size for all models. RFA-FFA-ADRNN consistently has the highest accuracy at larger batch sizes, close to 98–100%. FF-CNN, SVM, PNN, and DNN perform similarly, but FF-CNN and DNN appear slightly better than SVM and PNN. Smaller batch sizes (4, 8) show lower accuracy across all models. BM values improve as batch size increases. RFA-FFA-ADRNN has the best BM score across all batch sizes. Other models show close competition, with FF-CNN and DNN performing slightly better than SVM and PNN. BM values range between 150 and 200, with RFA-FFA-ADRNN reaching the upper limit. CSI follows the same trend as accuracy and BM, increasing with batch size. RFA-FFA-ADRNN has the highest CSI values, close to 90+%. FF-CNN, SVM, and DNN are closely matched, but SVM lags slightly.

PNN consistently has the lowest CSI values, indicating weaker performance. RWA-FFA-ADNN consistently achieves lower rates compared to other models, indicating better performance in terms of minimizing false negatives and false discoveries. SVM and FICNN tend to have higher FOR and FDR values, indicating they are less reliable in this context. RWA-FFA-ADNN generally shows higher values of MCC and MK, suggesting strong positive predictive power and balanced accuracy. FICNN and SVM often have slightly lower scores, while RNN and DNN are competitive but not as consistently high as RWA-FFA-ADNN.RWA-FFA-ADNN has consistently high precision across all batch sizes, followed closely by DNN. The difference in precision between models is less pronounced here than other metrics. Finally, as the batch size increases, performance for most metrics generally improves or remains stable, particularly for RWA-FFA-ADNN and DNN. Smaller batch sizes exhibit more variation and lower performance, especially in the FOR and FDR metrics. RFA-FFA-ADRNN outperforms all other models in Accuracy, BM, and CSI. DNN and FF-CNN are better than SVM and PNN but are still significantly weaker than RFA-FFA-ADRNN. SVM and DNN show the weakest results across all metrics.

6.5 Accuracy Analysis of Proposed Model

Table 3 presents the accuracy analysis of numerical outcomes Comparison analysis of optimization algorithms and approaches using Dataset 1 and the analysis based on Dataset 2.

The best design with the highest accuracy for LR-DoS attack detection is found by adjusting the number of hidden neurons. This aids in adjusting the model to detect LR-DoS attacks as effectively as feasible. As a result, accuracy analysis is crucial for confirming the suggested LR-DoS attack detection model's success rate. It should be considered with other metrics to assess its capacity to accurately identify LR-DoS assaults in an IoT context.

Table 3: Comparison analysis of optimization algorithms and techniques using datasets 1 and 2—accuracy analysis

Dataset 1										
	Comparison analysis of optimization algorithms					Comparison analysis of deep learning methods				
Hidden neuron counts	GTO-FFA-ADRNN [31]	EPC-FFA-ADRNN [32]	DPO-FFA-ADRNN [33]	FFA-FFA-ADRNN [30]	RRA-FFA-ADRNN	FFCNN [19]	SVM [20]	RNN [24]	DRNN	RRA-FFA-ADRNN
100	87.794	86.118	86.942	92.28	95.144	89.132	91.87	89.518	90.034	95.144
200	88.684	87.154	88.098	92.666	95.266	88.856	91.4	89.306	90.152	95.266
300	89.492	88.066	89.292	92.942	95.412	88.596	90.944	89.204	90.204	95.412
400	90.35	89.248	90.582	93.324	95.604	88.348	90.47	89	90.246	95.604
500	91.234	90.358	91.756	93.658	95.716	88.04	90.012	88.82	90.356	95.716
Dataset 2										
100	86.93	85.23	86.002	91.38	94.17	88.692	91.43	88.76	89.584	94.17
200	87.856	86.384	87.3	91.712	94.378	88.294	90.872	88.46	89.416	94.378
300	88.622	87.482	88.5	92.1	94.536	87.992	90.23	88.374	89.186	94.536
400	89.568	88.584	89.764	92.486	94.724	87.658	89.708	88.102	89.102	94.724
500	90.494	89.736	91.034	92.842	94.916	87.26	89.212	88.008	88.864	94.916

The accuracy of the RRA-FFA-ADRNN model varies based on the quality of the training and testing datasets. A well-balanced dataset with diverse attack types and benign samples will likely produce a more accurate model. At the hidden neuron count 500, the accuracy of the proposed RRA-FFA-ADRNN model is 95.7% using Dataset 1 and 94.1% using Dataset 2, which is better than the existing models. Thus, our approach's accuracy is more extensive compared to baseline techniques.

7 CSI Analysis of Proposed Model

The CSI analysis of the designed LR-DoS attack detection model using Dataset 1 is provided in Table 4, and the analysis using Dataset 2 is provided in Table 5. A high detection rate is essential for the effectiveness of attack detection as it demonstrates the model's capability to recognize malicious traffic. At the hidden neuron count 500, the CSI of the proposed model is progressed with 16.7% than FFCNN, 12.1% than SVM, 14.8% than RNN, and 11.3% than DRNN using Dataset 1. CSI analysis helps in achieving a balance between bias and variance. Based on Dataset 2, the CSI of the proposed model is 90.3% using Dataset 2, which is better than traditional algorithms. A model with few hidden neurons exhibits high bias, while a higher count generates high variance. Hidden neuron count is a critical hyperparameter for selecting the most effective configuration for the LR-DoS detection system. Thus, CSI analysis proved that the evaluation and improvement of the detection system with the support of the RRA-FFA-ADRNN model is better than existing models. As new types of LR-DoS attacks emerge, continuous monitoring of CSI helps the RRA-FFA-ADRNN model to maintain high detection rates.

Table 4: Comparison analysis of optimization algorithms and techniques using datasets 1 and 2—CSI analysis

Dataset 1										
	Comparison analysis of optimization algorithms					Comparison analysis of deep learning methods				
Hidden Neuron Counts	GTO-FFA-ADRNN [31]	EPC-FFA-ADRNN [32]	DPO-FFA-ADRNN [33]	FFA-FFA-ADRNN [30]	RRA-FFA-ADRNN	FFCNN [19]	SVM [20]	RNN [24]	DRNN	RRA-FFA-ADRNN
100	78.2377	75.6096	76.8974	85.6633	90.7345	80.3989	84.9656	81.0328	81.8734	90.73459
200	79.6738	77.2234	78.7289	86.3339	90.959	79.9438	84.1626	80.6765	82.0776	90.95909
300	80.983	78.6771	80.6547	86.8143	91.2245	79.5253	83.395	80.5238	82.1488	91.22451
400	82.3995	80.5906	82.7906	87.4854	91.5785	79.1295	82.598	80.1866	82.2176	91.57854
500	83.8854	82.4192	84.7694	88.0699	91.7817	78.6344	81.8367	79.887	82.4072	91.78177
Dataset 2										
100	76.8909	74.2853	75.4498	84.1299	88.9842	79.6786	84.2144	79.7819	81.1345	88.9842
200	78.3514	76.0315	77.4702	84.6904	89.3546	79.0432	83.2716	79.3071	80.8552	89.35469
300	79.5756	77.7481	79.3707	85.3589	89.6389	78.5632	82.1988	79.182	80.4815	89.63896
400	81.1185	79.4846	81.4208	86.0183	89.9772	78.0321	81.3449	78.7231	80.3476	89.9772
500	82.6368	81.3869	83.5456	86.6415	90.3246	77.4017	80.5256	78.5803	79.9705	90.32467

Table 5: Statistical analysis of the proposed LR-DoS attack detection model

	Dataset 1					Dataset 2				
	GTO-FFA-ADRNN [31]	EPC-FFA-ADRNN [32]	DPO-FFA-ADRNN [33]	FFA-FFA-ADRNN [30]	RRA-FFA-ADRNN	GTO-FFA-ADRNN [31]	EPC-FFA-ADRNN [32]	DPO-FFA-ADRNN [33]	FFA-FFA-ADRNN [30]	RRA-FFA-ADRNN
Best	1.1972	1.2629	1.2788	1.26854	1.1532	1.16806	1.1454	1.2562	1.2280	1.1190
Worst	1.7532	1.5789	1.73081	1.58517	1.6501	1.65642	1.3753	2.0832	1.5008	1.7728
Mean	1.2946	1.2780	1.30259	1.28754	1.1632	1.19062	1.1684	1.2967	1.2677	1.2838
Median	1.3136	1.2629	1.28302	1.26854	1.1532	1.16806	1.1454	1.2562	1.2458	1.3976
std.dev	0.1290	0.0522	0.07549	0.07519	0.0695	0.08368	0.0689	0.1279	0.0859	0.1549

Statistical Analysis of the Proposed LR-DoS Attack Detection Model

The proposed statistical analysis of the LR-DoS attack detection model, which involves varying the hidden neuron count, focuses on systematically assessing modifications in the number of hidden neurons among different conventional methods. For each configuration of hidden neurons, it is essential to compute the performance metrics' mean, median, variance, and standard deviation. The best value of the proposed RRA-FFA-ADRNN model is 4.19% better than GTO-FFA-ADRNN, 2.29% than EPC-FFA-ADRNN, 10.9 than DPO-FFA-ADRNN, and 8.8% than FFA-FFA-ADRNN using Dataset 2. This comprehensive statistical analysis identifies the best method for detecting LR-DoS attacks, ultimately informing the development of more effective IoT security systems. Table 5 presents numerical values of the statistical analysis among both Dataset 1 and Dataset 2.

8 Conclusion

An innovative deep-learning framework was created to recognize LR-DoS attacks in IoT settings. The framework employed an ADRNN model with a residual autoencoder and sparse attention for efficient feature extraction with classification. The RRA-FFA algorithm was utilized to fine-tune critical parameters of the ADRNN, thereby improving the classification accuracy. The findings indicate that this framework effectively reduces FDR and FPR while maximizing the MCC, highlighting its strong ability to identify LR-DoS attacks accurately. At the hidden neuron count 100, the designed model accuracy progressed with 6.1% than FFCNN, 2.9% than SVM, 6.09% than RNN, and 5.1% than DRNN using Dataset 2. Thus, the RRA-FFA-ADRNN model was designed to achieve high detection accuracy, which was crucial for effectively identifying LR-DoS attacks.

Research Limitations and Future Scopes: Despite the promising results demonstrated by the proposed RRA-FFA-ADRNN model, several areas warrant further investigation to enhance its applicability and robustness in real-world scenarios. The study's contribution would be strengthened, and future developments in LR-DoS detection would be guided by addressing these limitations and investigating potential research avenues. One of the main drawbacks of the proposed framework is its potential scalability issues when applied to large, dynamic datasets. It is unclear whether the model can maintain high detection accuracy and low computing costs on massive data streams as IoT networks continue to expand in size and complexity. To ensure scalability without compromising performance, future studies could focus on refining the model design and leveraging distributed computing strategies. Furthermore, the proposed architecture primarily relies on standard datasets, which may not accurately represent the varied and dynamic nature of actual LR-DoS attacks. Incorporating real-world traffic patterns and regularly updating the model with the latest attack signatures can increase robustness and adaptability to new threats. Although the RRA-FFA-ADRNN model outperforms current methods in terms of accuracy and error rates, the computational requirements of deep learning models provide a significant obstacle to their implementation on IoT devices with limited resources. This difficulty may be mitigated by exploring lightweight model architectures or incorporating model compression strategies, such as pruning or quantization. Additionally, investigating different optimization algorithms that reduce computational overhead without compromising accuracy might enhance the model's suitability for real-time situations. Integrating the suggested framework with edge computing platforms will enable quicker reaction times and lower latency, while real-time detection remains essential for preventing major damage from LR-DoS attacks. Furthermore, performance may be improved and the computational load on centralized cloud servers reduced by modifying the model to operate in a distributed manner across edge nodes. The RRA-FFA-ADRNN model can be integrated with other security strategies, such as anomaly detection systems and signature-based approaches, to enhance the system's overall defense. Meanwhile, hybrid models that combine deep learning and conventional techniques can address the limitations of false positives and ensure robust detection in various environments. Because cyberattacks are constantly evolving, its performance may deteriorate over time if the suggested model is not updated regularly. Adaptive learning methods that dynamically retrain the model as new attack patterns emerge should be implemented to preserve the model's effectiveness in identifying new threats. Rapid adaptability to different datasets and contexts may also be achieved by incorporating transfer learning approaches without the need for comprehensive model retraining.

Future research could focus on implementing pipeline parallelism to accelerate data processing, designing efficient data buffering mechanisms to handle continuous inputs without compromising accuracy, and optimizing a model for real-time deployment by reducing inference latency through model simplification or lightweight architectures. This approach aims to expand the current work and enhance its practical applicability. By adjusting the model to operate effectively on edge devices with constrained computational power, utilizing distributed model training and inference across multiple edge nodes, and distributing the

computational load between edge and centralized servers, integrating the model with edge computing can significantly reduce latency and increase responsiveness. Additionally, the model could update dynamically and adjust to new attack patterns without requiring total retraining by integrating adaptive and self-learning capabilities, such as online and federated learning, ensuring robustness even in various dynamic situations. Accuracy can be increased while reducing the chance of overfitting by utilizing a variety of datasets from different IoT scenarios and investigating multimodal learning with contextual data to enhance the model's generalizability. Comprehensive multi-layered protection against LR-DoS assaults could be achieved by combining the model with additional security strategies, such as anomaly detection and signature-based systems.

Furthermore, extending the application of the proposed model to blockchain ecosystem security for malicious attacks detection could offer decentralized and tamper-resistant protection, especially in smart contract environments and distributed ledgers. Leveraging blockchain's inherent transparency and immutability and real-time attack detection mechanisms could enhance trust and integrity within decentralized applications. Additionally, if the model's performance for devices with limited resources were improved by techniques like knowledge distillation, quantization, and model pruning, it would be more appropriate for deployment in actual IoT scenarios.

Acknowledgement: The authors thank for the environmental and finance support rendered by different affiliations of the article.

Funding Statement: This research was funded by the Ministry of Higher Education Malaysia, Fundamental Research Grant Scheme (FRGS), FRGS/1/2024/ICT07/UPNM/02/1.

Author Contributions: Conceptualization: Mariyam Aysha Bivi; Data Curation: Prasanalakshmi Balaji; Formal Analysis: Mahaveerakannan Renganathan; Funding Acquisition: Syarifah Bahiyah Rahayu; Investigation: Maode Ma and Prasanalakshmi Balaji; Methodology: Maode Ma and Prasanalakshmi Balaji; Project Administration: Syarifah Bahiyah Rahayu; Resources: Maode Ma and Zhaoxi Fang; Software: Syarifah Bahiyah Rahayu and Prasanalakshmi Balaji; Supervision: Syarifah Bahiyah Rahayu; Validation: Sangita Babu; Visualization: Zhaoxi Fang; Writing—Original Draft Preparation: Maode Ma and Prasanalakshmi Balaji; Writing—Review & Editing: Maode Ma. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Dataset 1: https://data.mendeley.com/datasets/bzf9jcvhx4/1 (accessed on 12 September 2024). Dataset 2: www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot (accessed on 12 September 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Ali MA, Al-Sharafi SAH. Intrusion detection in IoT networks using machine learning and deep learning approaches for MitM attack mitigation. Discov Internet Things. 2025;5(1):48. doi:10.1007/s43926-025-00104-w.
2. Tang D, Zhang S, Chen J, Wang X. The detection of low-rate DoS attacks using the SADBSCAN algorithm. Inf Sci. 2021;565(13):229–47. doi:10.1016/j.ins.2021.02.038.
3. Garcia N, Alcaniz T, González-Vidal A, Bernabe JB, Rivera D, Skarmeta A. Distributed real-time SlowDoS attacks detection over encrypted traffic using artificial intelligence. J Netw Comput Appl. 2021;173(34):102871. doi:10.1016/j.jnca.2020.102871.
4. Alashhab AA, Zahid MSM, Muneer A, Abdukkahi M. Low-rate DDoS attack Detection using deep learning for SDN-enabled IoT networks. Int J Adv Comput Sci Appl. 2022;13(11):371–7. doi:10.14569/IJACSA.2022.0131141.

5. Ali MN, Imran M, din MS, Kim BS. Low rate DDoS detection using weighted federated learning in SDN control plane in IoT network. Appl Sci. 2023;13(3):1431. doi:10.3390/app13031431.
6. Vedula V, Lama P, Boppana RV, Trejo LA. On the detection of low-rate denial of service attacks at transport and application layers. Electronics. 2021;10(17):2105. doi:10.3390/electronics10172105.
7. Ilango HS, Ma M, Su R. Low rate DoS attack detection in IoT-SDN using deep learning. In: Proceedings of the 2021 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics); 2021 Dec 6–8; Melbourne, Australia. doi:10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics53846.2021.00031.
8. Ain NU, Sardaraz M, Tahir M, Abo Elsoud MW, Alourani A. Securing IoT networks against DDoS attacks: a hybrid deep learning approach. Sensors. 2025;25(5):1346. doi:10.3390/s25051346.
9. Almaraz Rivera JG. The identification of DoS and DDoS attacks to IoT devices in software defined networks by using machine learning and deep learning models 2022 [Internet]. [cited 2025 Mar 10]. Available from: https://hdl.handle.net/11285/650696.
10. Kachavimath AV, Narayan DG. A deep learning-based framework for distributed denial-of-service attacks detection in cloud environment. Lect Notes Electr Eng. 2021;1:605–18. doi:10.1007/978-981-33-6977-1_44.
11. Sambangi S, Gondi L, Aljawarneh S. A feature similarity machine learning model for DDoS attack detection in modern network environments for Industry 4.0. Comput Electr Eng. 2022;100(1):107955. doi:10.1016/j.compeleceng.2022.107955.
12. Abiramasundari S, Ramaswamy V. Distributed denial-of-service (DDoS) attack detection using supervised machine learning algorithms. Sci Rep. 2025;15(1):13098. doi:10.1038/s41598-024-84879-y.
13. de Miranda Rios V, Inácio PR, Magoni D, Freire MM. Detection of reduction-of-quality DDoS attacks using fuzzy logic and machine learning algorithms. Comput Netw. 2021;186(6):107792. doi:10.1016/j.comnet.2020.107792.
14. Bakhsh SA, Khan MA, Ahmed F, Alshehri MS, Ali H, Ahmad J. Enhancing IoT network security through deep learning-powered intrusion detection system. Internet Things. 2023;24(10):100936. doi:10.1016/j.iot.2023.100936.
15. Wang J, Lei X, Jiang Q, Alfarraj O, Tolba A, Kim G. DoS attack detection based on deep factorization machine in SDN. Comput Syst Sci Eng. 2023;45(2):1727–42. doi:10.32604/csse.2023.030183.
16. Alabdulatif A, Thilakarathne NN, Aashiq M. Machine learning enabled novel real-time IoT targeted DoS/DDoS cyber attack detection system. Comput Mater Contin. 2024;80(3):3655–83. doi:10.32604/cmc.2024.054610.
17. Saiyed MF, Al-Anbagi I. Flow and unified information-based DDoS attack detection system for multi-topology IoT networks. Internet Things. 2023;24(1):100976. doi:10.1016/j.iot.2023.100976.
18. Dong S, Xia Y, Wang T. Network abnormal traffic detection framework based on deep reinforcement learning. IEEE Wirel Commun. 2024;31(3):185–93. doi:10.1109/MWC.011.2200320.
19. Ilango HS, Ma M, Su R. A feedforward-convolutional neural network to detect low-rate DoS in IoT. Eng Appl Artif Intell. 2022;114(7):105059. doi:10.1016/j.engappai.2022.105059.
20. Perez-Diaz JA, Valdovinos IA, Choo KKR, Zhu D. A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. IEEE Access. 2020;8:155859–72. doi:10.1109/ACCESS.2020.3019330.
21. Liu Z, Guo C, Liu D, Yin X. An asynchronous federated learning arbitration model for low-rate DDoS attack detection. IEEE Access. 2023;11(3):18448–60. doi:10.1109/ACCESS.2023.3247512.
22. Al-Fayoumi M, Al-Haija QA. Capturing low-rate DDoS attack based on MQTT protocol in software defined-IoT environment. Array. 2023;19(11):100316. doi:10.1016/j.array.2023.100316.
23. Pasha MJ, Rao KP, MallaReddy A, Bande V. LRDADF: an AI enabled framework for detecting low-rate DDoS attacks in cloud computing environments. Meas Sens. 2023;28:100828. doi:10.1016/j.measen.2023.100828.
24. Fu Y, Duan X, Wang K, Li B. Low-rate denial of service attack detection method based on time-frequency characteristics. J Cloud Comput. 2022;11(1):31. doi:10.1186/s13677-022-00308-3.
25. Tang D, Dai R, Tang L, Li X. Low-rate DoS attack detection based on two-step cluster analysis and UTR analysis. Hum Centric Comput Inf Sci. 2020;10(1):6. doi:10.1186/s13673-020-0210-9.

26. Yuvaraja T, Rajan Salem Jeyaseelan WG, Ashokkumar SR, Premkumar M. Detecting and mitigating low-rate DoS and DDoS attacks: multimodal fusion of time-frequency analysis and deep learning model. Teh Vjesn. 2024;31(2):495–501. doi:10.17559/TV-20230613000728.
27. Rostami SMH, Pourgholi M, Asharioun H. Enhancing resilience of distributed DC microgrids against cyber attacks using a transformer-based Kalman filter estimator. Sci Rep. 2025;15(1):6815. doi:10.1038/s41598-025-90959-4.
28. Yu J, Zhou X. One-dimensional residual convolutional autoencoder based feature learning for gearbox fault diagnosis. IEEE Trans Ind Inform. 2020;16(10):6347–58. doi:10.1109/TII.2020.2966326.
29. Sharma H, Kumar P, Sharma K. Recurrent neural network based incremental model for intrusion detection system in IoT. Scalable Comput Pract Exp. 2024;25(5):3778–95. doi:10.12694/scpe.v25i5.3004.
30. Trojovska E, Dehghani M, Trojovsky P. Fennec fox optimization: a new nature-inspired optimization algorithm. IEEE Access. 2022;10(4):84417–43. doi:10.1109/ACCESS.2022.3197745.
31. Abdollahzadeh B, Soleimanian Gharehchopogh F, Mirjalili S. Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Int J Intell Syst. 2021;36(10):5887–958. doi:10.1002/int.22535.
32. Harifi S, Khalilian M, Mohammadzadeh J, Ebrahimnejad S. Emperor penguins colony: a new metaheuristic algorithm for optimization. Evol Intell. 2019;12(2):211–26. doi:10.1007/s12065-019-00212-x.
33. Serani A, Diez M. Dolphin pod optimization. In: Lecture notes in computer science. Berlin/Heidelberg, Germany: Springer; 2017. p. 50–62.