RESEARCH ARTICLE

An Enhanced LSTM Approach for Detecting IoT-Based DDoS Attacks Using Honeypot Data

Arjun Kumar Bose Arnob[1] · M. F. Mridha[1] · Mejdl Safran[2] · Md Amiruzzaman[3] · Md. Rajibul Islam[4]

Received: 22 October 2024 / Accepted: 13 January 2025
© The Author(s) 2025

Abstract
One of the widening perils in network security is the Distributed Denial of Service (DDoS) attacks on the Internet of Things (IoT) ecosystem. This paper presents an enhanced Intrusion Detection System (IDS) through the proposal of an enhanced version of the long short-term memory (LSTM) model to detect DDoS attacks using honeypot-generated data. The proposed model aggregates the Conv1D, Bidirectional Long Short-Term Memory (Bi-LSTM), Bidirectional Gated Recurrent Unit (Bi-GRU), and dropout layers to extract temporal and spatial features from IoT traffic effectively. We tested the efficacy of the proposed system on a real-world IoT-DH dataset, which showed a remarkable accuracy of 99.41%, with an AUC score of 0.9999. A comparative analysis with other baseline models, such as LSTM, Bidirectional LSTM (Bi-LSTM), Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Feedforward Neural Network (FNN), and Temporal Convolutional Network (TCN), proved that enhanced LSTM outperformed the other models. This indicates the robustness of the proposed model in correctly detecting DDoS attacks with high generalization capability for unseen traffic data. The contribution of this paper will be an addition to the deep learning techniques applied for the solution of intrusion detection systems (IDS), which will also allow the building and implementation of more efficient security mechanisms in IoT environments.

Keywords IoT · DDoS attacks · Intrusion detection · Enhanced LSTM · Honeypot · IoT-DH Dataset · Cybersecurity

1 Introduction

The exponential increase in the number of Internet of Things (IoT) devices is changing in many fields, allowing them to connect and automate in real-time. Simultaneously, a wide range of security challenges has emerged. Among the greatest threats are Distributed Denial of Service (DDoS) attacks, which are among the top targets frequently exploiting the intrinsic vulnerabilities of poor protection in IoT devices [1]. One such example was the Mirai botnet attack in 2016, wherein hacked IoT devices surged to unprecedented levels of traffic. This attack demonstrates how easily IoT devices with default or weak credentials can be compromised and used as weapons [2].

With the continued expansion of IoT networks, the substantially enhanced leverage of attackers is owing to limited computational resources and inadequate security mechanisms. It is projected that 24.6 billion IoT devices will be connected by 2025, thereby increasing the potential for DDoS assaults [3]. Owing to their complexity, these assaults are now considered multidimensional threats that use botnets to impede the detection processes by embedding malicious and legitimate communications [4]. IoT networks are vulnerable to DDoS assaults, such as UDP and TCP SYN flooding, owing to the increasing number of devices linked to these networks [5].

https://doi.org/10.1007/s44196-025-00741-7

Published online: 05 February 2025

The outcomes of DDoS attacks against IoT systems go far beyond the simple disruption of services, causing huge losses at both operational and economic levels. These losses increase operational costs, service outages, and the erosion of consumer confidence [6]. These evolving threats in turn raise the urgent need for an enhanced, adaptable, and scalable detection system capable of effective mitigation within the IoT ecosystem [7].

Traditional Intrusion Detection Systems (IDS) rely heavily on signature-based detection and are increasingly inadaptable to modern DDoS attacks. Traditional security methods make it difficult to detect new and evolving threats that do not match the previously known patterns. As modern DDoS attacks have become extremely complex and dynamic, advanced IDS solutions have become crucial for ever-growing IoT networks [8, 9].

Such issues can now be solved using Deep Learning (DL), which is becoming an extremely useful tool. Unlike traditional methods, In IoT contexts, DL models are more successful at spotting subtle and changing threat vectors, because they can autonomously learn complicated patterns and features from raw data [10]. In particular, DL methods oriented towards sequence-based data are capable of ensuring high adaptability and precision in detecting sophisticated DDoS attacks, even in highly variable environments with diverse traffic flows [11].

Honeypots are essential for improving the precision and effectiveness of DL models for IoT-based DDoS detection. Honeypots attract malicious traffic because they operate as decoy systems [12]. Researchers should gather real-world attack data and examine the attacker's tactics, techniques, and procedures (TTPs) [13]. Consequently, tuning the system in real-world attack scenarios using a combination of honeypot data and LSTM-based models improves the performance of the model over real-time threat detection in binary classification tasks [14].

Another valuable application of honeypots is solving the challenge of dataset limitations in IoT networks. The amount of data captured by honeypots is rich in malicious traffic patterns, which can be useful for enhancing the generalization capabilities of DL models. By embedding honeypot data in DL models, the detection of DDoS attacks can be notably improved while providing real-time traffic patterns for the training and evaluation processes [15].

The integration of honeypots with Generative Adversarial Networks (GANs) enhances DL detection. GANs are adopted for synthesizing attack traffic, which highly resembles reality, enabling models to enhance the detection precision of unseen attack vectors [16]. Thus, the integration of honeypots with GANs forms a strong framework for the timely detection and classification of DDoS attacks in real-time, which can be adapted to evolving methods by perpetrators [17].

This paper presents an enhanced LSTM-based IDS for IoT to detect DDoS attacks. Fundamentally, the Long Short-Term Memory (LSTM) model is suitable for this task because it can learn both temporal dependencies within a sequence and recognize patterns that vary over time [18]. The proposed enhanced LSTM was compared with several DL architectures, including baseline architectures such as LSTM, Bidirectional LSTM (Bi-LSTM), Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Feedforward Neural Network (FNN), and Temporal Convolutional Network (TCN). These models were chosen to improve the detection of DDoS assaults because they can handle sequential data and capture different features of traffic patterns [19]. This study examined the performance of multiple models in identifying the optimal DL architecture for real-time DDoS detection in IoT scenarios.

Developing an Intrusion Detection System (IDS) capable of rapidly and accurately detecting DDoS attacks in real-time is crucial to address the rapid proliferation of IoT attacks in the modern world. The use of honeypots has empowered the system to sniff real-world attack scenarios, thereby enriching DL models with authentic malicious traffic data [20]. Through several steps such as filtering, normalization, and feature extraction, the preprocessed data become a high-quality input for model training. The IoT-DH dataset is used to train and test the models. This is based on the information gathered from honeypots [21].

The main contributions of this study are as follows:

- **Proposed Enhanced LSTM Model:** We introduce an enhanced Long Short-Term Memory (LSTM) model that integrates Conv1D, Bidirectional LSTM (Bi-LSTM), Bidirectional Gated Recurrent Unit (Bi-GRU), and dropout layers. This architecture effectively captures both temporal and spatial features from IoT traffic to detect DDoS attacks with a high precision.

- **Utilization of Honeypot-Generated Data:** By leveraging the publicly available IoT-DH dataset, which contains real-world IoT traffic data generated by honeypots, the model achieves robust performance in detecting both known and unknown DDoS attacks.
- **Minimizing False Positives:** The model architecture focuses on reducing false positive rates, ensuring that legitimate IoT traffic is not misclassified as malicious.
- **Scalability and Adaptability:** The model is designed to operate effectively in dynamic IoT environments, providing real-time threat detection suitable for diverse and large-scale networks.
- **Comprehensive Performance Evaluation:** Detailed benchmarking against traditional baseline deep learning models demonstrates the model's superior performance across metrics such as accuracy, precision, recall, and $F1$-score.

This work advances an LSTM-optimized IDS tailored to the IoT. The proposed approach uses the IoT-DH dataset, which is generated from honeypot environments simulating real-world IoT devices, to ensure that the dataset reflects robust real-world attack scenarios. Instead of directly deploying honeypots, this approach leverages honeypot-generated data to provide diverse and authentic datasets for training and evaluation. The two main goals of using honeypot data are to make real-world attack data robust and efficiently identify complicated DDoS attacks in IoT networks. The emphasis of this method on scalability also contributes to its application in various IoT settings, where safety is a primary concern. Finally, a few metrics that indicate how well the system performs in terms of high detection rate and minimal false positives are discussed, including recall, accuracy, precision, and $F1$-Score.

The remainder of this paper is organized as follows: Related studies on DL, honeypot-based IDS, and IoT security are discussed in Sect. 2. The research plan and techniques for the DL models used in this study are presented in Sect. 3. Section 4 presents the results and analysis of this study as well as an assessment of the proposed model. Section 5 summarizes the study and provides recommendations for future research to strengthen IoT security.

2 Related Work

Devices have proliferated rapidly in several industries, including banking, transportation, smart cities, healthcare, and industrial automation. However, this growth has been accompanied by serious security vulnerabilities. Among them, the most threatening are DDoS attacks based on the principle of interconnectedness that characterizes IoT devices [22]. Attackers can severely interrupt services and cause operational failures by flooding target systems with excessive traffic and by controlling a large number of infected devices. Owing to their distinctive qualities, such as high interconnectedness, varied communication protocols, and limited processing power, IoT networks are particularly susceptible to these types of attacks [23, 24].

2.1 Deep Learning Approach for DDoS Attacks

DDoS attacks are more likely to occur in IoT networks and researchers are investigating more sophisticated DL methods for IDS. The majority of these LSTM-based DL models perform exceptionally well in binary classification tasks, where the main goal is to distinguish between malicious and benign traffic [25].

For the binary classification of DDoS detection, LSTM networks can detect minute and subtle irregularities in network traffic that can potentially be signs of an impending assault [26]. The effectiveness of the LSTM model in identifying a botnet-based DDoS attack was demonstrated by [27], who trained the model using network traffic data to identify the long-term interdependence of the attack patterns. Comparably, [28] proposed combining LSTM with sophisticated feature selection techniques, which significantly increased detection accuracy in scenarios involving binary classification.

The ability of conventional LSTM models to learn from both past and future states of network traffic data is expanded by Bi-LSTM networks. This is particularly helpful in identifying increasingly complex DDoS attacks in which malevolent actions are masked by legitimate traffic to avoid detection. It has been discovered that using Bi-LSTM increases the detection rates and accuracy of distinguishing between legitimate and malicious communications [29, 30].

Other hybrid DL architectures that combine LSTM with convolutional neural Networks have been proven to improve detection accuracy. Although the CNN structure is superior in extracting spatial features of traffic data, LSTM looks after temporal dependencies, thereby providing a holistic approach for DDoS detection in IoT environments [31]. This type of hybrid model functions well in binary classification, where the challenge is to distinguish between benign and malicious patterns in a highly dynamic IoT network [32].

2.2 Hybrid DDoS Detection Frameworks

In addition to using separate LSTM models, hybrid approaches have also been employed to increase the overall efficacy of IoT IDS. These strategies integrate additional detection techniques, such as anomaly- or signature-based detection, with deep learning. For example, employing LSTM hybrid models with a CNN or any other DL architecture has been shown to boost detection accuracy and adaptability to novel attack patterns [33].

Combining the two methods offers the following benefits, as mentioned by [34] while signature-based detection is significantly more effective at identifying known threats, anomaly based detection is useful for identifying unexpected attacks. These hybrid models will be more thorough for DDoS detection in IoT scenarios because they consider both known and undiscovered threats.

Dynamic feature selection methods can potentially contribute significantly to enhancing the accuracy of DL models during the DDoS detection process. These methods optimize feature selection and reduce computational overhead without compromising detection performance, thereby making deep-learning models more efficient in resource-constrained IoT environments [35].

However, several issues persist with the rapid development of DL-based DDoS detection methods. The heterogeneity of IoT devices, as depicted by their diverse hardware capabilities, operating systems, and communication protocols, necessitates the development of a one-size-fits-all DL model for DDoS detection. Meanwhile, various dynamic factors have caused continuous addition and removal in IoT networks, raising different questions regarding model scalability and adaptation [36].

Another critical challenge is the unavailability of labeled datasets specific to IoT traffic. Most open-source datasets are not sufficiently comprehensive to fully represent the diversity of IoT devices or the dynamic nature of DDoS attack vectors [37, 38]. This further constrains the generalization capability of the DL models trained on such data. Therefore, future research should focus on semi-supervised or unsupervised learning techniques that would reduce the dependence on labeled data and allow for adaptive and scalable models in IoT security [39].

In addition, high-interaction honeypots may provide more insights into the methodologies of attackers, and thus, could further enhance the efficiency of deep-learning-based IDSs. Real-time DDoS attack detection and mitigation can be facilitated by combining cloud-based DL systems with honeypot data to provide scalable and affordable solutions [40].

3 Research Plan and Methodology

The construction and assessment of the suggested enhanced LSTM model for DDoS assaults in IoT systems using DL models are the main topics of this section. The steps involved in conducting this research were as follows: gathering data, preparing datasets, developing models, and evaluating the models.

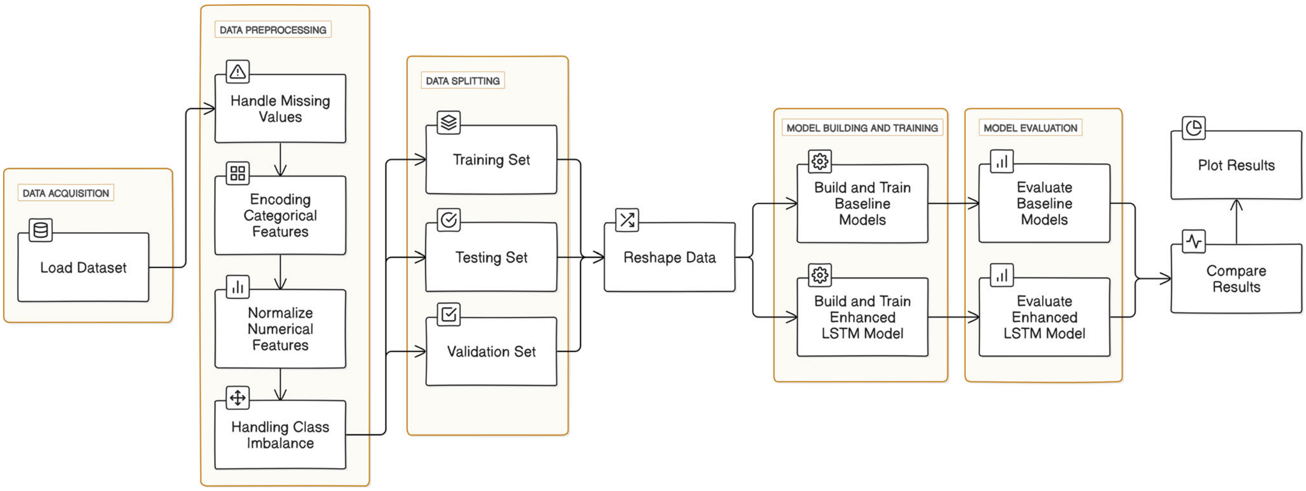

Fig. 1 Overall flow of methodologies of proposed research

The entire workflow for the IDS in IoT-based DDoS attack detection is depicted in Fig. 1 as follows: data collection from IoT-DH data generated by honeypots, followed by preprocessing that handles missing values and involves encoding and normalization. Following preprocessing, the data were split into sets for testing, validation, and training. Better temporal and spatial feature extraction was made possible in this study by utilizing an enhanced LSTM architecture in which a Conv1D layer was merged with Bidirectional LSTM and GRU layers. To regularize the model, layers for dropout and batch normalization were included.

3.1 Dataset and Preprocessing

3.1.1 Dataset

It is publicly available for training and testing. The IoT-DH dataset is a labeled network traffic dataset with 15,817,209 records, labeled as either benign or malicious. Several key features of the dataset, such as the packet count, byte count, duration, and protocol type, can be useful for analyzing patterns related to DDoS attacks. The honeypot environment emulates common IoT devices for luring attackers to capture comprehensive data sets related to different attack vectors. Feature information, such as packet count, byte count, flow rate, and protocols, facilitates the identification of patterns that are crucial for detecting and mitigating DDoS attacks [21].

Table 1 provides the main characteristics of the dataset concerning the span and variety of features in the intrusion detection model. For example, features such as $PKTRATE$, $BYTEPERFLOW$, SRC, DST, and $LABEL$ in a dataset are important for classifying normal and malicious traffic. Thus, the fine-grained features used in the proposed IDS can capture attack patterns with a high granularity. This makes it robust to sophisticated DDoS attack detection.

3.1.2 Representativeness of the IoT-DH Dataset

This dataset includes the traffic generated from different IoT scenarios, considering both benign and malicious activities, including different DDoS attack vectors such as UDP and TCP SYN floods. Therefore, the comprehensiveness inherent in the dataset will enable model testing across diverse traffic patterns, which is very important for developing robust and generalized models. Moreover, with real-world attack scenarios and realistic traffic in the IoT-DH dataset, the gap between simulated and practical environments decreases, ensuring representativeness in dynamic IoT ecosystems.

Table 1 Dataset features and unique values

Feature name	Unique values	Description
DT	558,200	Timestamp of the flow
SWITCH	11	Switch ID
SRC	2914	Source IP address
DST	3014	Destination IP address
PKTCOUNT	5,383,039	Packet count
BYTECOUNT	15,012,780	Byte count
DUR	558,180	Duration of the flow (ms)
DUR_NSEC	1,000	Flow duration in nanoseconds
TOT_DUR	561,523	Total flow duration
FLOWS	12,521	Number of flows
PACKETINS	533,375	Number of packets per flow
PKTPERFLOW	896,732	Packets per flow
BYTEPERFLOW	15,201,358	Bytes per flow
PKTRATE	1,523,889	Packet rate
PAIRFLOW	12,438	Paired flows between two entities
PROTOCOL	3	Protocol type (e.g., TCP, UDP)
PORT_NO	28,304	Port number
TX_BYTES	12,258	Transmitted bytes
RX_BYTES	1,800	Received bytes
TX_KBPS	15,015,909	Transmission rate (KBps)
RX_KBPS	13,735,382	Reception rate (KBps)
TOT_KBPS	13,735,911	Total rate (KBps)
LABEL	2	Attack or normal (classification)

3.1.3 Preprocessing

To ensure data quality, the following preprocessing steps were applied:

- **Handling Missing Values:** Any Missing data entries were imputed using median values to preserve the integrity of the dataset.
  - Missing values were imputed using medians. This choice was made after conducting the Shapiro–Wilk normality test, which showed that the key numerical features (pktcount, bytecount, pktperflow, pktrate, etc.) were not normally distributed (p-values < 0.05).
  - Visual inspection using histograms and KDE plots confirmed skewed distributions with potential outliers.
  - The median was chosen because it is more robust than the mean in handling skewed data and outliers, thereby ensuring that the imputation process does not distort the dataset.
- **Encoding Categorical Features:** Categorical variables, such as the protocol types, were one-hot encoded to transform them into a numerical format suitable for deep learning models.
- **Normalizing Numerical Features:** Normalization was applied to continuous characteristics, such as packet rate and bytes per flow, to guarantee that each feature contributed equally to the model training process.
- **Handling Class Imbalance:** Addressing common class imbalance issues in intrusion detection datasets is critical for improving the model performance. In this study, the dataset exhibited a significant imbalance between benign (63,561 records) and malicious traffic (40,784 records) traffic. Techniques such as the synthetic minority over-sampling technique (SMOTE) were utilized to artificially generate instances of the minority class ensuring a more balanced representation of both classes as presented in Table 2. SMOTE generates synthetic samples by interpolation within the feature space of the minority class, thereby reducing the risk of biased outcomes

Table 2 Dataset traffic record numbers

Traffic Type	Record Count
Total Records	15,817,209
Benign Records	63,561
Malicious Records	40,784
Training Set Records	11,072,046
Validation Set Records	2,372,581
Test Set Records	2,372,582

caused by underrepresented classes. After applying SMOTE, the class distribution was effectively balanced, which not only mitigated the impact of class imbalance but also enhanced the intrusion detection ability of the model across all traffic types, thereby improving the overall generalization and detection performance [2, 3, 27, 32].

The pre-processing pipeline involves encoding categorical features, scaling numerical attributes, and encoding target labels to prepare the dataset for training. The final dataset consisted of 23 features, as listed in Table 1. The dataset contained 15,817,209 traffic records, with 63,561 benign records and 40,784 malicious records, as presented in Table 2. After preprocessing, class weights were computed to address the class imbalance, resulting in weights of 82.95, 129.28, and 0.34 for benign, malicious, and additional classes, respectively.

3.2 Data Splitting

The dataset was preprocessed to prepare for model training, after which it was split into three distinct subsets: the training, test, and validation sets. The splits are in a ratio of 70% for training, 15% for validation, and 15% for testing, ensuring that the majority will be retained for training the models with the respective record counts listed in Table 2, the training set consists of (11,072,046 records), validation and test set consist of respectively (2,372,581 records). However, only a small portion was retained for model testing and performance evaluation. Stratification ensures that the created subsets provide the same attack-to-normal instance ratio as that expected from the original dataset. This ensures the conservation of the distribution of classes within all subsets. This was done with care in splitting the data to show more realistic and honest results for better model evaluations that would minimize any risk of bias due to the imbalance in class distribution.

3.3 Model Building and Training

3.3.1 Baseline Models:

Several baseline deep learning models were implemented for comparison with the proposed enhanced LSTM model. The specific architecture details of each baseline model are as follows:

- **LSTM (Long Short-Term Memory)**: A single LSTM layer with 128 units, followed by a dense layer with 64 units.
- **GRU (Gated Recurrent Unit)**: A single GRU layer with 128 units, followed by a dense layer with 64 units.
- **RNN (Recurrent Neural Network)**: A single simple RNN layer with 128 units, followed by a dense layer with 64 units.
- **FNN (Feedforward Neural Network)**: Two dense layers, the first with 128 units using *ReLU* activation, followed by a dense layer with 64 units.
- **Bidirectional LSTM**: A Bidirectional LSTM layer with 128 units in each direction (forward and backward) followed by a dense layer with 64 units.

- **TCN (Temporal Convolutional Network)**: Stacked 1D convolutional layers (*Conv1D*) with 128 filters, followed by a *maxpooling* layer and a dense layer with 64 units.

The architecture of each model varied, but the following configuration details were common across all models:

Activation Function: The hidden layers used the *tanh* activation function and the output layer used a *sigmoid* activation function for binary classification.

Regularization: A dropout layer with a rate of 0.3 was applied to prevent overfitting.

Optimizer: All the models were trained using the Adam optimizer with a learning rate of 0.001.

Training Configuration: The models were trained using binary cross-entropy loss with early stopping and learning rate reduction (ReduceLROnPlateau) to enhance generalization.

Hyperparameter Tuning: Table 3 provides an overview of the hyperparameters of the baseline models, which include the LSTM, GRU, RNN, Bi-LSTM, FNN, and TCN. All the models used the Adam optimizer, configured with a learning rate of 0.0003, and adopted binary cross-entropy as the loss function. The LSTM, GRU, RNN, and Bi-LSTM models had 64 units in their recurrent layers, with Bi-LSTM using bidirectional connections to draw representations from both forward and backward sequences. The FNN and TCN models had architectures with dense and convolutional layers, respectively, with different variations. All models were trained with a batch size of 64 for 20 epochs, with callbacks to reduce the learning rate and stop the training process early for optimal convergence. These were chosen to ensure that the test was uniform and fair among the models.

3.3.2 Enhanced LSTM Model

The architecture of the proposed enhanced LSTM is shown in Fig. 2, and the pseudocode of the model is presented in Algorithm 1. The proposed architecture makes fundamental changes to the base architecture concerning the effective capture of the spatial and temporal complexities of IoT traffic data. While the basic LSTM model may work well for capturing temporal dependencies, it almost fails to attend to the important task of capturing the spatial correlations, which are of prime importance for finding the patterns associated with DDoS attacks [28]. This will extend such design features by introducing convolutional layers, bidirectional recurrent layers, and dropout regularization to compensate for these deficiencies. Such combinations allow the model to detect DDoS attacks with high precision and recall while maintaining low computation. Major architectural improvements include the following:

- **Spatial Feature Extraction via Conv1D:** A convolutional one-dimensional layer, whereby the input sequence undergoes processing to capture localized spatial dependencies in the data. This is instrumental in network traffic analysis, whereby important patterns tend to spread over several consecutive time steps. In essence, it can understand a localized pattern, which may indicate an attack because it has early feature extraction of spatial features in the architecture [15, 23]. The Batch Normalization layer stabilizes the gradient flow and promoted faster convergence during training. Further dimensional and computational reduction was performed with the MaxPooling1D layer, which retains important features but eliminates redundant data points.
- **Bidirectional LSTM for Temporal Dynamics:** Bidirectional LSTM with 128 units enables the network to learn long-range dependencies in both forward and backward directions. Traditional LSTMs normally process sequential data in one direction from the past to the future [27]. However, this may not highlight the pattern well when considering both temporal directions. By leveraging a bidirectional approach, the model enhances its capacity to detect subtle patterns in the flow of IoT network traffic that usually get missed out; then, a dropout layer after the LSTM-the rate is 0.3 to avoid overfitting, a common issue arising in deep models trained on a complex dataset [29, 30].
- **Bidirectional GRU:** A Bidirectional Gated Recurrent Unit (GRU) layer with 128 units was incorporated to further enhance temporal pattern recognition.Similar to LSTMs, GRUs have simpler gate mechanisms and are

Table 3 Hyperparameters of baseline models

Model Architecture	Hyperparameter	Value
LSTM	LSTM units	64
	Optimizer	Adam
	Learning rate	0.0003
	Loss function	Binary Crossentropy
	Batch size	64
	Epochs	20
GRU	GRU units	64
	Optimizer	Adam
	Learning rate	0.0003
	Loss function	Binary Crossentropy
	Batch size	64
	Epochs	20
RNN	RNN units	64
	Optimizer	Adam
	Learning rate	0.0003
	Loss function	Binary Crossentropy
	Batch size	64
	Epochs	20
Bi-LSTM	Bi-LSTM units	64
	Optimizer	Adam
	Learning rate	0.0003
	Loss function	Binary Crossentropy
	Batch size	64
	Epochs	20
FNN	Dense layers	[128, 64]
	Optimizer	Adam
	Learning rate	0.0003
	Loss function	Binary Crossentropy
	Batch size	64
	Epochs	20
TCN	Conv1D filters	[64, 128]
	Conv1D kernel size	3
	Optimizer	Adam
	Learning rate	0.0003
	Loss function	Binary Crossentropy
	Batch size	64
	Epochs	20

therefore computationally more efficient. This architecture helps in combining the LSTM and GRU layers to achieve performance in places where there is a need for computational efficiency [18], as inherently there is in constrained IoT environments. It is further used with dropout regularization and Batch Normalization for GRU to be robustly generalized to unseen data.

- **Dense Layer for Nonlinear Feature Aggregation:** After processing through the recurrent layers, the model applies a fully connected (dense) layer with 256 units and ReLU activation. This layer is used for feature aggregation, that is, bringing into one set the learned spatial and temporal features from the previous layers to prepare them for a classification task. In addition, adding another dropout layer, this time at a rate of 0.3, enhances the generalization of the model without overfitting the training data.

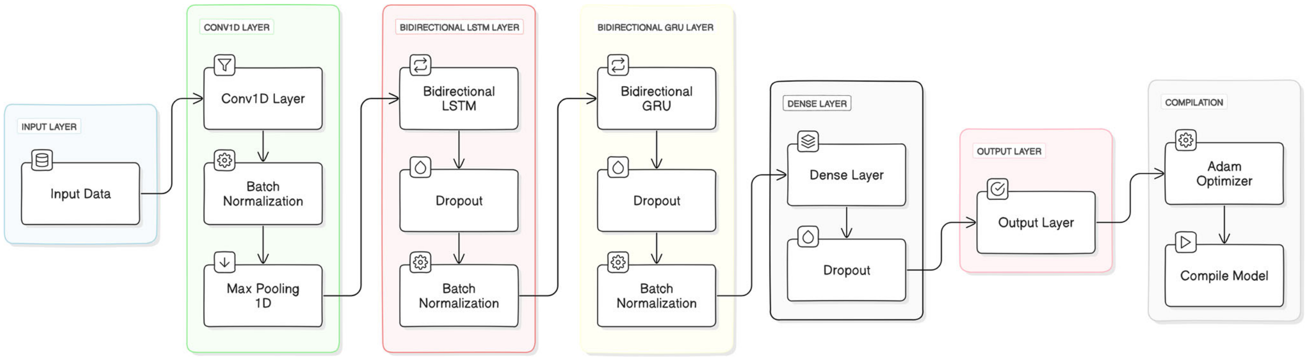

Fig. 2 Proposed Enhanced LSTM Model

- **Output Layer for Binary Classification:** The output layer comprises only one unit using a sigmoid function as its activation function, tailored for binary classification. This configuration allows the model to accurately classify network traffic as either benign or indicative of DDoS attack. Architecture such as this makes this model capable of even detecting subtle anomalies in traffic patterns that may signal the onset of an attack.

The entire architecture is trained using an Adam optimizer, whose learning rate is set to 0.0003 because it presents a good tradeoff between fast convergence and the stability of the model. In addition, it has been used as a binary cross-entropy function to optimally differentiate whether the traffic is benign or malicious. Accordingly, the accuracy is the main metric for training and evaluation.

3.4 Training and Evaluation

3.4.1 Training Configuration

We trained our model with a batch size of 64, which means that the model processed 64 samples at one time before updating the parameters. To provide sufficient time for the model to obtain a proper feel for the data, we set the maximum number of training epochs to 20, so that the model could iterate over the entire dataset several times. However, to prevent the overfitting of the model, an early stop was applied. Early stopping helps prevent overfitting by checking the validation loss during training. If the validation loss does not change for 15 epochs in a row, the training will stop automatically, which shows that the model has stopped improving and may start overfitting the training data. This is one method to ensure that the model generalizes well to new data and prevents unnecessary overtraining.

In addition to the early stopping, the learning rate of the model dynamically varied during training using the ReduceLROnPlateau callback. This callback is a very useful tool for further enriching the model performance and preventing possible issues such as slow convergence. If the validation loss did not improve within the sequences of five epochs, the learning rate was reduced by a factor of 0.5. A smaller learning rate forces the model to take smaller steps during the gradient descent to make finer adjustments to the model weights as it approaches an optimal solution. The gradual reduction in the learning rate avoids overshooting of the minima and results in better generalization. This process continues until the learning rate reaches a minimum threshold predefined as 1e-6. This threshold prevents the learning rate from becoming too small, which results in negligible parameter updates and stagnation of the optimization process. Setting a lower bound ensures that the model continues to learn effectively throughout the training process while benefiting from the reduced learning rates during the later stages of convergence.

Algorithm 1 Pseudocode of enhanced LSTM model

```
Input: Preprocessed honeypot dataset $D = \{x_1, x_2, ..., x_n\}$, input shape $input\_shape$, number of classes $num\_classes$,
       training parameters (epochs, batch size, learning rate).
Output: Trained Enhanced LSTM model $M$, classification results with metrics.
Step 1: Define Model Architecture
    Initialize Sequential model $M$
    Add Conv1D layer with 128 filters, kernel size 3, ReLU activation, and input shape $input\_shape$
    Apply Batch Normalization
    Add MaxPooling layer with pool size 1
    Add Bidirectional LSTM layer with 128 units (return sequences enabled)
    Apply Dropout with rate 0.3
    Apply Batch Normalization
    Add Bidirectional GRU layer with 128 units
    Apply Dropout with rate 0.3
    Apply Batch Normalization
    Add Dense layer with 256 units and ReLU activation
    Apply Dropout with rate 0.3
    Add Dense output layer with $num\_classes$ units and sigmoid activation
Step 2: Train the Model
    Split $D$ into training ($D_{train}$), validation ($D_{val}$), and testing ($D_{test}$) sets
    foreach epoch in total epochs do
        foreach batch $B \subseteq D_{train}$ do
            Perform forward propagation for $B$
            Compute loss using binary cross-entropy
            Perform backpropagation and update weights using Adam optimizer
        end
        Evaluate $M$ on $D_{val}$ and log metrics (accuracy, loss)
        if validation loss increases for n consecutive epochs then
            Apply early stopping
        end
    end
Step 3: Evaluate the Model
    Evaluate $M$ on $D_{test}$ to compute metrics: accuracy, precision, recall, $F1$-score, AUC-ROC
Step 4: Real-Time Detection
    foreach incoming IoT network packet $p$ do
        Preprocess $p$ (normalize and reshape for $input\_shape$)
        Obtain prediction from $M(p)$
        if $M(p)$ detects a DDoS attack then
            Trigger alert and mitigation mechanisms
            Log attack details
        end
        else
            Log as benign traffic
        end
    end
Step 5: Release Resources
    Free memory and close datasets
```

Table 4 Evaluation metrics and their definitions

Metric	Definition
Accuracy	Proportion of correctly classified instances: Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$
Precision	Number of true positives divided by the sum of true and false positives: Precision $= \frac{TP}{TP+FP}$
Recall	Number of true positives divided by the sum of true positives and false negatives: Recall $= \frac{TP}{TP+FN}$
$F1$-Score	Harmonic mean of precision and recall, providing a balanced measure when there is a class imbalance: $F1$-Score $= 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$

3.5 Evaluation Metrics

Table 4 summarizes the definitions of these evaluation metrics, which serve as the basis for performance benchmarking across various models.

4 Result and Analysis

The performance evaluation of the proposed enhanced LSTM model in comparison with various baseline models, including LSTM, GRU, RNN, TCN, FNN, and Bidirectional LSTM, is discussed in this section. A range of performance indicators, including accuracy, precision, recall, $F1$-score, confusion matrix, ROC curves, training and validation accuracy, and loss curves, was employed to examine the efficacy of each model in identifying DDoS attacks targeting IoT traffic.

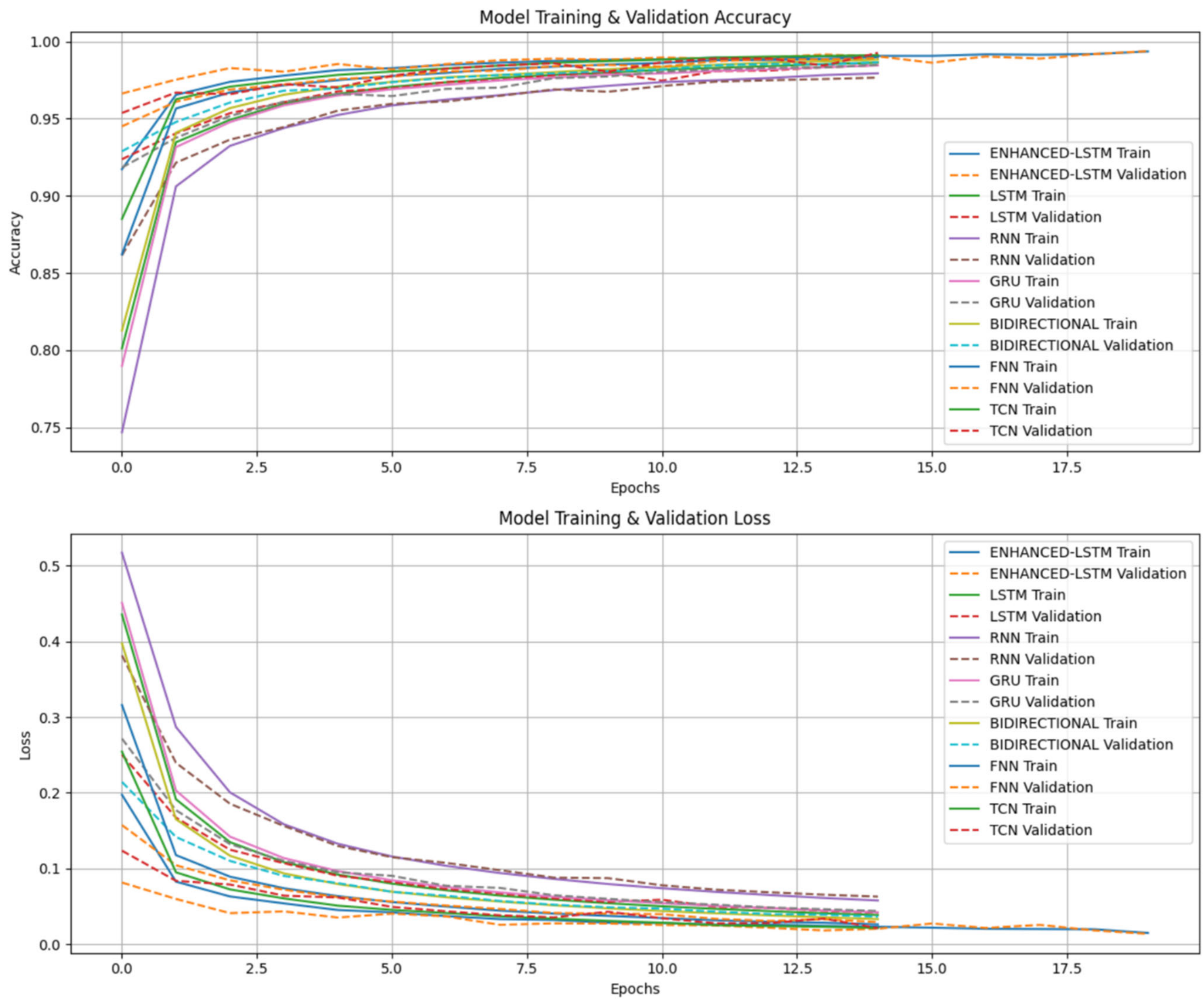

Fig. 3 Learning curves for all the models

4.1 Training and Validation Curves

Figure 3 provides insight into the convergence behavior of the models during training. These curves illustrate the learning behavior of the models over time and their generalization capabilities for unseen validation data.

Smooth and steady convergence was observed in the enhanced LSTM model for the training and validation accuracy. It achieved almost complete accuracy within 15 epochs without significant overfitting. Further confirmation of good generalization is the early stabilization of the validation loss for the enhanced LSTM. Moreover, the model was set to train for up to 20 epochs; however, it was planned to stop early after 15 epochs if it reached its plateau concerning the validation loss. This means that overfitting was avoided, and the model was strong and did not become overtrained.

Although the validation accuracy of the GRU model fluctuated slightly more than that of the improved LSTM model, both the models behaved similarly. Both models have a stabilization accuracy of approximately 95%, which indicates a good generalization. However, the somewhat larger validation loss when compared to the enhanced LSTM is indicative of their inferior capacity to represent the temporal relationships present in the IoT traffic data.

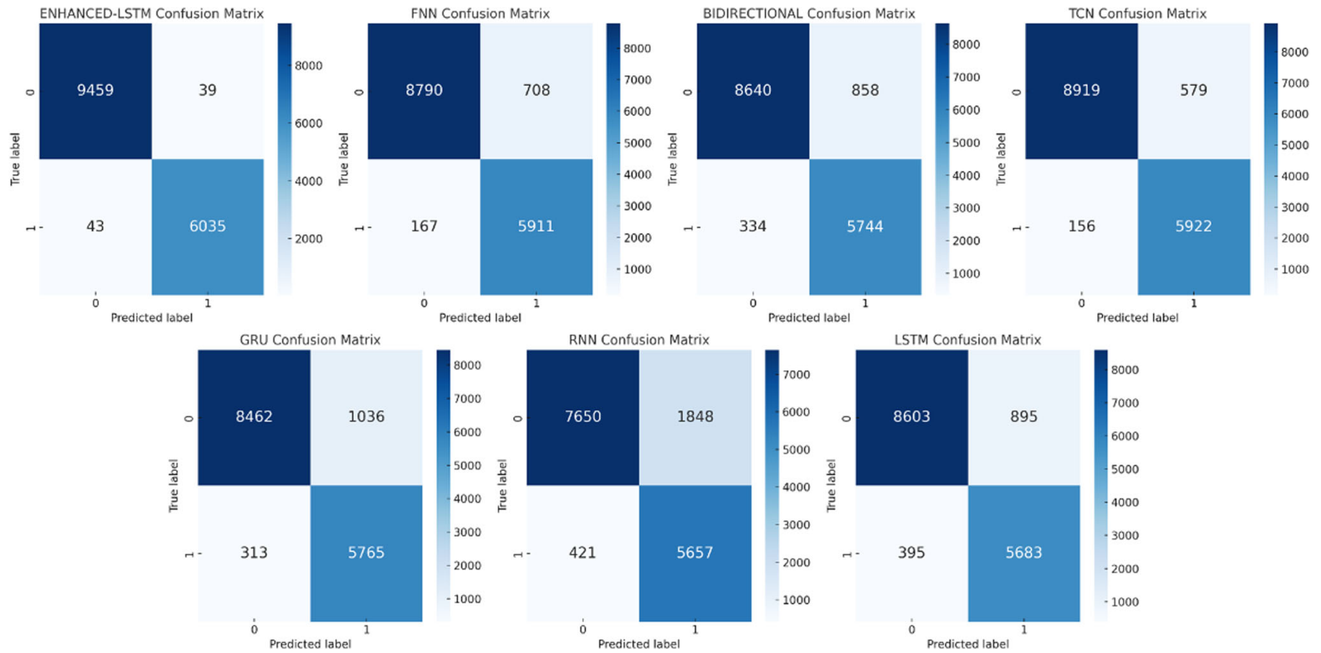

Fig. 4 Confusion Matrices for all the models

Although the RNN model attempted to learn from the data, it barely competed for good performance. The training curves showed slow convergence, with very low accuracy and high validation loss throughout the process. At approximately the 10th epoch, the model started to overfit, as shown by the widening gap between the training and validation accuracies. This divergence indicates overfitting of the RNN to the training data but poor generalization of the unseen validation data. These results highlight the limitation of the RNN architecture in handling complex temporal patterns that may be present within the dataset and its inability to perform well in cases with high generalization requirements.

Both the TCN and Bi-LSTM models exhibited strong convergence, achieving approximately 95% validation accuracy within 15 epochs. The bidirectional LSTM demonstrated a slightly lower validation loss, suggesting potentially more efficient learning. Although both networks generalized well, their performance was marginally inferior to that of the Enhanced-LSTM.

Although the FNN model eventually converged during training, it exhibited a higher validation loss than the other temporal models. It achieved an accuracy of approximately 94% accuracy, indicating its inability to effectively capture the sequential and temporal nature of the IoT traffic data. Consequently, its performance was significantly surpassed by that of the Enhanced-LSTM and other temporal models.

4.2 Confusion Matrices for Different Architectures

The confusion matrices in Fig. 4 provide an essential depiction of each model's classification performance to differentiate between benign and assault cases. Whereas the diagonal values in each matrix indicate the total number of samples that have been correctly categorized, the off-diagonal values indicate misclassification.

The enhanced LSTM model has been used extensively in this respect because it classifies 9459 benign samples and 6035 attack instances correctly, with only 39 benign samples and 43 attack instances misclassified. This speaks well to the model's capability to effectively discriminate between benign and malicious traffic with a high degree of precision and recall. A reduced misclassification rate across both types of samples, both benign and attack, underlines the generalization strength of the Enhanced LSTM, so that it remains effective under different conditions.

The base model of LSTM retains only the same amount of precision and recall. It misclassified 895 benign samples as attacks and 395 attack instances as benign samples, out of a total. Although still competitive, these are considered misclassifications that lower the overall precision and recall of the model, indicating further room for improvement in detection capabilities. The GRU model, which showed a similar trend, could incorrectly classify 1036 benign and 313 attack samples. This represents a small degradation in performance compared with Enhanced-LSTM, especially in the classification of benign traffic.

The RNN model was further limited, with 1848 benign and 421 misclassified attack samples. These results further suggest that RNNs are not good at handling the complex temporal dependencies that characterize IoT network traffic, and thus are unsuitable for the task. Similarly, the TCN model performed well but misclassified 579 benign and 156 attack samples. Although this performance is competitive overall, the generally low precision of the TCN in classifying benign traffic suggests that this network may not learn the features of this dataset as intricately as the Enhanced-LSTM.

Although simple, the FNN model exhibited reasonable performance but did not match the performance of the temporal models. It misclassifies 708 benign and 167 attack samples; hence, it has limitations in terms of handling temporal dependencies. Among all the models presented in this paper, the performance of the Bidirectional LSTM is the most balanced; however, it still falls below that of the Enhanced LSTM. There were 858 benign samples and 334 misclassified samples. Bidirectional LSTM captures both the forward and backward temporal dependencies. However, the accuracy of the final results were not as high as that of the proposed model.

4.3 ROC Curves Analysis

The ROC curves presented in Fig. 5 illustrate the models' ability to plot the true-positive rate versus the false-positive rate on the ROC curve, providing a comprehensive overview of each model's performance for the various categorization thresholds. The AUC determines the overall performance of the model.

The enhanced LSTM model has an almost perfect AUC of 0.9999, reflecting its superior performance in distinguishing between benign and attack traffic independent of the chosen threshold. This further proves that the enhanced LSTM constantly supports high classification performance, which also reiterates its robustness over the classification of IoT traffic data. The AUC score determines the position of the best classifier in this study.

The model closest to this was the TCN model, with an AUC score of 0.9899 indicating a strong overall performance of the model. Although it is slightly lower than that of the enhanced LSTM, it still characterizes the outstanding capability of the TCN in detecting DDoS attacks. This high AUC score indicates that the TCN is a competitive alternative model, particularly when computational efficiency needs to be pursued. Even the FNN model, although a feedforward architecture, has performed quite well with an AUC score of 0.9867; hence, it performs quite well in handling the classification task reasonably well. It simply does not perform as effectively as temporal models.

The ability of the Bidirectional LSTM model to incorporate temporal dependencies in both the forward and backward directions helped it present an AUC score of 0.9717. Although its bidirectional structure aids in its performance, the confusion matrix study shows that it still lags behind the enhanced LSTM because of its comparatively greater misclassification rate. The traditional LSTM model functioned satisfactorily with an AUC score of 0.9671. Nonetheless, the results indicate that the suggested modifications to the LSTM architecture are highly desirable.

With an RNN of 0.9404 and an AUC of 0.9633, they were ranked as GRU. These findings show that although the GRU's performance improves over RNN's, it is still unable to match the enhanced LSTM or TCN's excellent classification performance. The significantly lower AUC values for GRU and RNN imply that these two models are less capable of handling the complicated time and sequence patterns present in IoT traffic data.

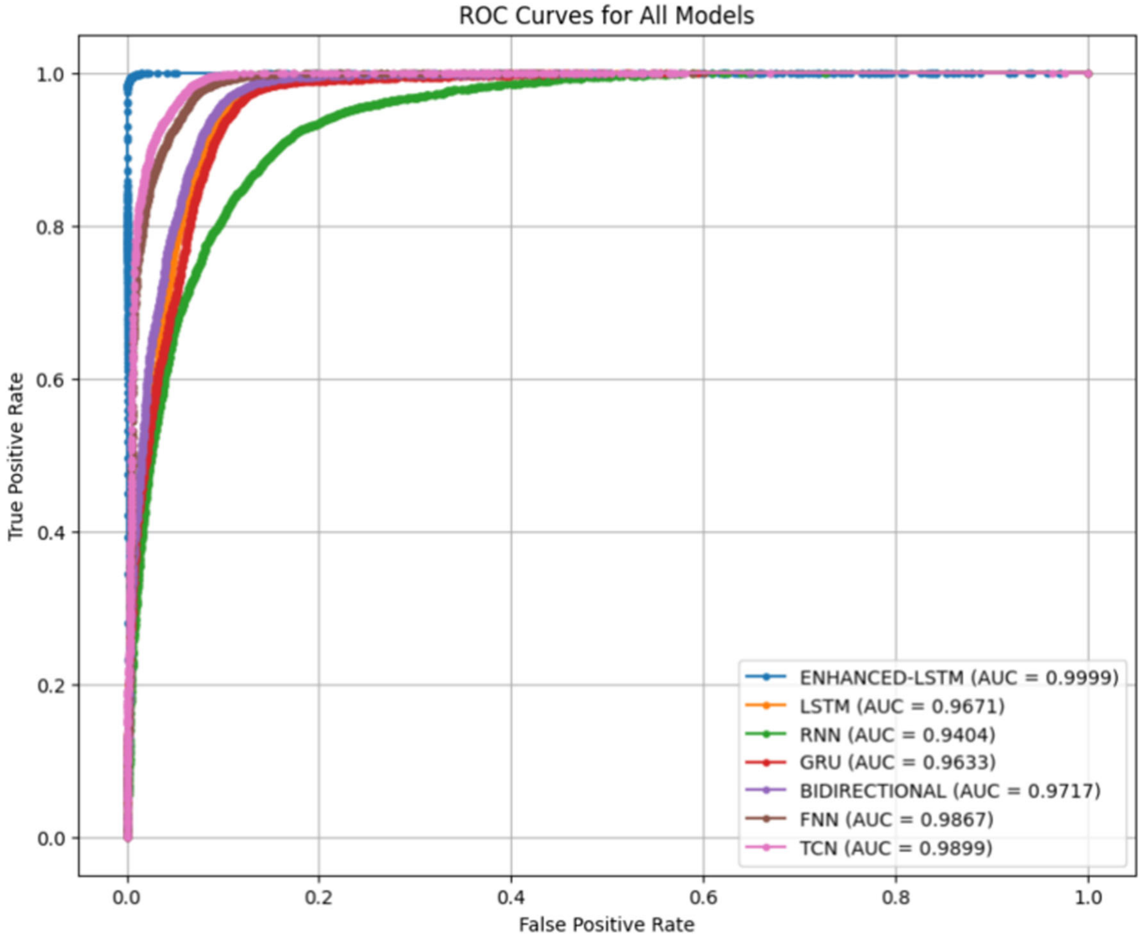

Fig. 5 ROC Curves for all the models

4.4 Features Correlation

Correlations between different network features were visualized using the correlation heatmap shown in Fig. 6. The correlation coefficients range from '–1' to '1', with a pair of features having a '–1' coefficient indicating a perfect negative correlation, and a pair of features having a coefficient of '1' -indicating a perfect positive correlation. Each cell in the heat map corresponds to a specific correlation between two features. The diagonal line, which has a value of '1', reflects the perfect self-correlation that should exist between a given feature and itself. Positive correlations are red, meaning that a direct relationship exists within the features; negative correlations are blue, showing an inverse relationship. For instance, it can be observed that the features src and dt are poorly correlated because the value of the coefficient is close to zero and hence no relationship. In contrast, the features $pktperflow$ and $byteperflow$ have a strong correlation close to 1, which indicates that these two variables are highly positively related.

4.5 Performance Analysis

Table 5 summarizes the performance metrics and, provides a thorough analysis of the categorization abilities of the models. The Enhanced-LSTM model achieved the best performance, with the highest accuracy (99.41%), precision (99.43%), recall (99.41%), and $F1$-score (99.48%). This demonstrates the the effectiveness of the pro-

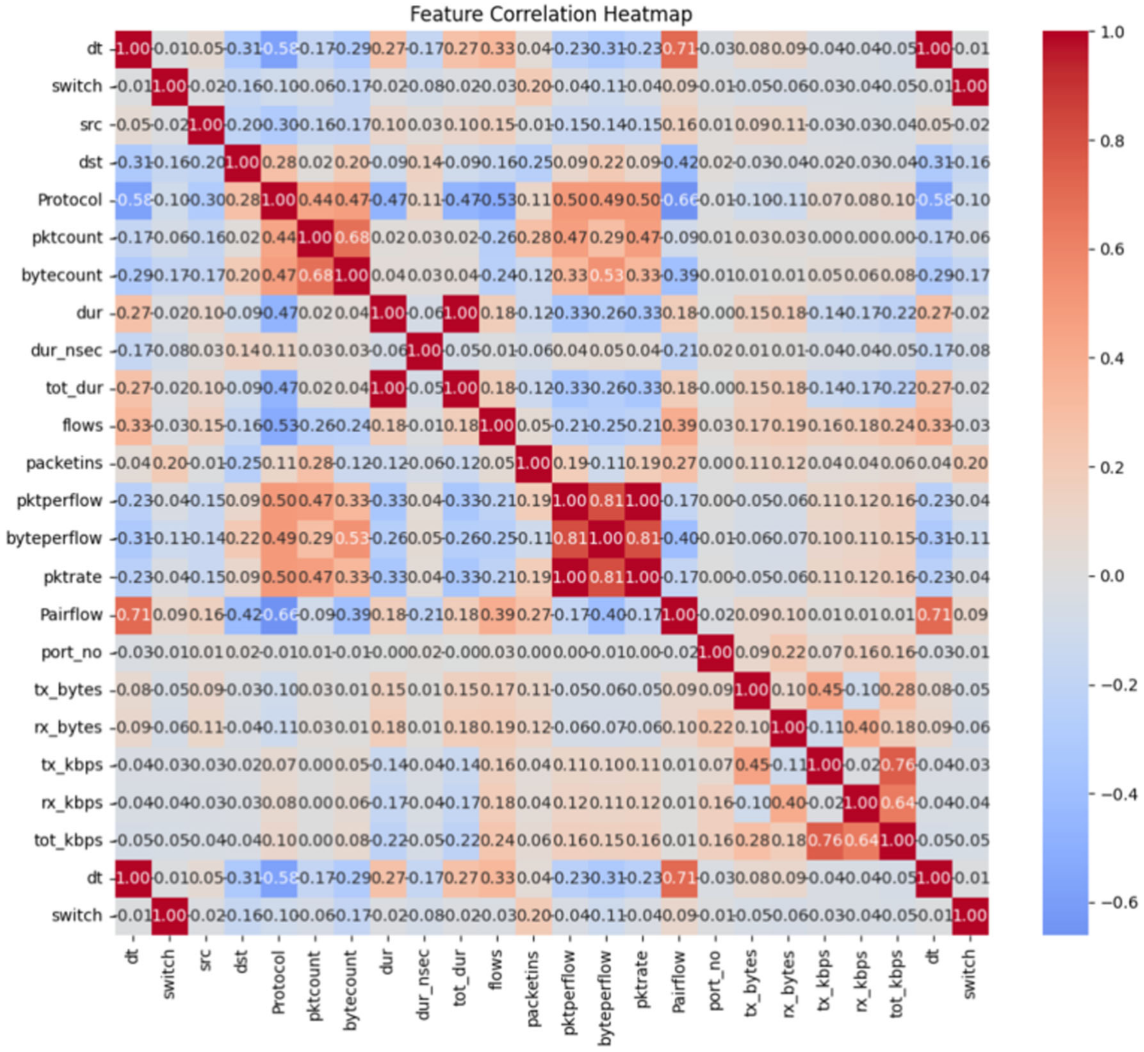

Fig. 6 Features Correlation Heatmap

Table 5 Performance metrics of Models

Model	Accuracy	Precision	Recall	$F1$-score
Enhanced-LSTM	0.994135	0.994335	0.994135	0.994831
TCN	0.952812	0.954764	0.952812	0.953068
FNN	0.943824	0.946891	0.943824	0.944197
Bidirectional	0.923472	0.926592	0.923472	0.923968
LSTM	0.917180	0.920139	0.917180	0.917697
GRU	0.913392	0.918808	0.913392	0.914110
RNN	0.854327	0.872107	0.854327	0.856063

posed approach in identifying DDoS attacks with low misclassification rates, thereby highlighting its resilience to complex IoT network traffic.

While accuracy, precision, recall, and $F1$-score are all marginally lower than those of Enhanced-LSTM, the TCN model still outperformed Enhanced-LSTM in all these metrics. In addition, the Bidirectional LSTM, FNN, and default LSTM models performed effectively; however, their accuracy and recall were significantly lower. The RNN model struggled to capture the entire range of temporal dependencies in the dataset despite performing better than the GRU model. This is notably true in recall, where the RNN's classification scores were considerably lower than those of the Enhanced-LSTM.

In all performance metrics, the RNN model exhibited the lowest performance, indicating its failure to effectively classify instances in IoT traffic data as benign or attacked. This underscores the advantage of using enhanced architectures, such as an Enhanced-LSTM, for optimal DDoS attack detection.

The Enhanced-LSTM model exhibited the most robust performance across all the evaluation metrics, making it highly suitable for intrusion detection in IoT environments. It introduces different enhancements: a combination of Conv1D, Bidirectional LSTM, and GRU layers with dropout regularization, which improves its ability to capture temporal patterns in the data, leading to superior classification results.

Table 6 provides a more detailed comparison between the proposed Enhanced LSTM and the other state-of-the-art DDoS detection methods. The table presents the strengths, weaknesses, and accuracies of each approach for different datasets. Compared with existing works, the proposed model shows promising performance with 99.41% accuracy, proving the feature extraction efficiency, including a low false-positive rate. It retains an enhanced generalization capability while preserving its effectiveness in detecting both traditional and new DDoS attacks in the IoT environment. This comparison further validates the efficiency of the proposed model in addressing the challenges posed by IDS.

5 Conclusions and Future Works

5.1 Discussions

This study addresses the critical problem of detecting Distributed Denial of Service (DDoS) attacks in the Internet of Things (IoT) architecture by proposing an enhanced Intrusion Detection System (IDS) based on Long Short-Term Memory (LSTM) architecture. The proposed model uses convolutional layers followed by Bidirectional LSTM and Bidirectional GRU layers with dropout layers added for regularization. These enhancements improve the ability of the model to extract temporal and spatial dependencies from IoT network traffic. Experimental analysis of the IoT-DH dataset showed superior results, with an accuracy of 99.41% and an AUC of 0.9999. The proposed architecture surpassed conventional deep-learning architectures such as LSTM, GRU, RNN, TCN, FNN, and Bidirectional LSTM in terms of all key performance indicators. These results demonstrate the efficiency, robustness, and suitability of the model for detecting malicious activities in IoT environments with high precision and few misclassifications. These findings further validated the efficacy of combining honeypot-generated data with advanced DL techniques for intrusion detection.

5.2 Limitations

Although the proposed Enhanced LSTM-based IDS provideds promising results, considering a few limitations is of prime importance. The ecosystem of IoT devices is vast and highly heterogeneous, owing to the wide variety of devices, protocols, and architectures. Model evaluation using the IoT-DH dataset alone may not fully represent the diversified attack surfaces across different IoT environments. This raises concerns about the generalization capability of this model to new and diverse attack patterns because IoT systems are continuously evolving and exposing themselves to novel threats. In addition, although honeypot-generated data bring in many real-world experiences, it may not capture all possible attack vectors, especially those just emerging or in sophisticated DDoS methodologies, and the model's evaluation does not explicitly address the evolving characteristics of recent DDoS attacks in IoT networks. This highlights the need for future studies to incorporate insights from into emerging threat patterns. Finally, the computational complexity of the model can limit its deployment in resource-constrained IoT devices, which requires further optimization to ensure scalability and real-time efficiency in practical settings. Because of resource constraints, we performed binary classification; however, future studies should consider

Table 6 Comparison of the proposed model with existing DDoS Detection Techniques

References	Models used	Dataset	Accuracy	Strengths	Weaknesses
Proposed Model	Enhanced LSTM (Conv1D + Bi-LSTM + BiGRU)	IoT-DH Dataset	99.41%	Robust feature extraction, Low false positives, High generalization	High computational cost
[28]	LSTM, Bi-LSTM, Snake Optimizer, Adadelta Optimizer	BoT-IoT Dataset	99.81% (Test)	High accuracy, Robust optimization	High resource usage
[3]	DNN, LSTM	CICDDoS2019	99.97%	High DDoS detection accuracy	Computationally intensive
[15]	Seq2Seq, OpenGAP	NeuPot, SWaT	99.53%	Resilient to unknown threats, Effective simulation	High complexity
[24]	VGG16, ResNet50, Ensemble Learning	MalImg, Real-World Dataset	99.36%	Robust against polymorphic malware	High model complexity
[30]	Bi-LSTM	UNSW-NB15	96.7%	Considers forward and backward dependencies	Computationally intensive
[29]	BiRNN-LSTM	Honeypot Data (2015-2022)	<5% Error	Captures long-term dependencies	Requires frequent retraining
[1]	CNN-LSTM	N-BaIoT	90.88%	Effective on diverse IoT devices	Lower accuracy on specific devices
[38]	LSTM, DLNN	NSL-KDD	99.53% (DLNN)	Hybrid optimization, High accuracy	Hardware-intensive
[26]	LSTM	CIC-IoT2023	98.75%	Effective in IoT traffic scenarios	Requires further validation
[11]	LSTM, GRU, RNN	NSL-KDD, UNSW-NB15	88.13% (NSL-KDD)	Reliable feature selection	Vulnerable to new attack vectors
[41]	Hybrid CNN-LSTM, DT, KNN, LB, NB, PART, RF	Testbed dataset with SCADA system	Hybrid Model: 98%, DT: 99%	High accuracy, Robust feature extraction	Computationally intensive, Requires robust hardware

performing multiclass classification to enhance the robustness and applicability of the model across various attack types.

5.3 Future Directions

This study demonstrates that future research should well serve to focus on several important paths towards enhancing the effectiveness and usability of the proposed Intrusion Detection System (IDS). One promising research focus is the incorporation of heterogeneous, real-time streams of data emanating from diverse Internet of Things (IoT) ecosystems, including industrial IoT, healthcare environments, and smart city infrastructures. Such datasets can represent a wider range of attack vectors and generalize the model. Another promising direction is the application of federated learning frameworks, which enable the distributed and privacy-preserving training of IDS models over multiple IoT environments without the need for centralized data collection. This improves scalability while addressing data-privacy issues. Future work could also explore the integration of more sophisticated anomaly detection mechanisms to enable dynamic adaptation to novel and evolving threats. In addition, optimization methods such as neural network pruning and quantization techniques are two means that can be considered to cope with the computing boundaries of IoT devices. Finally, the network architecture should be transferred to practical use within IoT systems, in which it would be tested with an additional active evaluation of its performance limitations hostile or otherwise encountered to gather insights helpful in robust, large-scale, and flexible intrusion detection systems for protecting IoT networks.

Acknowledgements The authors extend their appreciation to King Saud University for funding this research through Researchers Supporting Project Number (RSPD2025R1027), King Saud University, Riyadh, Saudi Arabia.

Author Contributions All authors contributed to the conceptualization of the study, with A.K.B.A. and M.F.M. taking the lead. A.K.B.A. was responsible for developing the methodology, software, validation, formal analysis, investigation, and data curation. A.K.B.A., M.F.M., M.S., M.A., and M.R.I. drafted the manuscript. A.K.B.A., M.F.M., M.S., M.A., and M.R.I. reviewed and edited the manuscript. Visualization was handled by A.K.B.A., while supervision and project administration were provided by M.F.M., and M.A., who were responsible for funding acquisition. All the authors have read and approved the final version of the manuscript.

Funding This research was funded by the Researchers Supporting Project Number (RSPD2025R1027) of the King Saud University, Riyadh, Saudi Arabia.

Data Availability Statement This dataset is publicly available in the Mendeley database. Introduced by Saif, Syaifuddin, Ferdiana, Ridi, Widyawan, Widyawan (2024), “IoT-DH Dataset,” Mendeley Data, V1, doi: https://doi.org/10.17632/8dns3xbckv.1

Declarations

Conflict of Interest The authors declare that they have no conflicts of interest.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

References

1. Alkahtani, H., Aldhyani, T.H.H.: Botnet attack detection by using cnn-lstm model for internet of things applications. Secur. Commun. Netw. **2021**, 1–23 (2021). https://doi.org/10.1155/2021/3806459
2. Affinito, A., Zinno, S., Stanco, G., Botta, A., Ventre, G.: The evolution of mirai botnet scans over a six-year period. J. Inform. Secur. Appl. **79**, 103629 (2023). https://doi.org/10.1016/j.jisa.2023.103629
3. Khempetch, T., Wuttidittachotti, P.: Ddos attack detection using deep learning. IAES Int. J. Artif. Intell. (IJ-AI) **10**, 382–388 (2021). https://doi.org/10.11591/ijai.v10.i2.pp382-388
4. Dadkhah, S., Mahdikhani, H., Danso, P.K., Zohourian, A., Truong, K.A., Ghorbani, A.A.: Towards the development of a realistic multidimensional iot profiling dataset. In: 2022 19th Annual International Conference on Privacy, Security & Trust (PST), pp. 1–11. IEEE, (2022). https://doi.org/10.1109/PST55820.2022.9851966 . https://ieeexplore.ieee.org/document/9851966/
5. Aslam, M., Ye, D., Tariq, A., Asad, M., Hanif, M., Ndzi, D., Chelloug, S.A., Elaziz, M.A., Al-Qaness, M.A.A., Jilani, S.F.: Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for sdn-enabled iot. Sensors **22**, 2697 (2022). https://doi.org/10.3390/S22072697
6. Kumari, P., Jain, A.K.: A comprehensive study of ddos attacks over iot network and their countermeasures. Comput. Secur. **127**, 103096 (2023). https://doi.org/10.1016/j.cose.2023.103096
7. Barbosa, R., Ogobuchi, O.D., Joy, O.O., Saadi, M., Rosa, R.L., Otaibi, S.A., Rodríguez, D.Z.: Iot based real-time traffic monitoring system using images sensors by sparse deep learning algorithm. Comput. Commun. **210**, 321–330 (2023). https://doi.org/10.1016/j.comcom.2023.08.007
8. Dimolianis, M., Pavlidis, A., Maglaris, V.: Signature-based traffic classification and mitigation for ddos attacks using programmable network data planes. IEEE Access **9**, 113061–113076 (2021). https://doi.org/10.1109/ACCESS.2021.3104115
9. Heidari, A., Jamali, M.A.J.: Internet of things intrusion detection systems: a comprehensive review and future directions. Clust. Comput. **26**, 3753–3780 (2023). https://doi.org/10.1007/s10586-022-03776-z
10. Awajan, A.: A novel deep learning-based intrusion detection system for iot networks. Computers **12**, 34 (2023). https://doi.org/10.3390/computers12020034
11. Kasongo, S.M.: A deep learning technique for intrusion detection system using a recurrent neural networks based framework. Comput. Commun. **199**, 113–125 (2023). https://doi.org/10.1016/j.comcom.2022.12.010
12. Ge, M., Syed, N.F., Fu, X., Baig, Z., Robles-Kelly, A.: Towards a deep learning-driven intrusion detection approach for internet of things. Comput. Netw. **186**, 107784 (2021). https://doi.org/10.1016/j.comnet.2020.107784
13. Sarkar, G., Singh, H., Kumar, S., Shukla, S.K.: Tactics, techniques and procedures of cybercrime: A methodology and tool for cybercrime investigation process. In: ACM International Conference Proceeding Series (2023). https://doi.org/10.1145/3600160.3605013
14. Raghul, S.A., Gayathri, G., Bhatt, R., Kumar, K.A.V.: Enhancing cybersecurity resilience: Integrating ids with advanced honeypot environments for proactive threat detection. Proceedings of the 3rd International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2024, 1363–1368 (2024) https://doi.org/10.1109/ICAAIC60222.2024.10575865
15. Shan, Y., Yao, Y., Zhao, T., Yang, W.: Neupot: a neural network-based honeypot for detecting cyber threats in industrial control systems. IEEE Trans. Industr. Inf. **19**, 10512–10522 (2023). https://doi.org/10.1109/TII.2023.3240739
16. Duy, P.T., Khoa, N.H., Hien, D.T.T., Hoang, H.D., Pham, V.-H.: Investigating on the robustness of flow-based intrusion detection system against adversarial samples using generative adversarial networks. J. Inform. Secur. Appl. **74**, 103472 (2023). https://doi.org/10.1016/j.jisa.2023.103472
17. Agrawal, G., Kaur, A., Myneni, S.: A review of generative models in generating synthetic attack data for cybersecurity. Electronics **13**, 322 (2024). https://doi.org/10.3390/electronics13020322
18. Mittal, M., Kumar, K., Behal, S.: Deep learning approaches for detecting ddos attacks: a systematic review. Soft. Comput. **27**, 13039–13075 (2023). https://doi.org/10.1007/s00500-021-06608-1
19. Dürauer, A., Jungbauer, A., Scharl, T.: Sensors and chemometrics in downstream processing. Biotechnol. Bioeng. **121**, 2347–2364 (2024). https://doi.org/10.1002/BIT.28499
20. Javadpour, A., Ja'fari, F., Taleb, T., Shojafar, M., Benzaïd, C.: A comprehensive survey on cyber deception techniques to improve honeypot performance. Comput. Secur. **140**, 103792 (2024). https://doi.org/10.1016/j.cose.2024.103792
21. Saif, S., Widyawan, W., Ferdiana, R.: Iot-dh dataset for classification, identification, and detection ddos attack in iot. Data in Brief (2024). https://doi.org/10.1016/J.DIB.2024.110496
22. Adedeji, K.B., Abu-Mahfouz, A.M., Kurien, A.M.: Ddos attack and detection methods in internet-enabled networks: concept, research perspectives, and challenges. J. Sens. Actuator Netw. **12**, 51 (2023). https://doi.org/10.3390/JSAN12040051
23. Liu, H., Han, F., Zhang, Y.: Malicious traffic detection for cloud-edge-end networks: a deep learning approach. Comput. Commun. **215**, 150–156 (2024). https://doi.org/10.1016/j.comcom.2023.12.024

24. Kumar, S., Janet, B., Neelakantan, S.: Imcnn: intelligent malware classification using deep convolution neural networks as transfer learning and ensemble learning in honeypot enabled organizational network. Comput. Commun. **216**, 16–33 (2024). https://doi.org/10.1016/j.comcom.2023.12.036
25. Assadhan, B., Bashaiwth, A., Binsalleeh, H.: Enhancing explanation of lstm-based ddos attack classification using shap with pattern dependency. IEEE Access **12**, 90707–90725 (2024). https://doi.org/10.1109/ACCESS.2024.3421299
26. Jony, A.I., Arnob, A.K.B.: A long short-term memory based approach for detecting cyber attacks in iot using cic-iot2023 dataset. J. Edge Comput. **3**, 28–42 (2024). https://doi.org/10.55056/jec.648
27. Singh, N.J., Hoque, N., Singh, K.R., Bhattacharyya, D.K.: Botnet-based iot network traffic analysis using deep learning. Secur. Privacy **7**, 355 (2024). https://doi.org/10.1002/SPY2.355
28. Aljebreen, M., Mengash, H.A., Arasi, M.A., Aljameel, S.S., Salama, A.S., Hamza, M.A.: Enhancing ddos attack detection using snake optimizer with ensemble learning on internet of things environment. IEEE Access **11**, 104745–104753 (2023). https://doi.org/10.1109/ACCESS.2023.3318316
29. Tomar, P.K., Kumar, K.S., Krishna, G., Soumya, K., Ibrahim, R.K., Alazzam, M.B.: Improved detection of cyber-attacks using a bi-directional rnn with lstm deep learning model. In: 2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2023, 2660–2664 (2023) https://doi.org/10.1109/ICACITE57410.2023.10182492
30. Sehrawat, S., Singh, D.: Deep learning approach to enhance ids accuracy using hybrid lstm algorithm. In: 2023 14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023 (2023) https://doi.org/10.1109/ICCCNT56998.2023.10307354
31. Khalaf, L.I., Alhamadani, B., Ismael, O.A., Radhi, A.A., Ahmed, S.R., Algburi, S.: Deep learning-based anomaly detection in network traffic for cyber threat identification. In: ACM International Conference Proceeding Series, 303–309 (2024) https://doi.org/10.1145/3660853.3660932
32. Khanday, S.A., Fatima, H., Rakesh, N.: Implementation of intrusion detection model for ddos attacks in lightweight iot networks. Expert Syst. Appl. **215**, 119330 (2023). https://doi.org/10.1016/j.eswa.2022.119330
33. Nguyen, X.-H., Le, K.-H.: Robust detection of unknown dos/ddos attacks in iot networks using a hybrid learning model. Internet of Things **23**, 100851 (2023). https://doi.org/10.1016/j.iot.2023.100851
34. Alsoufi, M.A., Razak, S., Siraj, M.M., Nafea, I., Ghaleb, F.A., Saeed, F., Nasser, M.: Anomaly-based intrusion detection systems in iot using deep learning: a systematic literature review. Appl. Sci. **11**, 8383 (2021). https://doi.org/10.3390/APP11188383
35. Zhou, L., Zhu, Y., Zong, T., Xiang, Y.: A feature selection-based method for ddos attack flow classification. Futur. Gener. Comput. Syst. **132**, 67–79 (2022). https://doi.org/10.1016/J.FUTURE.2022.02.006
36. Booij, T.M., Chiscop, I., Meeuwissen, E., Moustafa, N., Hartog, F.T.H.: Ton_iot: the role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets. IEEE Internet Things J. **9**, 485–496 (2022). https://doi.org/10.1109/JIOT.2021.3085194
37. Hussan, M.I.T., Reddy, G.V., Anitha, P.T., Kanagaraj, A., Naresh, P.: Ddos attack detection in iot environment using optimized Elman recurrent neural networks based on chaotic bacterial colony optimization. Clust. Comput. **27**, 4469–4490 (2024). https://doi.org/10.1007/s10586-023-04187-4
38. Sumathi, S., Rajesh, R., Lim, S.: Recurrent and deep learning neural network models for ddos attack detection. J. Sens. **2022**, 1–21 (2022). https://doi.org/10.1155/2022/8530312
39. Khraisat, A., Alazab, A.: A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity **4**, 18 (2021). https://doi.org/10.1186/s42400-021-00077-7
40. Priya, V.S.D., Chakkaravarthy, S.S.: Containerized cloud-based honeypot deception for tracking attackers. Sci. Rep. **13**, 1–14 (2023). https://doi.org/10.1038/s41598-023-28613-0
41. Batchu, R.K., Bikku, T., Thota, S., Seetha, H., Ayoade, A.A.: A novel optimization-driven deep learning framework for the detection of ddos attacks. Sci. Rep. **14**, 28024 (2024). https://doi.org/10.1038/s41598-024-77554-9

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Arjun Kumar Bose Arnob[1] · **M. F. Mridha**[1] · **Mejdl Safran**[2] · **Md Amiruzzaman**[3] · **Md. Rajibul Islam**[4]

✉ M. F. Mridha
firoz.mridha@aiub.edu

✉ Mejdl Safran
mejdl@ksu.edu.sa

Arjun Kumar Bose Arnob
arjunkumarbosu@gmail.com

Md Amiruzzaman
mamiruzzaman@wcupa.edu

Md. Rajibul Islam
mdrajibul.islam@polyu.edu.hk

[1] Department of Computer Science, American International University-Bangladesh, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

[2] Department of Computer Science, College of Computer and Information Sciences, King Saud University, 11543 Riyadh, Saudi Arabia

[3] Department of Computer Science, West Chester University, 700 S High St, West Chester 19383, USA

[4] Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon 999077, China