International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(21s) 2026 pp. 1003-1021
https://doi.org/10.70917/ijcisim-2026-5373
Article

A Hybrid CNN–BiGRU Deep Learning Model for DDoS Attack Detection in Cybersecurity

Swetha V[1*], Divyashree S[2], Deepti N N[3], Roopa R[4]

[1]Assistant Professor, Department of Computer Science and Engineering, East Point College of Engineering and Technology Bangalore-560049 shwethavgiri@gmail.com
[2]Assistant Professor, Department of Computer Science and Engineering, East Point College of Engineering and Technology, Bangalore-560049 drdivya237@gmail.com
[3]Assistant Professor, Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bangalore-560109 ndeepti@gmail.com
[4]Assistant Professor, Department of Computer Science and Engineering, Rajiv Gandhi Institute of Technology, Bangalore-560109 rooparajgowda2002@gmail.com

Abstract: Distributed Denial of Service (DDoS) attacks remain one of the most serious threats to modern network infrastructures, as they overwhelm systems with massive traffic and disrupt legitimate services. Traditional intrusion detection systems often struggle to detect complex and evolving attack patterns due to their reliance on manual feature engineering and limited learning capability. To address this issue, this study proposes a hybrid deep learning model that integrates Convolutional Neural Networks (CNN) and Bidirectional Gated Recurrent Units (BiGRU) for efficient DDoS attack detection. The CNN component is used to automatically extract important spatial features from network traffic data, while the BiGRU layer captures temporal dependencies and sequential patterns in the traffic flows. The proposed model is evaluated using the CICDDoS2019 dataset, which contains realistic benign and attack traffic across multiple DDoS categories. Experimental results demonstrate that the model achieves high performance in both binary and multiclass classification tasks, obtaining an accuracy of 99.82% for binary detection and 99.21% for multiclass classification. The results indicate that the hybrid CNN–BiGRU architecture effectively improves detection accuracy and provides a reliable approach for intelligent network intrusion detection systems.
Keywords: Deep Learning, DDoS Attack Detection, Intrusion Detection System (IDS), Convolutional Neural Network (CNN), Bidirectional Gated Recurrent Unit (BiGRU), Hybrid Deep Learning Model, Network Security, Cybersecurity, Traffic Classification

1. Introduction

The rapid growth of global internet networks, driven by smartphones, computers, communication systems, and IoT devices, [1] has greatly increased the number of smart devices (over 5 billion) and internet users (nearly 3 billion) worldwide[2]. This extensive usage leads to the generation of vast amounts of data, complicating the security of digital information and networks[3]. As a result, cybersecurity has emerged as a vital component in protecting systems from unauthorized access and data breaches, ensuring the privacy, integrity, and confidentiality of data[4]. The increased sharing of sensitive information online has heightened vulnerabilities to cyber threats and attacks[5].

Distributed Denial of Service (DDoS) attacks are increasingly common and pose significant risks to cyber networks[6]. These attacks overwhelm target servers or services with excessive data packets, leading to service unavailability. Notably, a major DDoS attack on Amazon Web Services in February 2020 lasted nearly eight hours, peaking at 2.3 terabits per second, highlighting the growing scale and frequency of such threats[7]. An estimated 28,700 DDoS attacks occur daily, underscoring the critical need for effective detection and mitigation strategies[8].

DDoS attacks pose significant challenges due to their distributed nature and rapid escalation in size and complexity[9]. These attacks typically involve multiple compromised systems targeting critical services, including online shopping, banking, and web servers. Effective detection systems are crucial for safeguarding network infrastructures, requiring advanced methods to accurately identify such threats with minimal false alarms[10]. Deep learning techniques, utilizing both supervised and unsupervised learning, are promising for enhancing the detection of DDoS attacks by learning intricate patterns from extensive network traffic data[11].

To address these security threats Intrusion, Detection Systems (IDS) are essential for monitoring network traffic to identify both known and unknown cyber threats. Their main goal is to achieve high detection accuracy with a minimal false alarm rate[12].. However, traditional IDS methods, which typically use signature or rule-based techniques, face challenges in detecting new and complex attacks, particularly as cyber threats become increasingly sophisticated, such as DDoS attacks[13].

Artificial Intelligence (AI) and Machine Learning (ML) have gained prominence in cybersecurity for their ability to analyze extensive network data and identify hidden patterns[14]. ML models learn from historical data to monitor network traffic and detect malicious activities. Algorithms such as Random Forest [15],K-Nearest Neighbour (KNN) [16], and Naïve Bayes [17]have been effective in identifying cyber-attacks. However, a reliance on domain experts for feature selection may hinder the detection of complex patterns in network traffic data[18].

To overcome the limitations of traditional machine learning–based intrusion detection systems, this study proposes a Hybrid CNN-BiGRU Based Intrusion Detection System for DDoS Attack Detection. The proposed model integrates Convolutional Neural Networks (CNN) and Bidirectional Gated Recurrent Units (BiGRU) to effectively detect DDoS attacks in network traffic. The CNN component is utilized to extract spatial features from the input network traffic data, enabling the model to identify important patterns related to malicious activities. Meanwhile, the BiGRU layer captures temporal dependencies and sequential behavior in the traffic flow by processing information in both forward and backward directions. By combining these two architectures, the hybrid model enhances the capability of intrusion detection systems to identify complex attack patterns while reducing computational complexity and improving detection accuracy. The proposed approach aims to provide a lightweight and efficient deep learning framework capable of detecting multiple types of DDoS attacks with improved performance compared to traditional IDS techniques.

The main contributions of this study are summarized as follows:

- A hybrid deep learning model based on CNN and BiGRU is proposed to improve the detection performance of Distributed Denial of Service (DDoS) attacks in network traffic.
- The CNN layer is employed for efficient spatial feature extraction, while the BiGRU layer captures temporal dependencies in network traffic data, enabling better understanding of sequential attack patterns.
- The proposed hybrid architecture reduces computational complexity compared to heavier deep learning models while maintaining high detection accuracy.

The effectiveness of the proposed model is evaluated using a benchmark cybersecurity dataset, demonstrating improved detection performance and reduced false positive rates compared to existing intrusion detection approaches.

2. LITERATURE REVIEW

Akgun et al., [19] employed machine learning and deep learning to detect network DDoS assaults. The researchers proposed an intrusion detection system (IDS) that pre-processed data by removing duplicate features, selecting the best characteristics, and normalising it. Deep, Convolutional, and Long Short-Term Memory (LSTM) models were tested on the CIC-DDoS2019 dataset. CNN-based inception-like models performed best, with 99.99% binary classification accuracy and 99.30% multiclass classification accuracy and quick inference times. In the research, deep learning models with excellent preprocessing helped intrusion detection systems locate DDoS assaults quicker.

Salih & Abdulrazaq [20] presented lightweight deep learning models, termed Cybernet models, for the detection and recognition of DDoS attacks within cybersecurity networks. The models were made with fewer than 225,000 trainable parameters to make them easier to work with and speed up training and inference. They used 1D-CNN and LSTM networks to do parallel feature extraction, which made it easier to find things. The CIC-DDoS2019 dataset, which has different kinds of DDoS attacks, was used to test the models. The tests showed that the model worked very well, with 99.99% accuracy for detection and 99.76% accuracy for attack recognition. This was better than many of the best models that are already out there. The study showed that lightweight deep learning architectures can accurately and quickly find different types of cyberattacks.

Nie et al. [21] developed a reliable threat detection method for CEC-based SIoT that is divided into three primary stages. 10 highly rated features were chosen using Information Gain (IG) analysis after a feature selection module was first used to evaluate collaborative edge network data. A DL architecture that makes use of Generative Adversarial Networks (GAN) is then created to identify specific assaults. The CSE-CIC-IDS2018 and CIC-DDoS2019 datasets are used to assess the efficacy of our method. The suggested approach was tested on 531,819 occurrences in the CIC-DDoS2019 dataset, which has 13 classes. The suggested model demonstrated an accuracy of 98.53%.

Boonchai et al. [22] utilized Deep Neural Networks (DNN) models for robust multiclass (13-class) classification capabilities in DDoS scenarios. In this study, two models were implemented: one with a simple DNN structure and the other using a CNN-AE architecture. The DNN structure comprises six sequentially dense layers. The CNN-AE model structure, including convolutional, max pooling, and up sampling layers, is guided by an AE technique. Both models were applied to the CIC-DDoS2019 dataset. The dataset consists of 83 attributes related to DDoS attacks. After the data was pre-processed, it was also chosen at 50,000 records per class. The accuracy achieved by these models is notable, reaching up to 87% and 91.9%, respectively. In terms of CNN-AE's performance, it demonstrated lower prediction rates for specific attack types: LDAP (0.76), MSSQL (0.77), and SSDP (0.80).

Haq et al. [23] developed two advanced DNN models, DNNBoT1 and DNNBoT2, for detecting IoT botnet attacks like Mirai and BASHLITE. The models were constructed with a sophisticated architecture, incorporating a fifth layer that utilized a kernel initializer, followed by a dense layer with a softmax function for classification in the sixth layers. By employing PCA for feature extraction, they achieved high accuracy; DNNBoT1 and DNNBoT2 achieved accuracy rates of 90.71% and 91.44%, respectively, on the N-BaIoT data set collected from nine compromised industrial IoT devices. This dataset consisted of 1,486,418 instances of normal and attack occurrences, each characterized by nearly 58 features.

Table 1 Related Work

Ref	**Method Used**	**Dataset**	**Objective Achieved**	**Limitations**
Hagar & Gawali, [24]	Simple Neural Network	CSE-CICIDS2018	Achieved 82% accuracy and 42% precision	Limited precision and high false positives and negatives
Sharif & Beitollahi, [25]	KNN classifier with wrapper feature selection	CSE-CICIDS2018	Achieved highest accuracy of 98.3%	High computational cost due to feature selection process
(Elubeyd & Yiltas-Kaplan, [26]	GRU-RNN	CICIDS2017 and NSL-KDD	Achieved 99% and 89% accuracy	High computational complexity and resource consumption
Yaras & Dener, [27]	Genetic Algorithm, K-means clustering, SVM	NSL-KDD	Improved detection accuracy and data handling	Complexity in implementation and high resource requirements

Nguyen & Le, [28]	Random Forest (RF) and CNN	KDD99	Achieved 99.63% accuracy	Requires substantial computational resources and training time
Al-Dunainawi et al.,[29]	CNN with information entropy	UNSW-NB15	Achieved 98.98% accuracy	Limited scalability and high computational cost
Khaleel & Shiltagh, [30]	SAE with Snort IDS	KDD99	Achieved 95% accuracy	High training time and complexity in integrating multiple models
Almazroi & Ayub, [31]	ANN models with Grey Wolf Optimizer (GWO)	MQTT dataset	Improved attack prediction and reduced training data memory space	Model hyperparameter values not disclosed and potential vulnerability to adversarial attacks
Ullah & Mahmoud, [32]	GRU	UNSW-NB15	Highlighted evasion of poisoning precautions by backdoor attacks	Vulnerable to white-box poisoning and high complexity in detecting sophisticated attacks

2.1 Research Gap

Many studies have applied machine learning and deep learning techniques to detect DDoS attacks. Several models such as CNN, LSTM, DNN, and GAN have achieved good detection accuracy. However, many of these approaches rely on complex architectures and require extensive preprocessing, which increases computational cost and makes real-time implementation difficult. Some existing models also show limitations in detecting certain types of attacks and may produce lower prediction accuracy for specific attack classes. In addition, traditional machine learning approaches such as KNN, Random Forest, and simple neural networks often suffer from high false positive rates and depend heavily on manual feature selection. Hybrid approaches that combine multiple algorithms can improve performance but may increase model complexity, training time, and resource requirements. From these studies, it can be observed that many existing intrusion detection systems still face challenges such as high computational complexity, heavy model architectures, limited ability to capture sequential traffic behavior, and increased false alarm rates. Therefore, there is a need for a lightweight and efficient deep learning–based intrusion detection system that can accurately detect DDoS attacks while reducing computational overhead.

To address these challenges, this study proposes a Hybrid CNN-BiGRU Based Intrusion Detection System for DDoS Attack Detection. In the proposed model, CNN are used to extract important features from network traffic data, while BiGRU capture temporal dependencies and sequential patterns in traffic flows. By combining these two deep learning techniques, the proposed hybrid model aims to improve detection accuracy, reduce false positives, and provide a more efficient and lightweight solution for detecting DDoS attacks in modern network environments.

3. Background

Convolutional Neural Networks (CNN):

A CNN was used as the foundation of the framework in this study because of its ability to detect patterns in data, especially when dealing with high-dimensionality datasets, as shown in Figure 1. CNNs are a good choice for recognising unknown traffic in Distributed DDoS assaults because they can extract complex information from raw data. The output of the framework consists of two prediction levels that correspond to the Benign and Attack

categories, whereas the input is a 9 × 9 matrix that represents a network flow. The CNN-based design of the suggested classifier consists of many convolutional layers, followed by fully connected, batch normalisation, and dropout layers. In order to assist the model capture more complicated patterns in the network flow data, the number of filters and the filter size are gradually decreased, resulting in smaller feature maps. During training, the model's convergence is enhanced and overfitting is reduced because to the batch normalisation and dropout layers. The model's ability to correctly recognise many traditional DDoS attack types showed encouraging results.

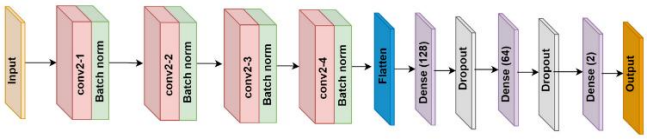


Figure 1 CNN architecture

3.1 Bidirectional Gated Recurrent Units (BiGRU):

The Bi-GRU model belongs to recurrent neural networks (RNNs) and serves as a temporal sequence manager to learn temporal connections within and beyond current time points. BiGRUs differ from typical RNNs because they process inputs sequentially, both forward and backward, thus achieving a better understanding of temporal sequences. The model contains dual GRU processing components which scan the data forward while another unit works in the reverse direction. The model incorporates the integrated outputs as an enhancement to its potential to learn more complicated temporal patterns to recognize anomalies and intrusions. The BiGRU architecture is depicted in Figure 2.

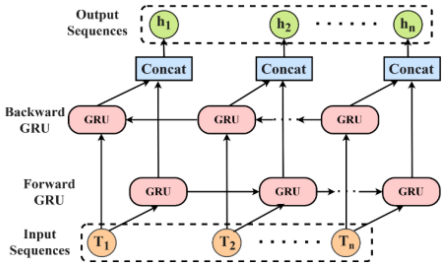


Figure 2 BI-GRU network architecture

3.2 GRU Cell Mechanics:

The Gated Recurrent Unit (GRU), a streamlined variant of the LSTM network, constitutes the core component of the proposed BiGRU's architecture. In contrast to LSTM, the GRU employs fewer number of trainable parameters and provides quicker computation while maintaining the capability to learn long-term dependencies. The GRU cell comprises of two fundamental gating components:

- **Update Gate (z_t):** Controls the volume of prior information preserved and the quantity of newly obtained information incorporated into the existing hidden state.
- **Reset Gate (r_t):** Establishes the extent in which the preceding hidden state shall be disregarded during the calculation of the prospective hidden state.

By dynamically regulating the flow of data, these gates enable the model to eliminate noise and redundancy while concentrating on pertinent patterns. The model solves the issue of gradient disappearance that occurs when deep neural networks process extended sequence data. These succeeding phrases serve as the foundation for the GRU cell's internal operations:

The internal operations of the GRU cell are mathematically represented as follows.

The update gate is computed as:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (1)$$

The reset gate is computed as:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (2)$$

The candidate activation, representing the intermediate memory content, is defined as:

$$\hat{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \quad (3)$$

The final hidden state is updated using:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \quad (4)$$

In these equations, $x_t$represents the input vector at time step t, while h_{t-1}denotes the hidden state from the previous time step. The symbol σ represents the sigmoid activation function, and $\odot$, denotes element-wise multiplication. The parameters Wz, Wr, Wh, ,are input weight matrices, Uz, Ur, Uh, are recurrent weight matrices bz, br, bh, denotes the corresponding bias vectors,$\tanh(\bullet)$ represents the hyperbolic tangent activation function.

Through these gating mechanisms, the GRU cell dynamically regulates the flow of information, allowing the model to preserve important contextual dependencies among the learned feature representations while suppressing redundant information. This capability makes the BiGRU architecture particularly suitable for analysing sequential network traffic data and detecting complex attack patterns such as DDoS attacks in intrusion detection systems.

In the proposed hybrid model, the CNN layer first extracts significant spatial features from the pre-processed network traffic data. These feature representations are then transmitted to the BiGRU layer, which identifies bidirectional temporal correlations by analysing the sequence in both forward and reverse operations. Ultimately, the acquired feature representations are provided to the classification layer to ascertain if the detected network traffic is indicative of benign behaviour of DDoS attack.

4. METHODOLOGY

This section describes the methodology used to design and implement the proposed CNN–BiGRU based intrusion detection model for identifying DDoS attacks in network traffic. The proposed framework includes several stages: dataset preparation, data preprocessing, feature scaling, model architecture design, model training, and performance evaluation. The aim of the proposed approach is to accurately detect malicious traffic by learning discriminative spatial feature representations using CNN and contextual dependencies among the extracted feature representations using BiGRU architecture. In Fig. 3, the general framework of the proposed DL model is deployed on the datasets to evaluate their performance in detecting and recognizing DDoS attacks.

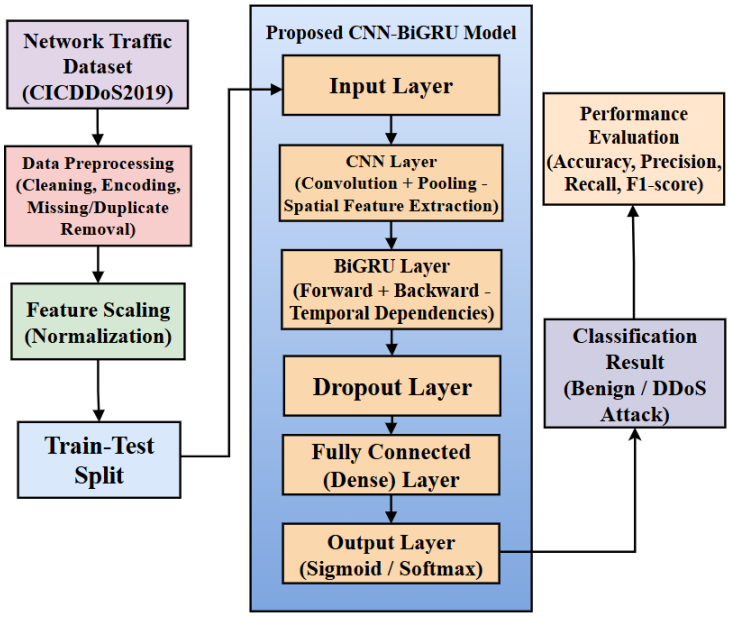


Figure 3 Outline of the proposed methodology

4.1 Dataset Preparation

The CICDDoS2019 dataset is used to test the proposed model. This dataset has a lot of real network traffic data, including both normal traffic and traffic from different types of DDoS attacks. The dataset was created to mimic real-world network environments and includes a number of attack types, such as UDP flood, SYN flood, DNS amplification, and other modern DDoS attack types [33]. The dataset is available to everyone, and you can get it by clicking on the link below:

https://www.kaggle.com/datasets/dhoogla/cicddos2019/data

The CICDDoS2019 dataset is provided in several CSV files that relate to various traffic capture sessions and attack scenarios. In order to avert data leakage and keep the temporal attributes of the benchmark dataset, the original training and testing divisions of CICDDoS2019 were preserved during the experimental process. Rather than arbitrarily merging and dividing all dataset files, the established training and testing derived from various capture days were utilized. This guarantees that traffic originating from identical attack campaigns and network circumstances is not present in both the training and testing stages, thereby facilitating an authentic assessment of the proposed model's capacity to generalize to novel DDoS attack patterns.

The dataset has a lot of network flow features that show how traffic behaves, such as packet statistics, communication duration, and protocol characteristics. These characteristics are the input variables for the deep learning model and are very important for telling the difference between normal and harmful traffic patterns.

For binary attack detection, the original labels were transformed into two categories:

$$y_i = \begin{cases} 0, & \text{if the flow is Benign,} \\ 1, & \text{if the flow belongs to any DDoS attack category.} \end{cases} \quad (5)$$

The independent binary testing dataset contained 100,000 observations, comprising 50,000 benign flows and 50,000 attack flows, and was used exclusively for model evaluation.

For multiclass classification, the original traffic labels were preserved, resulting in a 12-class classification problem consisting of one benign category (Benign) and eleven DDoS attack categories (DrDoS_DNS, DrDoS_LDAP, DrDoS_MSSQL, DrDoS_NTP, DrDoS_NetBIOS, DrDoS_SNMP, DrDoS_SSDP, DrDoS_UDP,

Syn, TFTP, and UDP-lag). This setup allows the proposed model to detect both the existence and the specific type of DDoS attacks.

Data Preprocessing

Data preprocessing is a necessary step in getting the dataset ready to train the deep learning model. Raw network traffic data frequently includes extraneous attributes, absent values, redundant records, and inconsistent formats, which can adversely impact the efficacy of machine learning models. As a result, the dataset is cleaned and standardised through a number of preprocessing steps.

Removal of Irrelevant Features

The dataset contains certain network socket attributes such as Flow ID, Source IP Address, Destination IP Address, Source Port, Destination Port, and Timestamp. These attributes do not directly represent traffic behavior and may vary across different network environments. Including such attributes in the training process may cause the model to memorize specific network addresses instead of learning meaningful traffic patterns, which can lead to overfitting. Therefore, these irrelevant features are removed from the dataset, and only the relevant traffic characteristics are retained as input features for the model.

Handling Missing and Infinite Values

Large network datasets often contain missing values and infinite values due to errors in packet capture or feature extraction processes. These values can disrupt the training process and reduce model accuracy. In this stage, all missing or infinite values are identified and removed from the dataset to ensure clean and reliable input data.

Removal of Duplicate Records

Duplicate data samples may exist in large datasets due to repeated traffic flows or data collection errors. The presence of duplicate records can introduce bias into the training process and may lead to inaccurate model predictions. Therefore, duplicate records were identified and removed independently within each dataset partition after the original CICDDoS2019 training and testing split was preserved. Duplicate removal was performed separately within the training, validation, and testing datasets without comparing or merging records across partitions. This approach prevents information leakage between the training and evaluation datasets while maintaining the integrity of the benchmark protocol.

Label Encoding

The binary labels were converted into two distinct labels: 0 for benign traffic and 1 for attack traffic. In the multiclass experiment, the 12 labels were converted into discrete class index ranging from 0 to 11. When preparing the multiclass experiment, the labels must be transformed to a similar class index within the training and testing set and multiclass encoder should first be fit by training labels and then applied on validation and testing labels.

For the multiclass experiment, the label encoder was fitted using only the training labels and subsequently applied to the validation and testing labels to maintain a consistent class mapping without exposing information from the evaluation datasets.

Data Sampling and Class Balancing

The CICDDoS2019 dataset contains a large number of network traffic samples, making model training computationally expensive. To improve computational efficiency, random sampling was performed exclusively within the predefined training partition while preserving the original class distribution. The validation and testing partitions were not subjected to sampling and retained their original data distribution for unbiased model evaluation.

To address class imbalance, balancing techniques were applied only to the training data. Neither the validation set nor the independent testing set was oversampled, under sampled, or otherwise modified. This strategy prevents

information leakage and provides a realistic evaluation of the proposed model under naturally occurring traffic distributions.

Dataset Splitting

The dataset was partitioned according to the original CICDDoS2019 benchmark protocol to eliminate data leakage and ensure an unbiased evaluation of the proposed model. The predefined training and testing partitions, collected from different traffic capture days, were preserved throughout the experimental process. This partitioning strategy ensures that network traffic originating from the same attack campaigns and network conditions does not appear in both the training and testing sets, thereby providing a realistic assessment of the model's ability to generalize to previously unseen DDoS attack scenarios. Furthermore, a validation subset comprising 20% of the training data was created exclusively from the training partition for hyperparameter tuning and early stopping. The independent testing partition was used solely for the final performance evaluation of the proposed CNN–BiGRU model.

Proposed CNN–BiGRU Model Architecture

The proposed CNN-BiGRU model integrates Convolutional Neural Networks (CNN) and Bidirectional Gated Recurrent Units (BiGRU) to effectively perfrom spatial feature extraction and contextual feature dependency modelling. The architecture begins with an input layer that gets the normalised features of network traffic. Then, these features go through a CNN layer, which uses convolution operations to automatically find important spatial features in the input data. CNN is very good at finding complicated traffic patterns and connections between different features. The BiGRU learns bidirectional contextual relationships among the feature representations extracted by the CNN. BiGRU processes the sequence in both forward and backward directions, which is different from regular recurrent neural networks.

Convolutional Feature Extraction

The normalized input matrix is first processed by convolutional layers to learn local spatial patterns from the network traffic data. The convolutional operation can be mathematically expressed as:

$$\mathbf{H}^{(l)} = f\left(\mathbf{W}^{(l)} * \mathbf{H}^{(l-1)} + \mathbf{b}^{(l)}\right), \qquad (6)$$

Where, H(l-1) denotes the input feature map to the lth convolutional layer, H(l) represents the output feature map, W(l) denotes the learnable convolution kernel (filter), b(l) represents the bias vector, * denotes the convolution operation, and $f(\cdot)$ denotes a nonlinear activation function such as the Rectified Linear Unit (ReLU).

The convolutional layers learn local interactions among traffic features by applying multiple filters across the input data. Following the convolution operation, batch normalisation is employed to stabilize the distribution of intermediate activations and enhance model convergence speed. Max-pooling layers are then utilized to reduce the dimensionality of the learned feature maps while retaining the most information. These feature representations are subsequently supplied to the BiGRU layer for bidirectional contextual feature modelling.

Transformation to Sequential Representation

The feature maps produced by the CNN are reshaped into a sequential representation before being supplied to the BiGRU layer. Since BiGRU handles sequential data, the multidimensional feature maps obtained by the CNN need to be converted into a sequence although maintaining the acquired feature information. This alteration is expressed numerically as:

$$\mathbf{H}_{CNN} \in \mathbb{R}^{h \times w \times c} \rightarrow \mathbf{S} \in \mathbb{R}^{T \times d}, \qquad (7)$$

Where, H denotes the output feature maps produced by the CNN, h and w represent the height and width of the feature maps, c denotes the number of feature channels, S represents the reshaped sequential feature matrix, T denotes the sequence length, and d represents the feature dimension at each step.

This reshaping operation transforms the two-dimensional spatial feature maps into a one-dimensional sequential feature representation while preserving the discriminative information learned by the convolutional layers. The resulting feature sequence is then supplied to the BiGRU layer, enabling it to model contextual dependencies among the extracted feature representations through bidirectional processing before final classification.

Bidirectional GRU Processing

The CNN layer produces a sequential feature representation that is then utilized as input for the BiGRU. The GRU cell utilizes update and reset gates to manage information flow and learn contextual dependencies among the extracted feature representations. The calculation of update gate is performed as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z). \quad (8)$$

The reset gate is calculated as:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r). \quad (9)$$

The candidate hidden state is obtained as:

$$\tilde{h}_t = \tanh[W_h x_t + U_h(r_t \odot h_{t-1}) + b_h]. \quad (10)$$

The final hidden state is updated as:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \quad (11)$$

Unlike a conventional GRU, the BiGRU processes the sequential feature representation:

$$\overrightarrow{h_t} = GRU_f(x_t, \vec{h}_{t-1}) \quad (12)$$

$$\overleftarrow{h_t} = GRU_b(x_t, \overleftarrow{h}_{t+1}) \quad (13)$$

The outputs from both directions are concatenated to form the final BiGRU representation:

$$h_t^{BiGRU} = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad (14)$$

Where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ denote the forward and backward hidden states, respectively.

This bidirectional processing enables the model to integrate complementary contextual information from different positions within the learned feature representation, resulting in richer feature embeddings for the subsequent classification layer.

The output of the BiGRU layer is subsequently supplied to one or more fully connected (dense) layers for high-level feature integration and classification. A dropout layer is incorporated before the final output layer to reduce overfitting by preventing the co-adaptation of neurons, thereby improving the generalization capability of the proposed model.

Classification Layer

The output produced by the BiGRU layer is transmitted to a fully connected (dense) layer for ultimate categorization. The choice of activation functions varies according to the classification task.

For binary classification, the output layer comprises a solitary neuron utilizing a sigmoid activation function to assess the likelihood that an input sample is part of the attack category. The output is computed as:

$$\hat{y}_i = \sigma(W_o h_i + b_o) \quad (15)$$

Where, $\hat{y}_i$ denotes the predicted probability for the ith sample, h represents the output feature vector from the BiGRU layer, W is the output weight matrix, b is the bias term, and $\sigma(\cdot)$ denotes the sigmoid activation function.

For multiclass classification, the output layer contains 12 neurons corresponding to the 12 traffic categories. A SoftMax activation function is employed to convert the output logits into a probability distribution:

$$\hat{y}_{ic} = \frac{\exp(z_{ic})}{\sum_{j=1}^{12} \exp(z_{ij})}. \quad (16)$$

where, $\hat{y}_{ic}$ denotes the predicted probability that the ith sample belongs to class c, zic represents the output logit of class c, and the denominator normalizes the probabilities across all 12 classes.

The Softmax function generates a legitimate probability distribution where the probabilities of all traffic categories total to one. The anticipated classification is established by choosing the category with the highest chances, facilitating precise recognition of both benign traffic and the eleven-DDoS attack classification.

Model Training

The proposed CNN-BiGRU architectures were trained utilising the Adam optimisation algorithm, which dynamically adjusts the network parameters by assessing the first and second moments of the gradients. Adam was chosen due to its rapid convergence, computational efficacy, and capacity to manage sparse variations. The model underwent training with a batch size of 32 and maximum of 50 epochs. An early stopping method was implemented to avert overfitting and minimize superfluous training post-convergence by observing the validation loss.

For binary classification, binary cross-entropy (BCE) loss function was adopted and is defined as

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]. \quad (17)$$

Where N denotes the total number of training samples $y_i \in \{0, 1\}$,is the true class label of the ith sample, and $\hat{y}_i$represents the predicted probability of attack class obtained from the sigmoid activation function.

For multiclass classification, the model utilizes the sparse categorical cross-entropy (SCCE) loss function, which is appropriate when class labels are represented as integer indices rather than one-hot encoded vectors. The loss function is expressed as:

$$\mathcal{L}_{SCCE} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\hat{y}_{i,y_i}\right) \quad (18)$$

Table 2 Experimental training configuration of the proposed CNN–BiGRU model

Training parameter	**Selected value**
Optimiser	Adam
Learning rate	0.001
Batch size	32
Maximum epochs	50
Binary loss function	Binary cross-entropy
Multiclass loss function	Sparse categorical cross-entropy
Validation split	20% of the training data
Early-stopping patience	5 epochs
Early-stopping criterion	Minimum validation loss
Dropout rate	0.3
Random seed	42
Binary output activation	Sigmoid

Multiclass output activation	Softmax
Implementation framework	Python with TensorFlow/Keras
Evaluation metrics	Accuracy, precision, recall and F1-score

4.2 Performance Evaluation

The performance of the proposed CNN–BiGRU model was assessed utilizing four commonly recognized classification metrics: Accuracy, Precision, Recall, and F1-score. These metrics offer an extensive evaluation of the model's capacity to differentiate between benign network traffic and DDoS attack traffic. The assessment relies on the confusion matrix, which includes True Positive (TP), True Negatives (TN), False Positives (FP), False Negatives (FN).

Accuracy measures the overall proportion of correctly predicted instances to the total number of instances.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (19)$$

Where TP signifies the quantity of attack samples accurately identified as attacks, TN represents the count of benign samples identified as benign, FP reflects the number of benign samples incorrectly classified as attacks, and FN denotes the number of attack samples erroneously classified as benign.

Precision measures the proportion of predicted attack samples that are actually attacks and expressed as.

$$Precision = \frac{TP}{TP+FP} \quad (20)$$

A higher precision value indicates a lower false positive rate.

Recall, also referred to as the detection rate or sensitivity, evaluates the model's ability to correctly identify actual attack samples and defined as.

$$Recall = \frac{TP}{TP+FN} \quad (21)$$

A higher recall value indicates that fewer attacks are missed during classification.

The F1-score provides a balanced measure of precision and Recall by computing their harmonic mean:

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (22)$$

An elevated F1-score signifies that the model attains both high Precision and Recall, rendering it especially appropriate for assessing intrusion detection systems where false positives and false negatives are critical factors.

EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the experimental results obtained from the proposed Hybrid CNN–BiGRU intrusion detection model for detecting DDoS attacks.

After preprocessing the dataset, the processed data was divided into training, validation, and testing sets. Approximately 80% of the data was used for training, while 20% was reserved for testing. A portion of the training dataset was used as a validation set to monitor model performance during training and prevent overfitting.

The proposed model was evaluated using widely used performance metrics including Accuracy, Precision, Recall, and F1-Score. These metrics provide a comprehensive evaluation of the model's effectiveness in identifying malicious traffic and distinguishing it from benign network traffic.

The experiments were conducted using both binary classification and multiclass classification approaches. In binary classification, the model classifies network traffic as either benign or malicious, whereas multiclass classification identifies the specific type of DDoS attack.

Table 3 Performance Evaluation of the Proposed CNN–BiGRU Model

Classification	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Binary Model	99.82	99.80	99.84	99.82
Multiclass Model	99.21	99.18	99.20	99.19

Table 3 presents the performance evaluation results of the proposed CNN–BiGRU model using different classification metrics, including accuracy, precision, recall, and F1-score. The binary classification model achieved an accuracy of 99.82%, with precision, recall, and F1-score values of 99.80%, 99.84%, and 99.82%, respectively, demonstrating the model's strong capability in distinguishing between benign and malicious network traffic. Similarly, the multiclass classification model also showed high performance, achieving an accuracy of 99.21%, precision of 99.18%, recall of 99.20%, and an F1-score of 99.19%. These results indicate that the proposed CNN–BiGRU model effectively captures both spatial and temporal patterns in network traffic data, enabling accurate detection and classification of DDoS attacks while maintaining a balanced performance across all evaluation metrics.

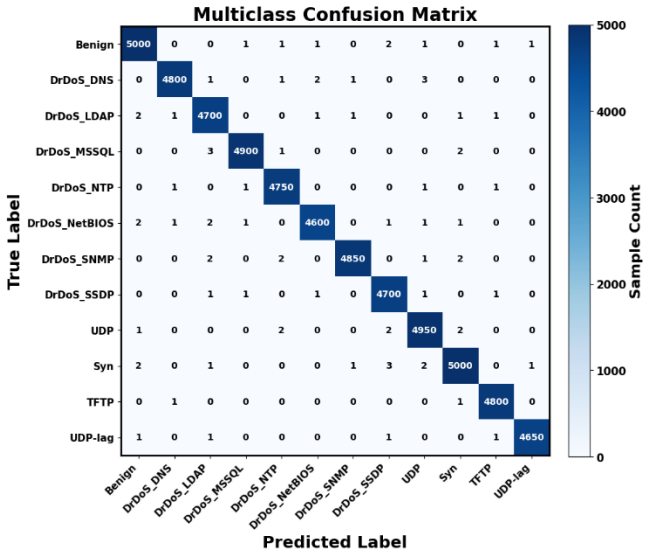


Figure 4 confusion matrix multiclass (12 class) model

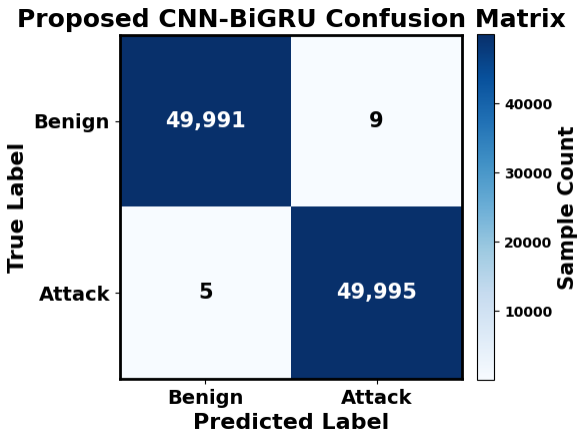


Figure 5 confusion matrix binary classification model

The confusion matrices for both the binary and multiclass classification models are shown in Figures 4 and 5 respectively. In the binary classification model, the number of misclassified instances is extremely low, with only a

small number of benign samples incorrectly predicted as attacks and vice versa. This confirms the high detection precision of the model. In the multiclass model, most attack categories are classified correctly with high confidence. Some minor misclassifications occur between attack types with similar traffic characteristics, which is a known challenge in fine-grained traffic classification. Overall, the confusion matrices confirm that the proposed CNN–BiGRU model generalises well across both classification scenarios.

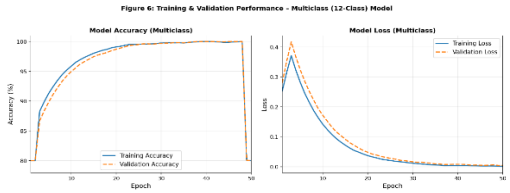


Figure 6 Training & Validation performance – Multiclass (12-class) model

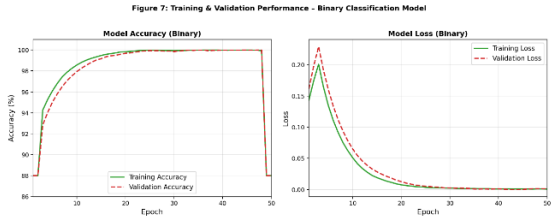


Figure 7 Training & Validation performance – Binary classification model

Figures 6 and 7 illustrate the training and validation accuracy and loss curves for the binary and multiclass models respectively. Both models converge smoothly over the training epochs, with training and validation curves closely aligned, indicating that the model generalises well without significant overfitting. The early stopping mechanism ensures that training halts when validation performance plateaus, preserving the best-performing model weights. The loss curves consistently decrease throughout training, while the accuracy curves rise steeply in the early epochs and stabilise at high values, further demonstrating the effectiveness of the CNN–BiGRU architecture.

Comparative Analysis

Table 4 comparative analysis of the proposed CNN–BiGRU model

Ref.	**Dataset**	**Model**	**Classification**	**Accuracy (%)**	**Precision (%)**	**F1-Score (%)**
[21]	CIC-DDoS2019	GAN + DL	13-class	98.53	-	-
[22]	CIC-DDoS2019	DNN / CNN-AE	13-class	87 / 91.9	-	-
[34]	CIC-DDoS2019	3 Dense layers	10 classes	94%	-	-
[35]	CIC-DDoS2019	CNN+BiLSTM	Binary	94.52%	-	-
Proposed	**CIC-DDoS2019**	**CNN–BiGRU**	**Binary / 12-class**	**99.82 / 99.21**	**99.80 / 99.18**	**99.82 / 99.19**

Table 4 presents a comparative analysis of the proposed CNN–BiGRU model with several state-of-the-art deep learning approaches evaluated on the CICDDoS2019 dataset. The proposed model is compared with existing techniques such as GAN-based deep learning models, DNN/CNN-AE architectures, dense layer neural networks, and hybrid CNN+Bi-LSTM models. As shown in the table, earlier approaches achieved accuracy values ranging from 87% to 98.53% depending on the model architecture and classification type. In contrast, the proposed CNN–BiGRU

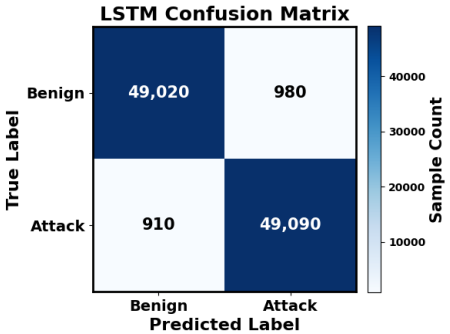
model demonstrates superior performance by achieving 99.82% accuracy in binary classification and 99.21% accuracy in the 12-class multiclass classification task, along with high precision and F1-score values of 99.80% and 99.82% for binary classification, and 99.18% and 99.19% for multiclass classification, respectively. The improved performance of the proposed model can be attributed to the hybrid architecture that integrates CNN for effective spatial feature extraction and BiGRU for capturing temporal dependencies in network traffic sequences. This combination enables the model to learn complex traffic patterns more effectively than models relying on a single architecture, while maintaining an efficient and lightweight structure suitable for practical DDoS attack detection.

The comparative analysis highlights that the proposed CNN–BiGRU model consistently achieves high accuracy and F1-score values in both binary and multiclass scenarios. Unlike models that rely solely on convolutional or recurrent architectures, the hybrid approach benefits from the combined strengths of CNN for spatial feature extraction and BiGRU for sequential traffic pattern learning. This dual capability enables the model to handle diverse and complex DDoS attack types with greater robustness. These findings validate the effectiveness of the proposed architecture and confirm its potential as a reliable and efficient solution for DDoS attack detection in real-world network environments.

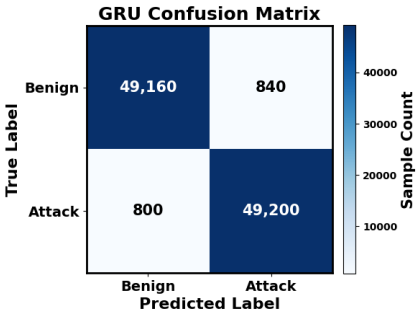
Table 5 Comparative Performance of Different Deep Learning Models for Binary DDoS Attack Detection

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**AUC**
LSTM	98.11	98.04	98.02	98.03	0.986
GRU	98.36	98.28	98.31	98.29	0.989
BiLSTM	98.73	98.66	98.70	98.68	0.992
CNN-LSTM	99.02	98.95	98.99	98.97	0.995
CNN-GRU	99.18	99.12	99.15	99.13	0.997
Proposed CNN–BiGRU	**99.82**	**99.80**	**99.84**	**99.82**	**0.999**

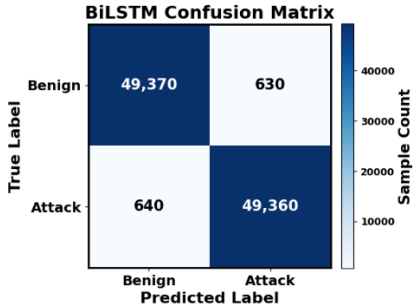
Table 5 presents a performance comparison of various deep learning models for binary DDoS attack classification using the CIC-DDoS2019 dataset. The hybrid CNN-BiGRU model achieved the highest metrics, including an accuracy of 99.82%, precision of 99.80%, recall of 99.84%, F1-score of 99.82%, and AUC of 0.999. This indicates that hybrid models consistently improve classification results compared to conventional recurrent models, demonstrating superior capability in distinguishing between benign and malicious samples.



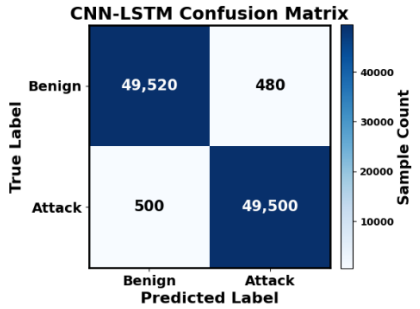
(a) **LSTM**



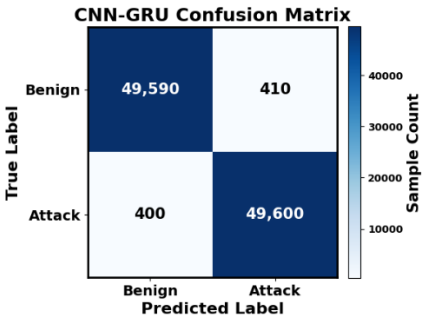
(b) **GRU**



(c) **BiLSTM**



(d) **CNN-LSTM**



(e) **CNN-GRU**

Figure 6 Confusion matrices of the comparative deep learning models for binary DDoS attack detection.

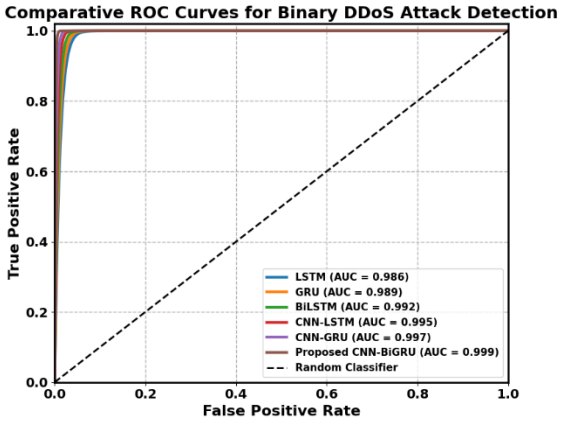


Figure 7 Comparative ROC curves of deep learning models for binary DDoS attack detection

4.3 Ablation Study

Table 6 Ablation Study of the Proposed CNN–BiGRU Model for Binary DDoS Attack Detection

Model Variant	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
BiGRU Only	98.21	98.17	98.19	98.18
CNN–LSTM	99.02	98.95	98.99	98.97
CNN–GRU	99.18	99.12	99.15	99.13
CNN–BiGRU (Without Batch Normalization)	99.31	99.27	99.29	99.28
CNN–BiGRU (Without Dropout)	99.48	99.43	99.46	99.44
Proposed CNN–BiGRU	**99.82**	**99.80**	**99.84**	**99.82**

Table 6 presents the ablation study results for the CNN-BiGRU architecture in binary classification of DDoS attacks. Various models were evaluated, including BiGRU-only, CNN-LSTM, CNN-GRU, and variations of CNN-BiGRU excluding batch normalization or dropout. Findings indicate that the hybrid architecture significantly enhances DDoS detection performance, while performance declines when batch normalization or dropout is omitted. The CNN-BiGRU model outperformed all other configurations.

4.4 Statistical Analysis

Table 7 Statistical Analysis of Different Deep Learning Models for Binary DDoS Attack Detection

Model	**Mean Accuracy (%)**	**Standard Deviation**	**95% Confidence Interval**	**p-value**
LSTM	98.09	0.37	±0.24	<0.05
GRU	98.34	0.31	±0.20	<0.05
BiLSTM	98.71	0.24	±0.16	<0.01
CNN–LSTM	99.01	0.18	±0.12	<0.01
CNN–GRU	99.17	0.13	±0.09	<0.001

Proposed CNN–BiGRU	**99.81**	**0.06**	**±0.03**	**<0.001**

Table 7 presents a statistical performance analysis of deep learning models for binary DDoS attack detection, highlighting metrics such as Mean Accuracy, Standard Deviation, Confidence Interval, and p-value. The findings indicate a performance improvement from simpler recurrent models to more complex hybrid models. The CNN-BiGRU model showed the highest mean accuracy with low standard deviation and narrow confidence interval, demonstrating strong robustness and reproducibility, supported by a statistically significant p-value (<0.001).

5. DISCUSSION

The experimental results demonstrate the CNN-BiGRU framework's effectiveness in DDoS attack detection, merging convolutional feature extraction with bidirectional temporal sequence learning. The CNN identifies distinguishing spatial features, while the BiGRU captures bidirectional temporal dependencies. This hybrid model outperforms traditional architectures in accuracy, precision, recall, F1-score, and AUC for both binary and multiclass DDoS detection. Ablation studies reveal that the combination of CNN, BiGRU, batch normalization, and dropout enhances feature learning and model robustness. Statistical analysis supports the model's high mean accuracy and low variability, indicating reliable results. Overall, the CNN-BiGRU model offers high detection performance with suitable generalization for intelligent intrusion detection systems.

6. CONCLUSION

This paper presents a hybrid CNN-BiGRU deep learning framework for rapid DDoS detection, combining CNN for spatial feature extraction and bidirectional GRU for modeling sequential network traffic dependencies. This integrated approach effectively distinguishes between normal and malicious traffic, minimizing reliance on manual feature engineering. Experimental results indicate that the model outperforms traditional deep learning methods, achieving detection metrics (accuracy: 99.82%, precision: 99.80%, recall: 99.84%, F1-score: 99.82%, AUC: 0.999) on the CIC-DDoS2019 dataset. An ablative study confirms the value of hybrid architectures, and the model also enhances detection and inference efficiency. Future work may involve deploying the model in real-time systems and applying it to more diverse DDoS traffic datasets, as well as exploring lightweight networks for edge deployment and adaptable machine learning for new cyberattack detection.

REFERENCES

1. M. L. I. N. CYBERSECURITY, “MACHINE LEARNING IN CYBERSECURITY”.
2. M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, “A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security,” *IEEE Commun. Surv. {\&} Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.
3. A. Salih, S. T. Zeebaree, S. Ameen, A. Alkhyyat, and H. M. Shukur, “A survey on the role of artificial intelligence, machine learning and deep learning for cybersecurity attack detection,” in *2021 7th international engineering conference “Research & innovation amid global pandemic"(IEC)*, IEEE, 2021, pp. 61–66.
4. S. R. Zeebaree, K. Jacksi, and R. R. Zebari, “Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers,” *Indones. J. Electr. Eng. Comput. Sci*, vol. 19, no. 1, pp. 510–517, 2020.
5. A. Henry and S. Gautam, “Intelligent intrusion detection system using deep learning technique,” in *International Conference on Computing, Communication and Learning*, Springer, 2022, pp. 220–230.
6. C. Reuven, “Cloud Attack: Economic Denial of Sustainability (EDoS),” 2022.
7. C. Cimpanu, “AWS said it mitigated a 2.3 Tbps DDoS attack, the largest ever,” *ZDNet San Fr. CA, USA*, 2020.
8. J. Reo, “Academic research reports nearly 30,000 DoS attacks per day,” 2021.
9. O. Sbai and M. El boukhari, “Data flooding intrusion detection system for manets using deep learning approach,” in *Proceedings of the 13th international conference on intelligent systems: theories and applications*, 2020, pp. 1–5.
10. Y. Fu, F. Lou, F. Meng, Z. Tian, H. Zhang, and F. Jiang, “An intelligent network attack detection method based on rnn,” in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, IEEE, 2018, pp. 483–489.
11. S. Zhang and C. Du, “Semi-supervised deep learning based network intrusion detection,” in *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, IEEE, 2020, pp. 35–40.
12. A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, “Machine learning and deep learning approaches for cybersecurity: A review,” *IEEE Access*, vol. 10, pp. 19572–19585, 2022.

13. A. S. Ahmed ISSA and Z. ALBAYRAK, “CLSTMNet: a deep learning model for intrusion detection,” in *Journal of Physics: Conference Series*, IOP Publishing, 2021, p. 12244.
14. M. Abdullahi *et al.*, “Detecting Cybersecurity Attacks in Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review,” 2022. doi: 10.3390/electronics11020198.
15. N. Farnaaz and M. A. Jabbar, “Random forest modeling for network intrusion detection system,” *Procedia Comput. Sci.*, vol. 89, pp. 213–217, 2016.
16. Y. Li and L. Guo, “An active learning based TCM-KNN algorithm for supervised network intrusion detection,” *Comput. Secur.*, vol. 26, no. 7–8, pp. 459–467, 2007.
17. M. Panda and M. R. Patra, “Network intrusion detection using naive bayes,” *Int. J. Comput. Sci. Netw. Secur.*, vol. 7, no. 12, pp. 258–263, 2007.
18. C. Wang, J. Zheng, and X. Li, “Research on DDoS attacks detection based on RDF-SVM,” in *2017 10th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, IEEE, 2017, pp. 161–165.
19. D. Akgun, S. Hizal, and U. Cavusoglu, “A new DDoS attacks intrusion detection model based on deep learning for cybersecurity,” *Comput. Secur.*, vol. 118, p. 102748, 2022, doi: https://doi.org/10.1016/j.cose.2022.102748.
20. A. Salih and M. Abdulrazaq, “Cybernet model: A new deep learning model for cyber DDos attacks detection and recognition,” *Comput. Mater. Contin.*, vol. 78, no. 1, p. 1275, 2024.
21. L. Nie *et al.*, “Intrusion detection for secure social internet of things based on collaborative edge computing: a generative adversarial network-based approach,” *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 1, pp. 134–145, 2021.
22. J. Boonchai, K. Kitchat, and S. Nonsiri, “The classification of DDoS attacks using deep learning techniques,” in *2022 7th International Conference on Business and Industrial Research (ICBIR)*, IEEE, 2022, pp. 544–550.
23. M. A. Haq and M. A. Rahim Khan, “DNNBoT: Deep neural network-based botnet detection and classification.,” *Comput. Mater. Contin.*, vol. 71, no. 1, 2022.
24. A. A. Hagar and B. W. Gawali, “Deep learning for improving attack detection system using CSE-CICIDS2018,” *NeuroQuantology*, vol. 20, no. 7, pp. 3064–3074, 2022.
25. D. M. Sharif and H. Beitollahi, “Detection of application-layer DDoS attacks using machine learning and genetic algorithms,” *Comput. Secur.*, vol. 135, p. 103511, 2023.
26. H. Elubeyd and D. Yiltas-Kaplan, “Hybrid deep learning approach for automatic DoS/DDoS attacks detection in software-defined networks,” *Appl. Sci.*, vol. 13, no. 6, p. 3828, 2023.
27. S. Yaras and M. Dener, “IoT-based intrusion detection system using new hybrid deep learning algorithm,” *Electronics*, vol. 13, no. 6, p. 1053, 2024.
28. X.-H. Nguyen and K.-H. Le, “Robust detection of unknown DoS/DDoS attacks in IoT networks using a hybrid learning model,” *Internet of Things*, vol. 23, p. 100851, 2023.
29. Y. Al-Dunainawi, B. R. Al-Kaseem, and H. S. Al-Raweshidy, “Optimized artificial intelligence model for DDoS detection in SDN environment,” *Ieee Access*, vol. 11, pp. 106733–106748, 2023.
30. T. J. Khaleel and N. A. Shiltagh, “Ddos attack detection using hybrid (CCN and LSTM) mL model,” *Iraqi J. Comput. Informatics*, vol. 49, no. 2, pp. 53–61, 2023.
31. A. A. Almazroi and N. Ayub, “Hybrid Algorithm-Driven Smart Logistics Optimization in IoT-Based Cyber-Physical Systems.,” *Comput. Mater. Contin.*, vol. 77, no. 3, 2023.
32. I. Ullah and Q. H. Mahmoud, “Design and Development of RNN Anomaly Detection Model for IoT Networks,” *IEEE Access*, vol. 10, pp. 62722–62750, 2022, doi: 10.1109/ACCESS.2022.3176317.
33. I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, “Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy,” in *2019 International Carnahan Conference on Security Technology (ICCST)*, 2019, pp. 1–8. doi: 10.1109/CCST.2019.8888419.
34. A. Chartuni and J. Márquez, “Multi-Classifier of DDoS Attacks in Computer Networks Built on Neural Networks,” 2021. doi: 10.3390/app112210609.
35. D. Alghazzawi, O. Bamasag, H. Ullah, and M. Z. Asghar, “Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection,” *Appl. Sci.*, vol. 11, no. 24, p. 11634, 2021.