# A New Deep Learning Paradigm for IoT Security: Expanding Beyond Traditional DDoS Detection

Saima Siraj Qureshi[1], Jingsha He[1], Nafei Zhu[1], Min Jia[2], Sirajuddin Qureshi[1], Faheem Ullah[1], Ahsan Nazir[1], and Ahsan Wajahat1[1]

*(Corresponding author: Nafei Zhu)*

Faculty of Information Technology, Beijing University of Technology[1]
Beijing 100124, China
School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150080, China[2]
Email: znf@bjut.edu.cn

*(Received Oct. 11, 2023; Revised and Accepted Jan. 23, 2024; First Online Apr. 25, 2024)*

The Special Issue on Cybersecurity and Privacy in the Industrial Internet of Things (IIoT)

Guest Editor: Prof. Zhengyi Chai (Tiangong University, China)

## Abstract

With the proliferation of Things connected to the Internet (IoT), network vulnerabilities to Attacks known as distributed denial of service (DDoS) have escalated. Conventional DDoS detection methods often falter in the multifaceted IoT landscape. Addressing this, our research introduces a novel hybrid deep learning model, termed CNN-LSTM-GRU, which synergistically integrates (CNN) Convolutional Neural Networks,(LSTM) Long Short-Term Memory, and (GRU) Gated Recurrent Units. Early findings indicate a marked enhancement in detection precision and a reduction in false alarms when juxtaposed with existing methodologies. This paper champions a cutting-edge, versatile deep learning strategy utilizing the CNN-LSTM-GRU fusion to adeptly discern varied network threats. Our methodology harnesses feature clusters from UNSW-NB15 and BOT-IoT Flow datasets, encompassing protocols like DNS, FTP, HTTP, MQTT, and TCP. Based on metrics like accuracy, recall, precision, and F1-score, performance evaluation reveals that our hybrid deep learning model boasts a 98.45% detection rate against IoT-centric threats. Additionally, a comparative analysis underscores the superiority of our model against other leading detection frameworks.

*Keywords: CNN; DDOS; Deep Learning; GRU; LSTM; Network Attacks*

## 1 Introduction

At the beginning of the digital age, the Internet of Things (IoT) ushered in a new era in terms of how people connect to and make use of technology. The Internet of Things has made it possible for objects to communicate with one another, share information with one another, and collaborate in real time. This level of connection was previously inconceivable. Today, the effect of the Internet of Things can be found virtually everywhere [16]. DDoS attacks aim to disrupt services by overwhelming target systems with packets beyond their processing capacity. To amplify these attacks, culprits utilize ”zombie” computers, which are essentially devices compromised by malware. As a result of these attacks, legitimate users often find their requests unanswered due to the network congestion caused by the flood of malicious packets. Among the various DDoS attack types, including the SYN, ICMP, and UDP floods, and http flood are the most commonly observed [20]. A flood attack occurs when an attacker uses the User Datagram Protocol (UDP) to send out a large number of packets without authorization. which is a fast data sharing technology [12]. A SYN flood assault is a form of attack that is based on the occupancy of servers by delivering packets with a spoofed IP address to the victim's servers. This sort of attack takes advantage of a vulnerability in the triple handshake the Transmission Control Protocol (TCP) protocol.An Internet Control Message Protocol (ICMP ) flood attack is an attack in which an excessive number of pings are sent to the computer of the target by taking advantage of sending ICMP packets, which are the messaging protocol for regulating network traffic, waiting without for any response. This type of attack is known as a DoS attack [4]. An HTTP flood attack is a kind of cyberattack that propagates fake request headers to the targeted websites via zombie machines, hence causing service disruptions. The server's resources may be exhausted by this kind of attack. Both automated factories and smart cities fall under this category. Something along the lines of this old proverb states, "With great power comes great responsibility." Distributed denial of service attacks have become a common danger due to fraudsters' increased access to a wider audience as a result of the widespread use of Internet of Things devices [27].

A disruptive denial-of-service attack, or DDoS attack, requires an excessive amount of traffic to be directed towards the targeted computer system, network, or online service [25]. The main goal is to deplete the target's resources to the extent that it becomes unusable and real users are denied access. The target's security will be compromised in order to do this. In the complex web of the Internet of Things, a successful distributed denial of service attack can have catastrophic consequences. For instance, picture a world where vital resources like water and electricity supplies, hospital medical equipment, and power systems are all under threat. Such disasters could have a cascading impact that endangers people's lives, destroys economies, and erodes confidence in digital infrastructure [5].

For a very long time, the cybersecurity community relied on conventional DDoS detection methods to protect the integrity of their systems. The mainstays of this field of study have been both signature-based strategies, which rely on previously identified patterns of known assaults, and anomaly-based methods, which look for significant deviations from norms [19]. However, the limitations of these strategies have been brought to light by the diversity and sheer volume of Internet of Things devices, as well as the dynamic nature of DDoS attacks. Never before has there been a moment when a more trustworthy, adaptable, and intelligent detecting system was not urgently needed.

One branch of machine learning called deep learning has shown itself to be highly skilled at finding subtle patterns in large amounts of data. It is the following stage of the procedure. The detection of distributed denial of service assaults in the Internet of Things may be revolutionized by deep learning models, which take their cues from the neural networks found in the human brain [14]. IoT traffic is unique in that it uses many different protocols, has a wide range of data speeds, and exhibits a wide range of device behaviors. Because of this, handling all of these characteristics of the traffic requires a customized solution.

We present a novel deep learning architecture created specifically for the detection of IoT DDoS in light of these difficulties. This model combines the features of GRU, LSTM, and CNN. The three distinct neural network types that this model successfully combines are as follows: Convolutional neural networks, or CNNs, are capable of spotting patterns in Internet of Things (IoT) data and capturing the subtleties of interactions between devices [7]. Their ability to extract spatial properties is widely acknowledged. With their expertise in modeling temporal sequences, the LSTM network can be used to track how traffic patterns change over time. These networks could be able to detect minute irregularities that point to a potential attack.

Last and Thirdly, Gated Recurrent Units (GRUs), renowned for their efficient learning dynamics and assurance of precise and quick identification, help modify the model's performance. These courses are well-known for their effective learning and have been around for a while. The main goal of the CNN-LSTM-GRU model is to combine the best features of these architectures to offer a comprehensive detection solution. As a result, the model will be able to identify the intricate patterns and sequences typical of Internet of Things data. This paper aims to explore this idea in greater detail by elucidating its methodology, experimental setup, results, and long-term consequences for Internet of Things security. With terms like "IoT security," "deep learning," "neural networks," "DDoS detection," and "hybrid model," this introduction sets the reader up for a thorough examination of the novel CNN-LSTM-GRU technique and its potential to fortify the defenses of the internet of things (IoT) against DDoS attacks. The section also uses phrases like "DDoS detection," "IoT security," and "hybrid model."Because of deep learning, this study suggests a hybrid IoT threat analysis technique that is robust, dependable, and efficient. Deep neural networks (CNN-LSTM-GRU) were employed in the proposed hybrid model to detect new cyber threats and attacks.

The primary contributions of the paper are as follows:

- The paper suggests a unique, flexible, and adaptive DL-based inquiry methodology that effectively identifies different threat classes in a conventional network through hybrid (CNN-LSTM-GRU) computations.

- We discovered that 29 features in the Bot-IoT are either measurable or equivalent to the features in the UNSW-NB15 data set after comparing the features in the two data sets with the attributes in the suggested system.

- The suggested method has been evaluated using common performance evaluation metrics, including F1-score, accuracy, recall, and precision.

- A comparison is also made between the present model and other hybrid deep learning-driven classifiers, such as long short-term memory, deep neural networks, and other earlier research. A thorough evaluation of the suggested method using 10-fold cross-validation has been conducted.

The remainder of the article, Section 2, addresses ideas for current literature from previous years.The shortcomings and difficulties with previous research are also listed in this section. Section 3 presents the approach (i.e., datasets, pre-processing, methodologies, and algorithms) for the proposed hybrid architecture. In Section 4, the results and assessment of the proposed method are outlined, together with a synopsis of the performance evaluation standards that were applied. Section 5 contains the paper's conclusion as well as a plan for future study.

## 2 Related Work

Advanced research and countermeasures have become necessary due to the increase of Distributed Denial of Service (DDoS) assaults, which are distinguished by their increasing complexity and regularity. Four general types of DDoS assaults can be distinguished: http flood, ICMP flood, SYN flood, and UDP flood.

Attackers use the User Datagram Protocol (UDP) to quickly send out a large number of packets without the recipient's permission in a UDP flood attack. The SYN flood attack, on the other hand, floods the target's servers with packets containing forged IP addresses by taking advantage of a flaw in the Transmission Control Protocol's (TCP) triple handshake procedure. ICMP packets, which are necessary for controlling network traffic, are used in the ICMP flood attack to bombard the victim's system with ping requests without waiting for a response. Finally, http flood assaults cause disruptions to services by using zombie devices to send erroneous requests to websites that they target, so using up server resources.

An entropy method known as Shannon entropy has been used to identify these DDoS attacks. In order to create the detection model, this approach primarily focuses on particular attributes, such as the source IP address. However, utilizing programs like scapy and hping, attackers have come up with ways to quickly change the original IP address. The validity of employing the diversity of this property as a detection criterion has been called into question due to its flexibility.

Numerous investigations in this field have focused on the source IP address and used the Shannon entropy method to detect DDoS attacks [3,10]. But attackers utilizing scapy and hping can quickly change this address, raising doubts about its effectiveness.13] argued that a crucial component of DDoS detection, the variety of the originating IP address, might not be a reliable measure. In [22] Deep learning intrusion detection methods, such as DNN, CNN, and RNN architectures, have been developed in the context of Agriculture 4.0. These models use binary and multiclass classifications to assess network performances. They used the CIC-DDoS2019 and TON_IoT datasets to train their algorithms. In [6,23] unveiled a thorough DDoS attack detection system for 5G and B5G that combines an effective feature extraction technique with a composite multilayer perceptron. Their suggested framework demonstrated a low loss of 0.011 and an astounding accuracy of 99.66%. In [23] presented an advanced network intrusion detection system (A-NIDS) that uses an LSTM classifier in conjunction with an improved Onevs-One approach NSL-KDD and CIC-IDS2017 datasets were used to evaluate this system's efficacy. [9] presented a brand-new deep learning system that uses a feed-forward neural network model with embedding layers for multi-class classification to detect Internet of Things intrusions. [8] created a hybrid model that combines two deep learning techniques to detect DDoS attacks. The autoencoder part of their model was quite good at extracting features and identifying the most important feature sets. Their model's Multi-layer Perceptron Network segment achieved an F1-score of over 98% while addressing performance overhead for various forms of DDoS attacks. [28] assessed the performance of feature selection methods on modern datasets, providing summaries of different approaches. Following feature extraction, they compared the lengths of feature selection and training on the same dataset. [13, 18] presented a DL model based on LSTM that may identify DDoS assaults in the SDN control layer with a 98.88% accuracy on the ISCX 2012 and IDS CTU-13 Botnet datasets. [11] presented a hybrid CNN-based intrusion detection technique that combines a GRU model with a CNN. The GRU module was selected because it can retrieve important information from previously collected data by using memory cells and capturing long-dependence properties. [2] developed a 96% accurate Bidirectional LSTM-based framework for IoT-botnet packet detection. CNN and RNN were used to analyze network traffic flow with 99.3% accuracy [1]. [17] benefited from website content and metadata by using a deep learning-based LSTM to detect bots with a 98% accuracy rate. [24] used a variety of deep learning (DL) methods, including CNN, RNN, and LSTM algorithms, to identify domain names independently of data context. The suggested method produced a 90% detection accuracy. This thorough analysis study emphasizes how DDoS detection methods are constantly changing and how attempts are being made to increase their effectiveness.

## 3 Methodology

In this section, we introduce the architecture of our proposed hybrid DL model, as illustrated in Figure 1. These models involve a series of crucial steps: first, In order to identify comparable features between the UNSW-NB15 and Bot-IoT datasets, a feature comparison is performed. This is followed by feature selection, data pre-processing, refinement, and finally, the training of the model using a hybrid approach that combines CNN, LSTM, and GRU deep learning techniques.

**Feature Comparison:** Within our system, we compared features from both datasets. From the Bot-IoT dataset, 29 traits were identified by our study. either matched or could be equated to those in the UNSW-NB15 dataset.

**Feature Selection:** For our system, we categorized features from the BOT-IoT and UNSW-NB15 datasets into clusters based on flow, DNS/FTP/HTTP, MQTT, and TCP. A significant number of these features were grouped into the flow and TCP clusters, as detailed in Table 1. This clustering was informed by a thorough analysis of each feature's description as provided by the original authors. Our goal was to retain a minimal set of features that still encompassed both the application and transport layers. The application layer is primarily represented by flow features, while the transport layer is dominated by the TCP protocol. By focusing on these clusters, we optimized the scenarios to retain the maximum packet information, which in turn considerably decreased the time spent computing during the learning stage.

**Data Preprocessing:** Here, we have delve into the various data preprocessing stages:

1) **Data Type Resolution**: Certain features in our model, such as 'saddr', 'daddr', and 'proto', are categorical and need conversion to a format suitable for algorithms. Specifically, 'saddr' and 'daddr' represent source and destination IP addresses, while 'proto' indicates the flow's protocol type. We gave each of these IP addresses a number.

   In the UNSW-NB15 dataset, there are 49 IP addresses, and the Bot-IoT dataset contains 301. When merging the datasets, We used 350 instead of the IP addresses' unique, randomly generated integers. This not only prevents overfitting but also retains the significance of IP addresses in training and validation datasets, especially for features that rely on them. Similarly, the 'proto' feature was converted to an integer type.

2) **Handling of Missing Port Numbers** : In the complete Bot-IoT dataset, packets using the ARP protocol lack source and destination port numbers. This omission is expected. As noted by Koroniotis *et al.* in [20], ARP port numbers (used by 5% of the Bot-IoT dataset) were assigned the value -1. We adopted this approach for our model, assigning this value to the port number in the entire dataset where the ARP protocol appeared.

3) **Z-scale Normalization**: Normalization ensures that data across different features have a similar distribution, allowing the model to assign comparable importance to each feature. If we consider a feature subspace with N rows and M columns, represented as X = RN × M, z-scale normalization can be applied in the following manner:

   - **Hybrid CNN-LSTM-GRU** After processing, the input data is directed to the training phase. Subsequently, we conducted tests using DNN-LSTM, CNN-LSTM, CNN-BiLSTM, and our newly proposed CNN-LSTM-GRU. The promising outcomes from the CNN-LSTM inspired us to design a hybrid model that combines the strengths of CNNs, LSTM, and GRU. This innovative approach yielded superior results. The structure of this hybrid model can be viewed in the training step of Figure 1.

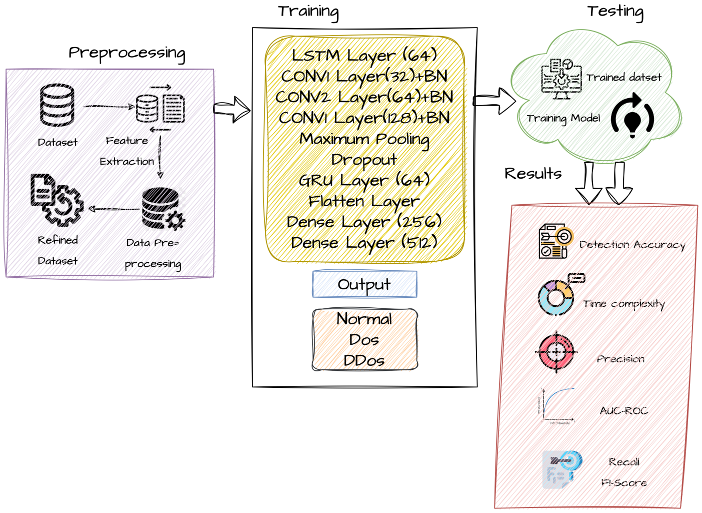


Figure 1: Proposed hybrid CNN-LSTM-GRU architecture

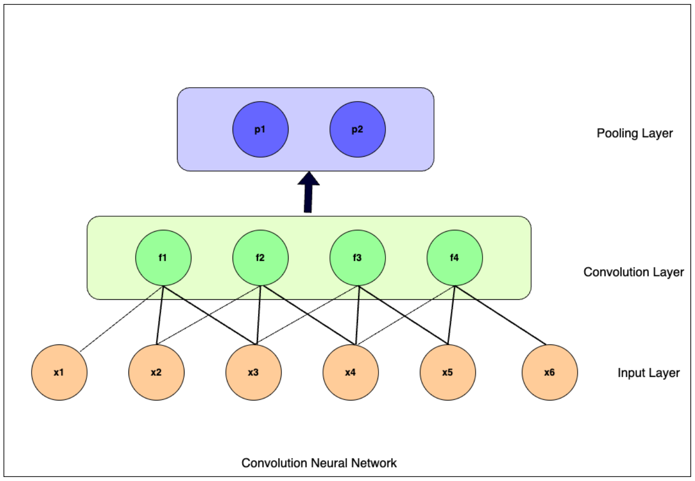


Figure 2: Simple convolution and pooling layer architecture

CNNs are usually used to handle two-dimensional data and are primarily developed for image classification. But time series analysis, which works with one-dimensional data, has also proven useful. The weight-sharing idea is a fundamental component of CNNs and provides improved performance for nonlinear tasks. Figure 2 shows the complex operation of the convolution and pooling layers. Figure 2 shows how input data points such as x1 through x6 are converted into feature maps f1 through f4 by applying convolution. These feature maps are further refined by a pooling layer after the convolutional layer, further abstracting them for usage in conjunction with memory cells and hidden layers.

RNNs, however, are not without their difficulties. The exploding and vanishing gradient problem afflicts them. This issue could lead to the gradient for long-term temporal components becoming exponentially quicker than for short-term ones., especially with expanding gradients. GRU and LSTM are the two most common forms of RNNs. RNNs have backward connections, which can occasionally negatively impact model accuracy, in contrast to CNNs. LSTMs, however, deal with these drawbacks. They are an example of a sophisticated RNN architecture designed with long-range temporal feature dependencies in mind. Looking closer, we can see that the LSTM is made up of cell blocks. These blocks switch between cell and hidden states, and memory blocks use gates to hold onto state information. The three gates of input, forget, and output define an LSTM cell. A GRU, on the other hand, just has two: the update (Z) and reset (Y) gates. The reset gate combines the input sequence of the next cell with the memory of the previous one, while the update gate decides how much of the previous cell's memory is still active. LSTMs are well known for their ability to assess long series and retain knowledge across datasets. According to [26], they outperform a lot of other deep learning algorithms in terms of test completion speed. To gain a deeper understanding of the LSTM cell, consider that it consists of two states (cell and hidden) and three gates (input, forget, and output). Below are the mathematical expressions for these LSTM gates.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f).$$

After deciding to keep the data, the next step is to update the cell's state, which is done by use of an input gate, $i_t$:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i).$$

The function than, which is a hyperbolic tangent, produces a vector of new potential values, $c_t$:

$$c_t = \tanh\left(W_c.[h_{t-1}, x_t] + b_c\right).$$

Multiplying the current candidate value by the previous value and $f_t$ yields the new value under consideration. The equation is further complicated by the addition of $i_t$* $c_t$ is added to the equation.

$$c_t = f_t * c_{t-1} + i_t * c_t$$

The final result is a filtered representation of the cell state, denoted by$o_t$ .

$$\begin{aligned} o_t &= \sigma(W_o * [h_t - 1, x_t] + b_o) \\ h_t &= o_t * \tanh(c_t) \end{aligned}$$

The basic LSTM cell accepts organized data as input, and additionally, the input layer is linked to hidden layers. The size of the output layer is determined by the quantity of classes that must be classified. But LSTM is a little different in a few respects. To start, whereas the GRU cell has two gates, LSTM has three. Second, the input and forget gates in the LSTM are combined to create the update gate, and the reset gate for the hidden state is applied immediately.

A popular paradigm for deep learning algorithms is GRU. GRU is thought to train models 3.6% quicker than other deep learning algorithms, making it the fastest learning model [15]. The cell state is swapped out for a concealed state for data transfer in the modified GRU design. A reset gate and an update gate are the two gates in the GRU model. By managing the data flow through the model with these two gates, the model may refine the output. Information can be retained in a longer sample sequence using a gated recurrent model.The updated gate functions as an input and forget gate for the LSTM. Therefore, the updated gate chooses which data to erase and keep in certain cells. When and what are forgotten are decided by the reset gate. The GRU learns more quickly than the LSTM because it uses fewer tensor operations. The GRU equations that examine the values of two gates and the state of the cell using the GRU algorithm are defined below. Figure 3 displays the general architecture of the RNN, LSTM, and GRU.

$$z_t = \sigma(W_z.[h_{t-1}, x_t]) \qquad (1)$$

Input is multiplied by weight in Equation (1) to determine the update gate at time step t.

$$r_t = \sigma(W_r.[h_{t-1}, x_t]) \qquad (2)$$

Equation (2) depicts the computation at the reset gate, where the input is multiplied by weight by the update gate at time step t.

$$\widetilde{h}_t = \tanh(W.[r_t * h_{t-1}, x_t]) \qquad (3)$$

The current memory is shown in Equation (3) when input is multiplied by weight.

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t \qquad (4)$$

Equation (4) depicts the final memory of the time step in which the update gate is multiplied element by element.

A CNN (Convolutional Neural Network) is a sort of model of deep learning that is designed to perform particularly well when processing structured grid data, such as pictures. CNNs automatically learn hierarchical characteristics from the data that is fed into them, and they do this by utilizing layers such as convolutional, pooling, and fully connected [15, 21]. They begin by identifying simple patterns and then progress to recognizing more complicated structures; as a result, they play an essential role in activities such as the classification of images, the detection of objects, and the identification of faces. CNNs have revolutionized computer vision applications because of their ability to learn spatial characteristics in an adaptable manner and reduce the requirement for feature extraction manuals.

In the proposed framework Figures 1 and 4, employ the LSTM layer for prioritizing sequential modeling over

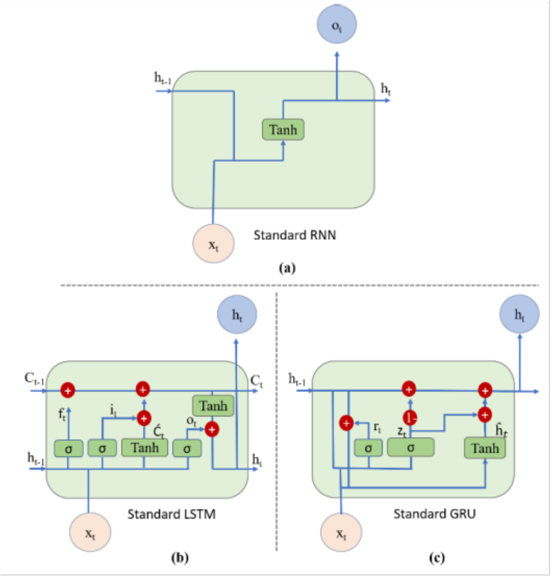


Figure 3: Standard architecture of RNN, LSTM and GRU

spatial feature extraction. A GRU learns a sequence effectively after using CNN features for sequence representation. After the input data has been adjusted, the CNN layers are utilized to extract spatial characteristics, which are then fed into GRU. In this research, we employed three CNN layers with a kernel size of three and a Relu activation function. The first, second, and third layers' filters were 1 × 32, 1 × 64, and 1 × 128 correspondingly. The spatial features are extracted, and then they are fed into GRU layers. Temporal features are modeled by a GRU layer, and IDS prediction is done by a dense layer. The datasets are divided into two parts: 80% and 20%, respectively, for training and testing. Input Layer: Depending on your specific task, the input data can be sequences (e.g., text or time series) or images.

**LSTM Layers:** The input data is directly fed into LSTM layer as the first step. These LSTM layers are responsible for capturing sequential dependencies and temporal patterns in the data [21].

**CNN Layers:** Extract relevant features from the sequences generated by the LSTM layers. Flatten or Global Max Pooling Layer: After the CNN layers, you can flatten the output or use global max pooling to convert the 2D feature maps into a 1D vector.

**GRU Layers:** Optionally, after the CNN layers, add GRU layer to further model sequential information. This can be especially useful if there are complex temporal dependencies that the LSTM layers may not capture adequately.

**Output Layers:** Add appropriate output layers, such as dense layers for sequence tasks.

**Output:** The final output of the model is used for making IDS predictions

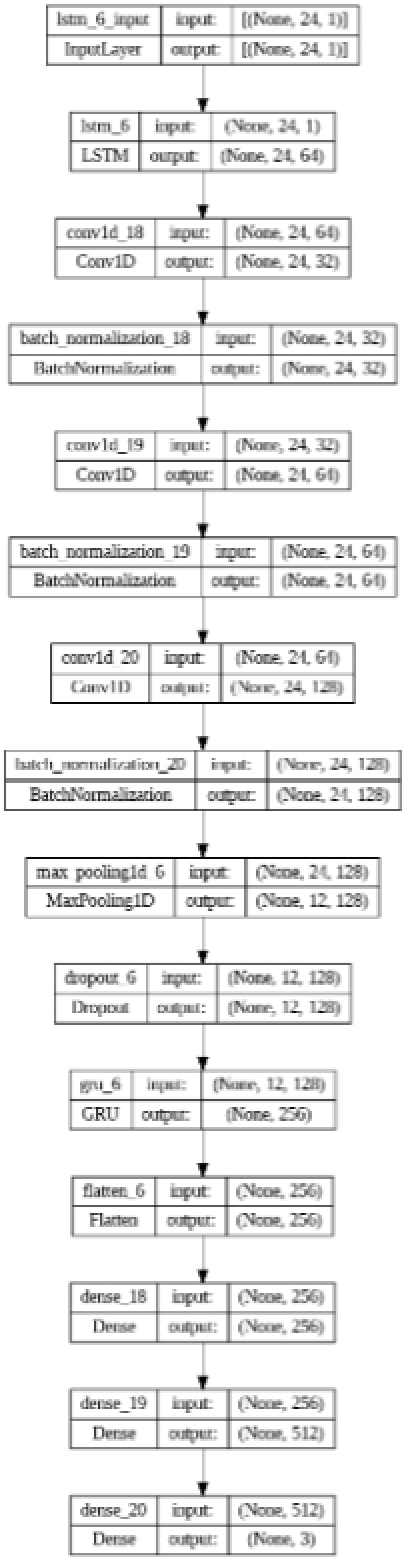

lstm_6_input | input: | [(None, 24, 1)]
InputLayer | output: | [(None, 24, 1)]
lstm_6 | input: | (None, 24, 1)
LSTM | output: | (None, 24, 64)
conv1d_18 | input: | (None, 24, 64)
Conv1D | output: | (None, 24, 32)
batch_normalization_18 | input: | (None, 24, 32)
BatchNormalization | output: | (None, 24, 32)
conv1d_19 | input: | (None, 24, 32)
Conv1D | output: | (None, 24, 64)
batch_normalization_19 | input: | (None, 24, 64)
BatchNormalization | output: | (None, 24, 64)
conv1d_20 | input: | (None, 24, 64)
Conv1D | output: | (None, 24, 128)
batch_normalization_20 | input: | (None, 24, 128)
BatchNormalization | output: | (None, 24, 128)
max_pooling1d_6 | input: | (None, 24, 128)
MaxPooling1D | output: | (None, 12, 128)
dropout_6 | input: | (None, 12, 128)
Dropout | output: | (None, 12, 128)
gru_6 | input: | (None, 12, 128)
GRU | output: | (None, 256)
flatten_6 | input: | (None, 256)
Flatten | output: | (None, 256)
dense_18 | input: | (None, 256)
Dense | output: | (None, 256)
dense_19 | input: | (None, 256)
Dense | output: | (None, 512)
dense_20 | input: | (None, 512)
Dense | output: | (None, 3)

Figure 4: Graphical representation of CNN-LSTM-GRU model

# 4 Result and Discussion

We conducted a thorough evaluation of our proposed intrusion detection method using standard performance metrics such as accuracy, precision, recall, F1-score, and more. These metrics are derived from the confusion matrix using mathematical computations. Additionally, we've illustrated the AU-ROC curves to visually represent the relationship between positive and negative rates. Essential parameters like true positive (TP), false positive (FP), true negative (TN), and false negative (FN) are also extracted from the confusion matrix. Here's a concise overview and mathematical foundation of these performance metrics:

1) **Confusion Matrix**: This 2D matrix showcases the relationship between actual and predicted values. True rates reflect the classifier's overall correct predictions, whereas negative rates highlight incorrect predictions.

2) **Accuracy**: A primary metric, accuracy gauges the classifier's overall performance. It captures the proportion of samples correctly classified, both positives and negatives. Its formula is:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

3) **Precision**: Precision quantifies the proportion of true positive detections to the total positive detections. Its formula is:

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

4) **Recall**: Recall, on the other hand, is the ratio of the true positive rate to the sum of the true positive and false negative rates. Its formula is:

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

5) **F-measure**:Representing the harmonic mean of recall and precision, the F1-score's formula is:

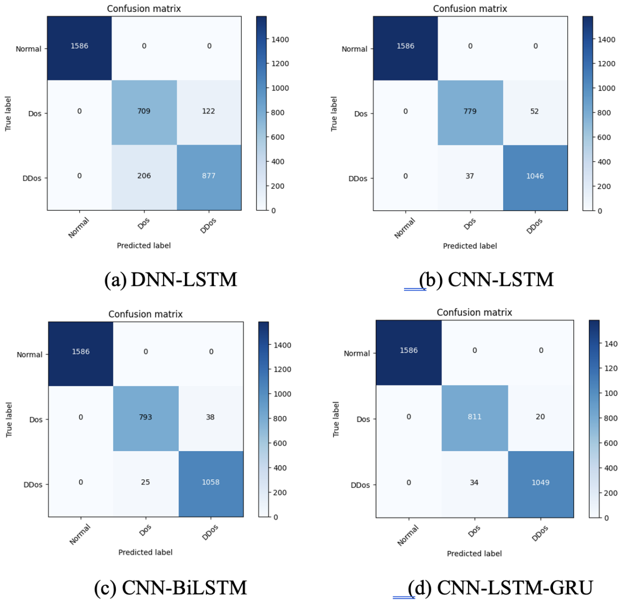
$$F1-score = \frac{2*TP}{2*TP+FP+FN} \tag{8}$$

6) **AU-ROC**: This metric illustrates the classifier's diagnostic capability graphically. The curve plots the true positive rate against the false positive rate across varying thresholds.

$$P(X_1 > X_0\ ) = P(X_1 - X_0 > 0) \tag{9}$$

Its formula involves X1, the random variable denoting the rate for random positive samples, and X0, the continuous random variable representing the rate for randomly chosen negative samples.

$$AU-ROC = \int_0^1 TPR(FPR)d(FPR) \tag{10}$$



(a) DNN-LSTM (b) CNN-LSTM

(c) CNN-BiLSTM (d) CNN-LSTM-GRU

Figure 5: Confusion matrix for multi-classification derived from various deep learning methods

$$AU-ROC = \int_0^1 TPR(FPR_1(x))dx \tag{11}$$

The outcomes of our evaluation are tabulated in Table 1. A glance reveals the superior performance of our CNN-LSTM-GRU model relative to other methods. Specifically, the DNN-LSTM algorithm lags behind other deep learning models, with the standard CNN achieving an accuracy of 90.62%. Remarkably, the fusion of CNN with LSTM surpasses all other algorithms, achieving an impressive accuracy of 98.45%. This underscores the potency of our hybrid CNN-LSTM-GRU model in intrusion detection. Additionally, the hybrid CNN-LSTM model boasts superior precision and recall compared to its counterparts. Yet, when considering the F1-score across three classes, the CNN-LSTM-GRU model emerges as the clear frontrunner.

Furthermore, we present the efficacy of our suggested approach in classifying both regular and malicious data, specifically DOS and DDoS attacks. Figure 1showcases the confusion matrix (CM) derived from the testing phase for various deep-learning strategies. Every instance in this test set is categorized as either regular or malicious activity. Notably, our advanced CNN-LSTM-GRU model demonstrates superior precision in accurately identifying malicious events. Our CNN-LSTM-GRU model performed well in experiments, with F1-scores between 0.9766 and 0.9811. Table 2 shows the model's high precision and recall metrics, which indicate strong predictive dependability and demonstrate its ability to effectively identify and define dataset cases."

Figure 5 shows the confusion matrix for multi-classification derived from various deep learning methods. To delve deeper into the performance of our ad-

Table 1: Comparison of different models

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| DNN-LSTM | 0.9062 | 0.9087 | 0.9062 | 0.9066 |
| CNN-LSTM | 0.9745 | 0.9745 | 0.9745 | 0.9745 |
| CNN-BiLSTM | 0.9820 | 0.9820 | 0.9820 | 0.9819 |
| CNN-LSTM-GRU | 0.9845 | 0.9846 | 0.9845 | 0.9845 |

Table 2: Metrics for Models 1-10

| Metrics | Model | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | DNN-LSTM | 0.7697 | 0.7617 | 0.7742 | 0.7725 | 0.7685 | 0.672 | 0.7684 | 0.7661 | 0.7581 | 0.7026 |
| | CNN-LSTM | 0.9771 | 0.9748 | 0.9662 | 0.9668 | 0.9148 | 0.972 | 0.9817 | 0.9251 | 0.9651 | 0.8982 |
| | CNN-BiLSTM | 0.9657 | 0.9782 | 0.972 | 0.9634 | 0.9742 | 0.9634 | 0.9834 | 0.9748 | 0.9645 | 0.9285 |
| | CNN-LSTM-GRU | 0.9765 | 0.9782 | 0.9714 | 0.9857 | 0.9748 | 0.9851 | 0.9788 | 0.9765 | 0.9799 | 0.9811 |
| Precision | DNN-LSTM | 0.8666 | 0.6958 | 0.8490 | 0.7875 | 0.6348 | 0.6712 | 0.8354 | 0.7312 | 0.8175 | 0.809 |
| | CNN-LSTM | 0.9777 | 0.9749 | 0.9662 | 0.9668 | 0.9354 | 0.9719 | 0.9818 | 0.9330 | 0.9651 | 0.9266 |
| | CNN-BiLSTM | 0.9658 | 0.9785 | 0.9719 | 0.9634 | 0.9759 | 0.9635 | 0.9835 | 0.9760 | 0.9659 | 0.9372 |
| | CNN-LSTM-GRU | 0.9767 | 0.9784 | 0.9714 | 0.9858 | 0.9748 | 0.9856 | 0.9788 | 0.9773 | 0.9803 | 0.9811 |
| Recall | DNN-LSTM | 0.7697 | 0.7617 | 0.7742 | 0.7725 | 0.7685 | 0.672 | 0.7684 | 0.7661 | 0.7581 | 0.7026 |
| | CNN-LSTM | 0.9771 | 0.9748 | 0.9662 | 0.9668 | 0.9148 | 0.972 | 0.9817 | 0.9251 | 0.9651 | 0.8982 |
| | CNN-BiLSTM | 0.9657 | 0.9782 | 0.972 | 0.9634 | 0.9742 | 0.9634 | 0.9834 | 0.9748 | 0.9645 | 0.9285 |
| | CNN-LSTM-GRU | 0.9765 | 0.9782 | 0.9714 | 0.9857 | 0.9748 | 0.9851 | 0.9788 | 0.9765 | 0.9799 | 0.9811 |
| F1-score | DNN-LSTM | 0.6864 | 0.6873 | 0.6974 | 0.7002 | 0.6838 | 0.5751 | 0.7514 | 0.6922 | 0.7409 | 0.6367 |
| | CNN-LSTM | 0.9772 | 0.9748 | 0.9662 | 0.9668 | 0.9149 | 0.9719 | 0.9817 | 0.9255 | 0.9651 | 0.8978 |
| | CNN-BiLSTM | 0.9657 | 0.9783 | 0.9719 | 0.9634 | 0.9743 | 0.9634 | 0.9834 | 0.9749 | 0.9646 | 0.9289 |
| | CNN-LSTM-GRU | 0.9766 | 0.9783 | 0.9714 | 0.9857 | 0.9748 | 0.9851 | 0.9788 | 0.9766 | 0.9800 | 0.9811 |

Table 3: FDR, FNR, FOR and FPR values of DNN-LSTM, CNN-LSTM, CNN-BiLSTM and CNN-LSTM-GRU

| Metrics | DNN-LSTM | CNN-LSTM | CNN-BiLSTM | CNN-LSTM-GRU |
|---|---|---|---|---|
| FDR | 0.1221 | 0.0473 | 0.0346 | 0.0187 |
| FNR | 0.1902 | 0.0341 | 0.0230 | 0.0313 |
| FOR | 0.2251 | 0.0453 | 0.0305 | 0.040 |
| FPR | 0.1468 | 0.0625 | 0.0457 | 0.024 |

vanced CNN-LSTM-GRU model, we employ the receiver operating characteristics (ROC) curve, depicted in Figure 6. This curve elucidates the relationship between true-positive and false-positive rates, with the area under the curve (AUC) serving as an indicator of the model's proficiency. Impressively, our CNN-LSTM model boasts the highest AUC at 0.972. This is closely followed by the CNN-BiLSTM and CNN-LSTM-GRU algorithms, registering AUC values of 0.965 and 0.951, respectively. On the other end of the spectrum, the DNN-LSTM lags behind, recording the lowest AUC at 0.831, suggesting its subpar efficacy in detecting network anomalies.

We have also determined values for FNR, FPR, FDR, and FOR, comparing our proposed methods with existing algorithms, as detailed in Figure 7. Table 3 reveals occasional misclassification of benign class samples. Additionally, our proposed method's TNR, MCC, and NPV metrics are depicted in Figure 8 and Table 4. With optimal values ranging between 90 and 95 for TNR, MCC, and NPV, it underscores the classifier's robust performance, making it apt for deployment in IIoT systems and networks for intrusion detection. Furthermore, Figure 9 illustrates the processing speed of our proposed method in comparison to other contemporary classifiers. Specifically, the CNN-LSTM processed 1000 samples in a mere 300 microseconds during testing. When we extended the experiment to various models, it provided insights into the performance dynamics of different deep learning classifiers. Notably, our CNN-LSTM-GRU algorithm outshines its counterparts in terms of time efficiency. The graph suggests that while the CNN-LSTM-GRU does have a slight trade-off, it remains competitive in testing time compared to recent algorithms.

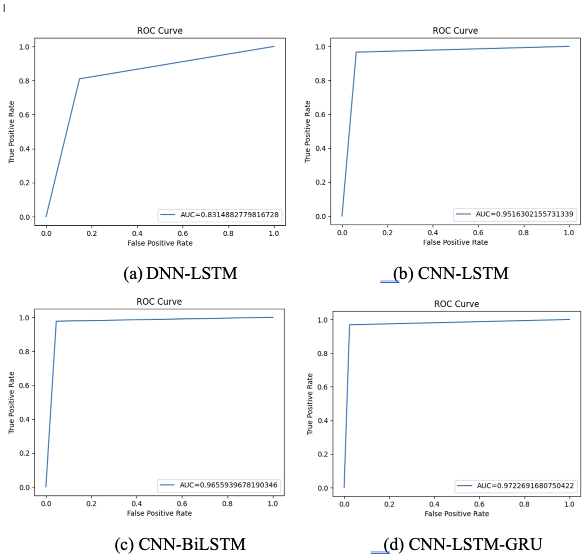
Comparison of Techniques with Existing Techniques For a thorough assessment of our proposed method, we juxtaposed our results with those of benchmarked existing techniques. Table 5 provides a detailed comparative analysis, highlighting how our envisioned approach stacks up against leading-edge IoT intrusion detection systems tailored for industrial IoT.

Table 4: TNR, MCC, and NPV values of DNN-LSTM, CNN-LSTM, CNN-BiLSTM, and CNN-LSTM-GRU

| **Metrics** | **DNN-LSTM** | **CNN-LSTM** | **CNN-BiLSTM** | **CNN-LSTM-GRU** |
|---|---|---|---|---|
| TNR | 0.8531 | 0.9374 | 0.9542 | 0.9759 |
| MCC | 0.6578 | 0.9052 | 0.9329 | 0.9427 |
| NPV | 0.7748 | 0.9546 | 0.9694 | 0.9697 |

Table 5: Comparison of Algorithms

| **Work** | **Algorithms** | **Accuracy** | **Precision** | **Recall** | **F1-score** |
|---|---|---|---|---|---|
| ours | CNN-LSTM-GRU | 98.45% | 98.46% | 98.45% | 98.45% |
| [24] | Cu-ConvLSTM2D | 97.74% | 98.11% | 98.22% | 98.22% |
| [24] | Hybrid(CNN-LSTM) | 97.29% | 97.25% | 97.50% | 97.29% |



(a) DNN-LSTM

(b) CNN-LSTM

(c) CNN-BiLSTM

(d) CNN-LSTM-GRU

Figure 6: (ROC) curve elucidates the relationship between true-positive and false-positive rates

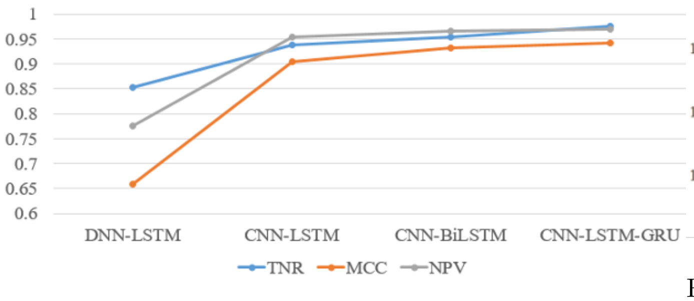


Figure 7: TNR, MCC and NPV values of DNN-LSTM, CNN-LSTM, CNN-BiLSTM and CNN-LSTM-GRU

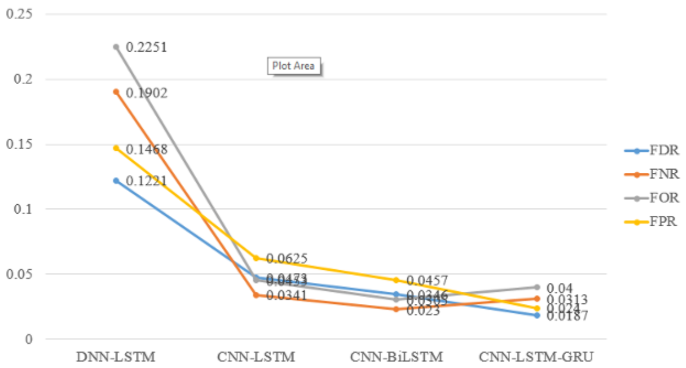


Figure 8: FDR, FNR, FOR and FPR values of DNN-LSTM, CNN-LSTM, CNN-BiLSTM and CNN-LSTMGRU

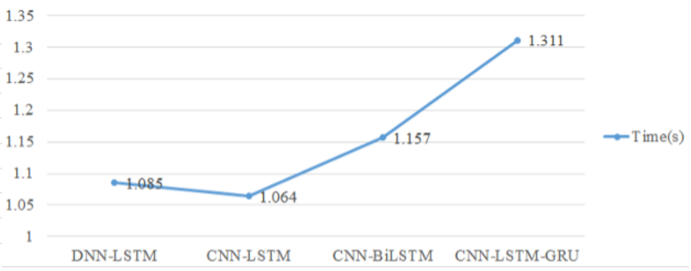


Figure 9: Testing Time of DNN-LSTM, CNN-LSTM, CNN-BiLSTM and CNN-LSTM-GRU

# 5 Conclusion

In this modern era of smart devices, the increased interconnectedness has inadvertently prepared the way for major cybersecurity concerns, particularly Distributed Denial of Service (DDoS) assaults. These kinds of attacks can take down a whole network by overwhelming it with requests for services. We used important measures such as accuracy, recall, precision, and F1-score in order to evaluate the adapted version of our solution that we had provided for this shifting environment. The hybrid deep learning solution that we presented, which integrated the strengths of CNN, LSTM, and GRU, displayed an impressive 98.45% detection rate when put to the test against IoT-centric issues. Beyond its comparative superiority to other leading detection procedures, our methodology highlights the potential of integrated architectures in strengthening threat detection in our interconnected digital world. This marks a pivotal contribution to our research and is one of the most important takeaways from it.

# 6 Funding Information

The work presented in this paper has been supported by Beijing Natural Science Foundation (No. IS23054).

# References

[1] M. M. Alani, "Botstop: Packet-based efficient and explainable iot botnet detection using machine learning," *Computer Communications*, vol. 193, pp. 53–62, 2022.

[2] L. A. Aldossary, M. Ali, and A. Alasaadi, "Securing scada systems against cyber-attacks using artificial intelligence," in *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, pp. 739–745. IEEE, September 2021.

[3] M. A. Amanullah, R. A. A. Habeeb, F. H. Nasaruddin, A. Gani, E. Ahmed, A. S. M. Nainar, N. M. Akim, and M. Imran, "Deep learning and big data technologies for iot security," *Computer Communications*, vol. 151, pp. 495–517, 2020.

[4] J. B. Cabrera, L. Lewis, X. Qin, W. Lee, R. K. Prasanth, B. Ravichandran, and R. K. Mehra, "Proactive detection of distributed denial of service attacks using mib traffic variables-a feasibility study," in *Proceedings of the 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No. 01EX470)*, pp. 609–622. IEEE, May 2001.

[5] D. Curran, "Surveillance capitalism and systemic digital risk: The imperative to collect and connect and the risks of interconnectedness," *Big Data & Society*, vol. 10, no. 1, p. 20539517231177621, 2023.

[6] M. A. Ferrag, L. Shu, H. Djallel, and K. K. R. Choo, "Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0," *Electronics*, vol. 10, no. 11, p. 1257, 2021.

[7] S. Gallacher, D. Wilson, A. Fairbrass, D. Turmukhambetov, M. Firman, S. Kreitmayer, O. Mac Aodha, G. Brostow, and K. Jones, "Shazam for bats: Internet of things for continuous real-time biodiversity monitoring," *IET Smart Cities*, vol. 3, no. 3, pp. 171–183, 2021.

[8] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for internet of things," *Computer Networks*, vol. 186, p. 107784, 2021.

[9] N. Gupta, V. Jindal, and P. Bedi, "Lio-ids: Handling class imbalance using lstm and improved one-vs-one technique in intrusion detection system," *Computer Networks*, vol. 192, p. 108076, 2021.

[10] S. Hosseini and M. Azizi, "The hybrid technique for ddos detection with supervised learning algorithms," *Computer Networks*, vol. 158, pp. 35–45, 2019.

[11] R. H. Hwang, M. C. Peng, C. W. Huang, P. C. Lin, and V. L. Nguyen, "An unsupervised deep learning model for early network traffic anomaly detection," *IEEE Access*, vol. 8, pp. 30387–30399, 2020.

[12] T. Khempetch and P. Wuttidittachotti, "Ddos attack detection using deep learning," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 2, p. 382, 2021.

[13] M. Di Mauro, G. Galatro, G. Fortino, and A. Liotta, "Supervised feature selection techniques in network intrusion detection: A critical review," *Engineering Applications of Artificial Intelligence*, vol. 101, p. 104216, 2021.

[14] N. Mishra and S. Pandya, "Internet of things applications, security challenges, attacks, intrusion detection, and future visions: A systematic review," *IEEE Access*, vol. 9, pp. 59353–59377, 2021.

[15] M. Mudassir, D. Unal, M. Hammoudeh, and F. Azzedin, "Detection of botnet attacks against industrial iot systems by multilayer deep learning approaches," *Wireless Communications*.

[16] A. Nazir, J. He, N. Zhu, A. Wajahat, X. Ma, F. Ullah, S. Qureshi, and M. S. Pathan, "Advancing iot security: A systematic review of machine learning approaches for the detection of iot botnets," *Journal of King Saud University-Computer and Information Sciences*, p. 101820, 2023.

[17] A. Pektaş and T. Acarman, "Deep learning to detect botnet via network flow summaries," *Neural Computing and Applications*, vol. 31, pp. 8021–8033, 2019.

[18] R. Priyadarshini and R. K. Barik, "A deep learning based intelligent framework to mitigate ddos attack in fog environment," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 825–831, 2022.

[19] L. L. Puryear. *A quantitative correlational study of malicious software (malware) identification and security practitioners' training*. PhD thesis, University of Phoenix, 2019.

[20] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Ullah, A. Nazir, and A. Wajahat, "An adaptive multi-layer architecture for iot based idps for attacks using deep learning method," *International Journal of Network Security*, vol. 24, no. 5, pp. 815–827, 2022.

[21] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Akhtar, F. Ullah, A. Nazir, and A. Wajahat, "A hybrid DL-based detection mechanism for cyber threats in secure networks," *IEEE Access*, vol. 9, pp. 73938-73947, 2021.

[22] S. Qureshi, J. He, S. Tunio, N. Zhu, F. Ullah, A. Nazir, and A. Wajahat, "Analysis distributed denial-of-service attack deploy deep learning techniques," *International Journal of Network Security*, vol. 25, no. 5, pp. 745–757, 2023.

[23] S. S. Qureshi, J. He, N. Zhu, Z. A. Zardari, T. Mahmood, and A. Wajahat, "Sdn-enabled deep learning based detection mechanism (ddm) to tackle ddos attacks in iots," *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 6, pp. 10675–10687, 2023.

[24] S. Rao, A. K. Verma, and T. Bhatia, "A review on social spam detection: Challenges, open issues, and future directions," *Expert Systems with Applications*, vol. 186, p. 115742, 2021.

[25] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in iot: A survey," *The Journal of Supercomputing*, vol. 76, pp. 5320–5363, 2020.

[26] S. Salmi and L. Oughdir, "Performance evaluation of deep learning techniques for dos attacks detection in wireless sensor network," *Journal of Big Data*, vol. 10, no. 1, pp. 1–25, 2023.

[27] M. Shafiq, Z. Gu, O. Cheikhrouhou, W. Alhakami, and H. Hamam, "The rise of "internet of things": Review and open research issues related to detection and prevention of iot-based security attacks," *Wireless Communications and Mobile Computing*, pp. 1–12, 2022.

[28] Y. Wei, J. Jang-Jaccard, F. Sabrina, W. Xu, S. Camtepe, and A. Dunmore, "Reconstruction-based lstm-autoencoder for anomaly-based ddos attack detection over multivariate time-series data," *arXiv preprint arXiv:2305.09475*, 2023.

# Biography

**SAIMA SIRAJ QURESHI** received the BSIT(Hons) with gold medal from Sindh Agriculture University Tandojam, Pakistan. Afterwards, she pursued her MSIT from Isra University Hyderabad, Pakistan. Currently, she is pursuing PhD in Information Technology at the Beijing University of Technology, China. She has more than five research publications to her credit as main author and co-author, which featured national and international journals and conferences. Saimas research areas include but not limited to Information security. IoT security, Digital Forensics, Cyber Security, Computer Networks.

**JINGSHA HE** received a bachelor's degree in computer science from Xi'an Jiaotong University, China, and the master's and Ph.D. degrees in computer engineering from the University of Maryland, College Park, MD, USA. He worked for several multinational companies in USA, including IBM Corp., MCI Communications Corp., and Fujitsu Laboratories. He is currently a Professor with the Faculty of Information Technology, Beijing University of Technology(BJUT), Beijing. He has published more than ten articles. He holds 12 U.S. patents. Since August 2003, he has been published over 300 papers in scholarly journals and international conferences. He also holds over 84 patents and 57 software copyrights in China and authored nine books. He was a principal investigator of more than 40 research and development projects. His research interests include information security, wireless networks, and digital forensics.

**NAFEI ZHE** received the B.S. and M.S. degrees from Central South University, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science and technology from the Beijing University of Technology, Beijing, China, in 2012. She was a Postdoctoral Research Fellow with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, from 2015 to 2017. She is currently an Associate Professor at the Faculty of Information Technology, Beijing University of Technology. She has published over 20 research papers in scholarly journals and international conferences. Her research interests include information security and privacy, wireless communications, and network measurement.

**Min Jia (Senior Member, IEEE)** received the M.Sc. degree in information and communication engineering from the Harbin Institute of Technology (HIT), Harbin, China, in 2006, and the Ph.D. degree from Sungkyunkwan University and HIT in 2010.,She is currently a Professor and a Ph.D. Supervisor with the School of Electronics and Information Engineering, HIT. Her research interests include advanced mobile communication technology for LTE and 5G, cognitive radios, digital signal processing, and advanced broadband satellite communication systems. She is a member of the Steering Committee of the WiSATs International Conference. She has won six best paper awards at several international conferences. She is also the Winner of the Science Fund for Excellent Young Scholars for Heilongjiang Province. She is the General Chair of the IEEE GLOBECOM 2019 Workshop Intelligent and Cognitive Space, Terrestrial and Ocean Internet, Systems and Applications.

**SIRAJUDDIN QURESHI** received his bachelor degree in Computer Sciences from Quaid-e-Awam University of Engineering, Science & Technology, Pakistan. Afterwards, he pursued his Master's in Information Technology from Sindh Agricultural University Tandojam,Pakistan. Currently, he is pursuing PhD in Information Technology at the Beijing University of Technology, China. He has nine research publications to his credit as main author and co-author, which featured national and international journals and conferences. Sirajuddin research areas include but are not limited to Network Forensics Analysis, Digital Forensics, Cyber security, Computer Networks and Network Security.

**FAHEEM ULLAH** received M.S degrees from the Xian Jiaotong University, China,in 2017. He is currently pursuing a Ph.D. degree at the Beijing University of Technology, Beijing, China. His research interests include information security, Blockchain, and data mining. He has received awards and honors from the China Scholarship Council (CSC), China.

**AHSAN NAZIR** has received his M.Sc degree from the University of Engineering and Technology Lahore in 2016. From September 2015 to August 2018 he worked as Software Engineer at Dunya Media group Lahore since September 2018 he is doing Ph.D. in Software Engineering from Beijing University of Technology, Beijing China . He has published more than 10 journals and conference papers .His area of research include eGovernment,IoT, Software Engineering and Machine learning applications .

**AHSAN WAJAHAT** received the B.S. and M.S degrees in information technology from the Sindh Agriculture University, Pakistan in 2012 and 2016, respectively. He is currently pursuing a Ph.D. degree at the Beijing University of Technology, Beijing, China. His research interests include machine learning, information security, forensic networks and data mining. He has received awards and honors from the China Scholarship Council (CSC), China.