applied sciences

MDPI

Article

Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection

Daniyal Alghazzawi [1], Omaima Bamasaq [2], Hayat Ullah [3] and Muhamad Zubair Asghar [3,*]

[1] Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 80200, Saudi Arabia; dghazzawi@kau.edu.sa
[2] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 80200, Saudi Arabia; obamasek@kau.edu.sa
[3] Institute of Computing and Information Technology (ICIT), Gomal University, Dera Ismail Khan 29220, Pakistan; hayyat.ullah2468@gmail.com
* Correspondence: zubair@gu.edu.pk

Abstract: DDoS (Distributed Denial of Service) attacks have now become a serious risk to the integrity and confidentiality of computer networks and systems, which are essential assets in today's world. Detecting DDoS attacks is a difficult task that must be accomplished before any mitigation strategies can be used. The identification of DDoS attacks has already been successfully implemented using machine learning/deep learning (ML/DL). However, due to an inherent limitation of ML/DL frameworks—so-called optimal feature selection—complete accomplishment is likewise out of reach. This is a case in which a machine learning/deep learning-based system does not produce promising results for identifying DDoS attacks. At the moment, existing research on forecasting DDoS attacks has yielded a variety of unexpected predictions utilising machine learning (ML) classifiers and conventional approaches for feature encoding. These previous efforts also made use of deep neural networks to extract features without having to maintain the track of the sequence information. The current work suggests predicting DDoS attacks using a hybrid deep learning (DL) model, namely a CNN with BiLSTM (bidirectional long/short-term memory), in order to effectively anticipate DDoS attacks using benchmark data. By ranking and choosing features that scored the highest in the provided data set, only the most pertinent features were picked. Experiment findings demonstrate that the proposed CNN-BI-LSTM attained an accuracy of up to 94.52 percent using the data set CIC-DDoS2019 during training, testing, and validation.

Keywords: deep learning; DDoS attacks; hybrid deep learning; feature selection

check for updates

Citation: Alghazzawi, D.; Bamasaq, O.; Ullah, H.; Asghar, M.Z. Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection. *Appl. Sci.* **2021**, *11*, 11634. https://doi.org/10.3390/app112411634

Academic Editor: Jerry Chun-Wei Lin

Received: 23 October 2021
Accepted: 30 November 2021
Published: 8 December 2021

1. Introduction

DoS (Denial of Service) attacks diminish a particular system's network bandwidth and computational resources by overloading it with malicious traffic, blocking it from providing normal services to authorized users. A DoS attack is a cyber-attack in which an attacker attempts to make systems and servers inaccessible, preventing consumers from accessing servers and resources. DDoS (Distributed Denial of Service) [1] takes things a step further on a wider scale. Distributed Denial of Service (DDoS) attacks are DoS attacks that are executed in a distributed way to increase the resource usage for one or more targets [2].

As seen in Figure 1, DDoS attacks seize control of the majority number of compromised systems, known as a botnet, and execute coordinated attacks on the target machine. DDoS attacks are developing and increasing in magnitude, frequency, and complexity in tandem with the introduction and growth of innovative Web-based technologies. Companies confront possible network risks that might have serious consequences for their activities, such as outages, data theft, or even blackmail threats from cybercriminals [3].

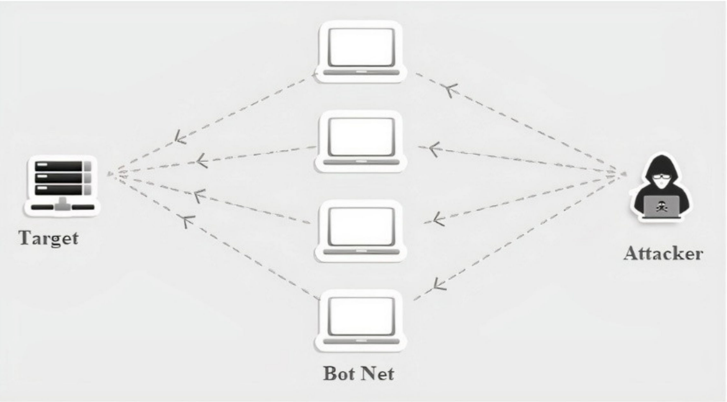


Figure 1. A botnet-driven DDoS attack.

Measures for DDoS mitigation should have been performed in the case of DDoS attacks, as described in [4]. Before any mitigation strategies can be used, DDoS strikes must first be detected. DDoS attacks were initially detected by traffic engineers using criteria they had written. This strategy appears to have fallen behind the changing and developing pattern of DDoS attacks. Academics and industry are studying the prospect of implementing machine learning/deep learning (ML/DL) for DDoS attack detection as ML/DL unlocks its potentiality in several domains. In identifying risks, conventional manual approaches have limited performance and a high latency. Attacks can be caught faster rapidly and effectively using machine learning techniques like Naïve bayes Bayesian, K-nearest neighbors, and support vector machine [5]. In machine learning, features for classification should always be chosen by humans or by feature selection algorithms. Selection of features, on the other hand, is an essential element of DL. Deep learning methods like CNN and recurrent neural networks use a succession of nonlinear processing elements to learn several levels of data interpretation from a large number of labeled data. As a result, DL can be a useful technique for DDoS attack detection [2]. DDoS detection using machine learning and deep learning has been shown to be successful. Section 2 will look at some prominent cases of ML/DL use for DDoS attack detection. We use the Bi-Directional Long Short-Term Memory (BI-LSTM) after considering various possibilities.

1.1. *Research Motivation (the Need to Predict DDoS Attacks)*

Intrusion detection systems (IDSs) are used in cyber security to identify and remove harmful activity [5]. As the number of malicious attacks continues to grow dramatically, IDSs are tasked with the responsibility of preventing such attacks from wreaking havoc on a broad region of cyberspace. DDoS attacks are a significant security risk in Iot environments. According to statistics from Amazon Web Services, the largest DDoS attack to date occurred in February 2020. The attack's highest network data is estimated to reach 2.3 Tbps. The hackers chose to exploit seized CLDAP webservers (Connection-less Lightweight Directory Access Protocol webservers), a protocol that is a substitute to LDAP, which is also used to manage public catalogs. Prior to this 2.3 Tbps DDoS attack in February 2020, the second most significant DDoS attack was the 1.3 Tbps DDoS attack on GitHub, which included delivering 126.9 million traffic packages every second [1]. As a consequence, there is an imperative necessity to develop a DDoS attack detection system for efficient classification of incoming traffic into "attack" or "normal" categories.

Deep learning (DL) is a novel field of computer science that uses a sophisticated collection of feature embedding techniques to automatically learn from past data and accurately anticipate outcomes [6]. It has been successfully utilized in a variety of applications throughout the years, including stock market prediction [7], assessment of student performance [8], predictive modeling [9], and text categorization [10], and many others [5]. Data analysts are driven to build practical solutions that might assist network administrators to effectively forecast DDoS attacks in the field of DDoS attack detection [5]. As a result, for effective DDoS attack prediction, it is critical to explore and use state-of-the-art hybrid DL models on DDoS data.

1.2. The Goal of the Research

Several studies [1,11,12] examined the use of DDoS data to anticipate DDoS attacks using computational approaches, such as machine learning (ML). The primary focus of these investigations, however, was primarily on the earlier detection of DDoS attack outcomes. They were also restricted by (i) a poor selection of predictor factors utilized to characterize DDoS attacks and (ii) traditional classifiers that yielded poor results to address the connection between both the predictor variables in the DDoS data.

As a result, the CNN + BiLSTM hybrid DL model presented in this work employed better feature selection strategies. In the first step, a chi-squared (x^2) test was utilized to find appropriate predictors for DDoS attack predictions. In the following step, a convolutional neural network (CNN) and a bidirectional long/short-term memory (BiLSTM) were utilized to efficiently anticipate DDoS attacks from the supplied data set.

1.3. Problem Statements

The authors of [11] used benchmark data to create a feed-forward DL-based method for predicting DDoS assaults. It sought to anticipate DDoS attacks using a DL method known as a Feed Forward DL classifier. However, choosing efficient predictors before applying DL to big data sets may yield interesting results. As a result, standard DL classifiers may not provide an effective mechanism for predicting DDoS attacks based on benchmark data.

Utilizing benchmark data to predict future DDoS attacks is difficult for a variety of reasons, including inadequate predictor selection and the use of classical feature sets followed by machine learning classifiers [1,11]. Moreover, because of improper predictor variable selection and a lack of hybrid models, DL models are less effective when used in DDoS attack prediction.

To overcome these concerns, we treat predicting DDoS attacks using benchmark data as a binary-label class cation problem, in which the DDoS attack is predicted from a given data set. A feed of training data 'D = [d1, d2, d3, . . . dn] was loaded into a hybrid neural network to predict the DDoS attack, i.e., T1 (normal) or T2 (attack). Our goal is to develop an automated approach that learns from provided training data to anticipate DDoS attacks using a hybrid deep neural network model with optimized feature selection.

1.4. Our Proposal

To overcome the shortcomings of the baseline study [11], we present an effective hybrid DL model (CNN + BiLSTM) enhanced with the feature selection approach. The suggested deep learning approach has already been utilized effectively in a variety of applications, including intent detection [6], rumor categorization [10], and extremist association identification [13]. We built an x^2 test for feature selection then utilized a CNN + BiLSTM hybrid model for DDoS attacks classification. The following is the process: (i) the x^2 test was used to identify highly rated features that contribute considerably to predict court case judgments; (ii) a CNN was used to extract such high-rated features; and (iii) these features then are were into a BiLSTM model, which maintains the prior as well as future context of the provided data. In this way, the suggested technique may predict DDoS attack outcomes from data using both optimum feature selection and CNN and BILSTM layers.

1.5. Research Questions

Table 1 lists the research questions that were addressed in order to efficiently predict DDoS attacks.

Table 1. Research questions.

Research Question	Motivation
RQ1: How can the CNN + BiLSTM hybrid DL model predict DDoS attacks based on benchmark data set?	Investigate the suggested hybrid deep neural network model (CNN+ BiLSTM) and apply it to predict DDoS attacks using benchmark data.
RQ2. How can we compare the suggested CNN + BiLSTM model to traditional ML techniques?	Examine traditional feature representation-based ML approaches such as the random forests (RF), k-nearest neighbors algorithm (k-NN), logistic regression (LR), and support vector machine (SVM) as well as various evaluation metrics like as accuracy, precision, recall, and F1-score.
RQ3 How do we compare the suggested technique's effectiveness in predicting DDoS attacks using benchmark data to baseline studies and other DL approaches?	In contrast to other DL models (CNN, long short-term memory (LSTM), recurrent neural network (RNN), and BiLSTM), and state-of-the-art benchmark studies that focus word embedding-based feature map, investigate the efficacy of the suggested DL model (CNN + BiLSTM) which forecasts DDoS attacks using various evaluation performance measures such as precision, accuracy, recall, and F1-measure.

1.6. Research Contributions

This study makes the following research contributions:

This study includes the following research contributions: (i) the use of an x^2 test to rank and choose optimal features plays an important role in predicting DDoS attacks, (ii) a CNN + BiLSTM model was used to predict DDoS attacks, (iii) comparison of the performance of traditional ML classifiers with the suggested method for predicting DDoS attacks, (iv) DDoS attack prediction using two decision classes, (v) comparison of the suggested approach's performance with that of previous DL models and baseline research, and (vi) the suggested model greatly improved the DDoS attack prediction capability.

The remainder of this study is divided into the following sections: the literature study is discussed in Section 2, and the methodology of the suggested approach is described in Section 3. The results and discussion are presented in Section 4, and the conclusion and future scope for the suggested technique are presented in the last section.

2. Related Works

This section summarizes and evaluates existing research papers on detecting attacks using the different IDS techniques listed above.

Various machine learning algorithms have been used to identify DDoS attacks, mostly as classifiers. To mention a few, there are k-Nearest Neighbors (KNN), the Nave Bayes Classifier, support vector machines (SVM), random forest (RF), and neural networks (ANNs) [1,14] presented an interactive intelligent detection method for detecting DoS/DDoS attacks. The detection algorithm made use of the random forest tree technique to identify different DoS/DDoS attacks, including flood TCP, flood UDP, flood HTTP, and sluggish HTTP. However, Ref. [15] employed bio-inspired machine learning metrics to quickly and accurately identify HTTP flood attacks. The developers of [15] used the Bat algorithm, which is a low-complexity algorithm, as a bio-inspired method. While [16] presented a TCP flood DDoS detection methodology. Different ML classifiers, such as SVM, Nave Bayes, and KNN, were all used in this model. However, Ref. [17] demonstrated detection based on the covariance matrix method. The suggested detection was separated into training and testing steps. A training phase was designed to build a typical network traffic profile. The testing step was designed to detect any anomalous traffic by measuring the difference between usual and any other network activity. The regular traffic was recorded in their cloud from end-users surfing the Internet, whilst the flooding attack traffic was created using the PageRebooter application. It was analyzed using the confusion matrix and the findings were presented for a public and private cloud system. The authors of [18] utilized the NB technique to accurately anticipate the occurrence of DDoS attacks based on the mean and significance variance of packet headers. An RF is a grouping of decision trees. The categorization is determined by the proportion of the outputs of particular decision trees. To cope with the variety of network attacks, Dincalp employed the DBSCAN clustering approach [19]. In their trials, the suggested scheme performed well with key attributes. The authors of [20] examined the issue of intrusion detection and developed a two-stage ADOA approach to tackle it. To begin, the detected anomalies are grouped, and the unlabeled examples are sorted to obtain prospective anomalies and trustworthy normal occurrences by resolving the differences among anomalies. This is done by assigning labels to the aforementioned examples, and then creating a graded multi-class framework that may be used to differentiate distinct anomalies from the regular cases. The suggested method operates much better than all other approaches in the aforesaid setting, which demonstrates the effectiveness of the suggested strategy in this context. The authors of [21] suggested a VWHL-based technique for detecting industrial anomalies. Through the vertex weights in the hypergraph, the suggested technique was able to investigate the effect of different training samples on the detection task. The suggested approach enabled the investigation of data correlations, even when the amount of training data from distinct categories varied greatly. Experiments were undertaken to assess the proposed technique on the industrial anomaly detection dataset, the ODDS dataset, and the SDP dataset. The experimental findings indicated that the suggested technique outperformed current methods. These findings support the improved data representation provided by a vertices-directed graph. The authors of [22] described unique strategies for identifying the structures that support DDoS magnification attacks. To address this issue, a two-step process is used. To begin, they build a mechanism for imprinting scanning that performs surveillance for amplified attacks with fingerprints that enable us to trace future intrusions down to the scanners. With a level of certainty of greater than 99.9 percent, their technique assigns over 58 percent of attacks to scanners. Next, they correlate detectors to the real structures initiating the attacks using duration multilateration algorithms. They identified 34 networks as the source of amplifying attacks with 98 percent accuracy utilizing the current method.

DDoS detection has also proven to be a success story for DL. The authors of [23] suggested a machine learning-based technique for detecting attacks that leverages over 200 features collected from both static and dynamic analysis of Android applications. The analysis of the modeling results reveals that the deep learning approach is particularly well-suited for Android malware detection, with such a greater extent of 96 percent accuracy when applied to actual Android software collections. The authors of [24] suggested a deep learning-based malware detection technique for the Android version. To do this, they retrieved five distinct sorts of features from Android platform static analysis. Following this, a deep learning model is constructed to learn characteristics from Android applications. Lastly, an unexpected Android threat is detected using the learnt traits. Their system outperformed various current malware detection algorithms in a test with 3986 safe applications and 3986 malicious, achieving a 99.4 percent detection rate. Additionally, DroidDeep boasts an impressive run-time accuracy, making it extremely adaptable to a wider scale of actual Android malware detection. To identify DDoS attacks, Ref. [25] used a combination of LSTM and Bayesian techniques. The LSTM algorithm is well suited to activities with lengthy time frames and latencies. In other words, LSTM may determine whether or not it will save information for an unlimited period of time. The authors utilized LSTM to determine the confidence index of DDoS attacks and then applied the Bayesian framework to increase the detection performance. The authors of [26] constructed the neural network architecture using Long Short-Term Memory (LSTM), which is a particular structure of a recurrent neural network (RNN). Whereas [27] utilized recurrent neural networks using LSTM and gated recurring units (GRUs). It achieved the highest efficiency in the range of 90–95%. Bi-Directional Long Short-Term Memory (BI-LSTM), a Gaussian Mixture Model (GMM), and incremental learning are used in a novel DDoS detection system [2]. Unidentified traffic collected by the GMM is subjected to traffic analyst screening and tagging before being sent back into the system as new training examples. The experiment findings show that the suggested BI-LSTM-GMM can obtain recall, accuracy, and precision up to 94 percent using the data sets CIC-IDS2017 and CIC-DDoS2019 for training, testing, and validation. However, Ref. [11] proposed a deep learning model for detecting DDoS attacks on a collection of frames collected from network activity in this research. Since it incorporates the extraction of features and classification methods in its structure, as well as layers that upgrade themselves as it is learned, the DNN model can perform rapidly and accurately even with tiny data. Experimentation using the CICDDoS2019 dataset, which contains the most recent DDoS attack types produced in 2019, revealed that attacks on network activity were identified with 99.99 percent accuracy and attack categories were characterized with a 94.57 percent accuracy rate. The authors of [28] suggested a hybrid deep generative model that efficiently detects malware variations by combining global and local data. While its virus is transformed into an image to effectively describe global features using a pre-defined implicit space, it retrieves local features utilizing machine code sequencing. The two features retrieved from the dataset are synthesized and passed to the intrusion detection systems. By combining all these features, the suggested model obtains a 97.47 percent accuracy, which is considered to be state-of-the-art efficiency. The CAM findings indicate that the created malware enhances the detection accuracy.

Research gap: While machine learning/deep learning-based approaches have been successful, an important issue—choosing optimal features—has been unaddressed. The authors of [11] developed a feed-forward DL-based approach for forecasting DDoS attacks using benchmark data. It attempted to predict DDoS attacks by utilizing a DL technique known as a feed-forward DL classifier. However, the selection of effective predictors prior to applying deep learning to large data sets may produce encouraging outcomes. As a result, traditional deep learning classifiers may be ineffective in predicting DDoS attacks using benchmark data, if optimal sets of features are not selected. To address the limitations of the baseline study [11], we propose an efficient hybrid deep learning model (CNN+BiLSTM) augmented with feature selection. For the sake of being practically useful, this study addresses the problem of DDoS attack detection by adding optimum feature selection into the proposed hybrid DL-based architecture.

3. Methodology

Recent increases in the arrival rates of online data streams have placed a premium on the amount of resources required by data mining processing systems. DataStream Mining (i.e., stream learning) is a technique for extracting knowledge structures from an infinitely long and ordered series of data that occurs throughout time (data in the stream) [29].

Incremental learning is a term that refers to the process of acquiring information using stream data mining [30]. Both academics and industry have placed a premium on incremental learning. It is a form of machine learning in which previously learned information is applied when new examples come, and previously learned knowledge is updated in response to the new occurrences [6].

Using two class labels, the proposed system will deploy hybrid classifiers with improved FS. The traits associated with normal behavior are labeled "normal" or aberrant behavior is labeled "attack." Based on the suggestions in previous research [1,11], it was discovered that some classifiers produce superior detection results than others. To construct the prediction models, we chose hybrid deep learning classifiers with improved FS.

3.1. Benchmark Dataset

To conduct an assessment of our DDoS attack detection architecture, a dataset that accurately depicts such attacks is necessary. The assessments are conducted using the CICDDoS 2019 dataset [31]. The Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick (UNB) created the CICDoS 2019 dataset. The dataset contains 86 network traffic package attributes that have been generated using the open-source [32] tool, which produces network packets and collects attributes from them. DDoS attacks based on reflection utilize authorized servers, such as Domain Name Server (DNS), Lightweight Directory Access Protocol (LDAP), Network Basic Input/Output System (NETBIOS), and (Simple Network Management Protocol) SNMP, that render various services over the network. DDoS attacks relying on exploits, such as WebDDoS, SYN flood, UDP flood, and UDPLag, make use of vulnerabilities in the TCP and UDP communication protocols. The dataset has become helpful for the training of the model by eliminating extraneous attributes from the CICDDoS2019 dataset, which we chose for the detection and characterization of DDoS attacks. Packets using the TCP connection can be differentiated from the simplified UDP packets by the SYN, ACK, FIN, URG, PSH, RST, ECE, and CWR flag sections in the header elements. The network traffic of the first and second days is included in [31]. The excel spreadsheets were merged and utilized entirely during the investigation.

3.2. Pre-Processing

The initial step before training the deep learning models is always to preprocess the dataset in order to make it more appropriate for training and minimize overfitting. Preprocessing is accomplished in the following ways:

- The CICDDoS2019 dataset in csv file format, which we utilized in the investigations, was condensed to facilitate simpler training because it comprises a huge number of socket information like flowID, destination IP, scoure IP, etc. To conform to the suggested framework's numeric composition, the non-numeric elements were converted to numeric data using a one-shot encoding method.
- During importing of the dataset, the downsizing procedure was accomplished by omitting records at random times to ensure that the sample was randomized. The 'infinity' number was changed with '$-1'$', and the rows with 'NaN' entries were removed. The dataset was cleaned of nine attributes with a just '$0'$' value, and the model was provided with training with 69 attributes. The discarded 9 features include: Fwd Bulk Rate (Avg), Fwd URG (Flags), Bwd URG (Flags), Fwd Bytes/Bulk (Avg), Fwd Packet/Bulk (Avg), Bwd PSH (Flags), Bwd Bytes/Bulk (Avg), Bwd Packet/Bulk (Avg), and Bwd Bulk Rate (Avg).
- CICDDoS2019 class tags were categorized according to reflection- and exploitation-based attacks [31]. To identify an attack on network activity, the term 'BENIGN' was tagged with a value of '$0'$', whereas other attacks were marked with a value of '$1'$'. The normalization technique was used in the range of 0–1 numbers to ensure that the quantities in the dataset did not have an undue impact on training [11].

3.3. Feature Selection

Instead of choosing all features in the source data, we concentrate on identifying the appropriate attributes to forecast DDoS attacks. To choose the most optimum features from the raw data, numerous approaches, such as principal component analysis (PCA) [33], decision tree [34], Random Forest Regressor [35], and chi-squared (x^2) test [36], can be used. This study used an x^2 test to rank and choose features, as used by [36], and it yielded encouraging results.

An x^2 test analyzes whether the frequencies of particular classes and features are independent or reliant on the correlation among predictor and target variables. It was calculated in the following way:

$$Y_c^2 = \sum \frac{(Ai - Bi)^2}{Bi}$$

where c represents the degree of freedom, A represents the observed value, and B represents the anticipated observation in the ith class. On the original data set, the x^2 test was used to pick the most relevant features that had a strong relationship with the target variables. The Python-based Sklearn package was used to pick relevant features, which were then combined using the Select KBest score and the Chi^2 function, because the more optimum features have a greater correlation with the target attribute. To build our learning model, we selected a subset of 24 attributes from the original CICDDoS2019 dataset. Table 2 displays a partial list of the top 10 most important features, which were chosen based on their link to the attribute values. The highest-scoring features are ordered according to their rankings, which indicate a significant relationship and reliance on the target class.

Table 2. Optimal feature set.

F.No	Optimal Features	Rank Score
f1	Fwd Packet Length Max	6.41
f2	Fwd Packet Length Min	6.13
f3	Max Packet Length	5.21
f4	Min Packet Length	4.23
f5	Average Packet Size	3.87
f6	FWD Packets/s	3.52
f7	Fwd Header Length	3.36
f8	Fwd Header Length 1	3.31
f9	Min Seg Size Forward	2.72
f10	Fwd Packet Length std	2.33

3.4. Hybrid Deep Learning Model for DDoS Attack Classification

A convolutional neural network (CNN) with a bidirectional long short time memory (BiLSTM) model is proposed for detecting DDoS attacks. The system detects and categorizes traffic into two classes: “normal”, which is assigned to normal activity, and “anomaly”, which is assigned to aberrant behavior. The proposed work’s main structure (Figure 2) consisted of five components.

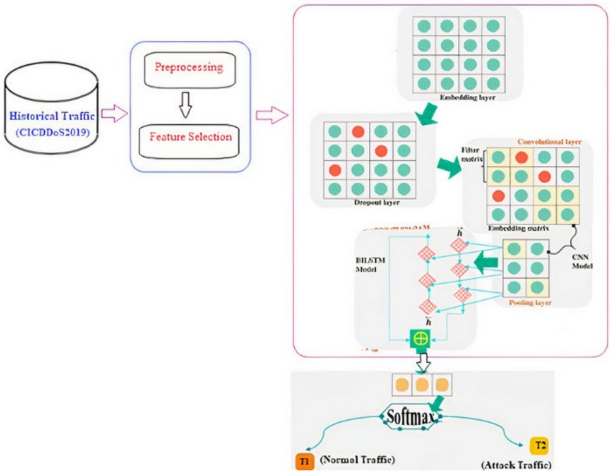


Figure 2. Proposed system for DDoS attacks classification.

3.4.1. Embedding Layer

The data embedding vector in this study was generated using a Keras embedding layer. A two-dimensional embedding matrix (feature matrix) was constructed within the embedding layer, as follows: $W \in \mathcal{R}^{t \times n}$. The length of the input data is denoted by ‘t’, and the dimension of a word embedding is denoted by ‘n’. Following the embedding matrix’s construction, it was transported to the next layer.

3.4.2. Dropout Layer-Based Overfitting Reduction Strategy

Dropout, which seeks to replace some activations in a neural network layer with zero, can also be used to decrease overfitting because dropouts and visible units in a neural network are both hidden [9]. The CNN model is made up of layers, such as the convolutional layer, which performs feature extraction, and the pooling layer, which reduces spatial size. The following is an overview of the CNN layers used in the neural network.

3.4.3. Convolutional Layer

At each data input D, a filter matrix $F \in \mathcal{R}^{q \times s}$ was put to a sliding window with a length of '*q*' during convolution. To obtain the feature space, the filter matrix was convolved across 'k' data chunks. Although the filter matrix was initially initialized at random, the values within it represent the weight of neurons (parameters) that were altered during individual training sessions. During convolution, the following feature map was created:

$$M_{ij} = a\left(F \otimes w_{i:i+q-1,\ j+s-1} + b\right) \tag{1}$$

In Equation (1), i ranges from 1 to (I + q − 1) whereas j ranges from 1 to (j + s − 1), b functions as a bias term, ⊗ represents the convolution operation in the embedding (input) matrix and filter matrix, and a represents the nonlinear activation function (ReLU).

In Equation (2), the adjusted feature map $E \in \mathcal{R}^{i+q-1}$ was generated as an outcome of the convolutional layer:

$$E = \left[e_{1,1}, e_{1,2}, e_{1,3}, \ldots, e_{i+q-1,j+s-1}\right] \tag{2}$$

3.4.4. Pooling Layer

This layer's major task was to conduct dimensionality reduction in each feature map. The overhead of calculation was lowered at the pooling layer while important information was preserved [6]. Equation (3) defines the max pooling operation mathematically:

$$L_{ij} = \max\left(w_{i+q-1,\ j+s-1}\right) \tag{3}$$

The maximum pooling process yielded a matrix $L \in \mathcal{R}^{u+q-1,\ z+s-1}$ of pooled feature maps, as shown in Equation (4).

$$L = \left[l_{1,1},\ l_{1,2}, l_{1,3}, \ldots, l_{u+q-1,\ z+s-1}\right] \tag{4}$$

3.4.5. BiLSTM-Based Context Information Extraction

Because both previous and upcoming contexts are equally important, Bi-LSTM models have lately received a lot of interest due to their improved capacity to keep this unique sequence of information [37]. This overcomes the limits of traditional RNNs, which can only remember information for a short amount of time, and unidirectional LSTMs, which could only maintain prior context [10]. As a result, this research implemented a BiLSTM model, which employs two distinct hidden layers to predict DDoS attacks based on historical traffic. BiLSTM is made up of two sub-networks: forward and backward pass LSTM [8]. BiLSTM computes the next (ahead) hidden vector "$\overrightarrow{h}$" and the prior (backwards) hidden vector "$\overleftarrow{h}$" given an input sequence of $x_1,\ x_2,\ x_3\ \ldots\ldots.,\ x_n$, of "$n$" words. The output sequence $h_1,\ h_2,\ h_3,\ \ldots\ldots,\ h_t$ is produced as an input to the output layer to forecast each data traffic [8] by recombining the right and left contextual depictions: $\overleftrightarrow{h} = \left[\overleftarrow{h}, \overrightarrow{h}\right]$.

3.4.6. SoftMax-Based Prediction Strategy

The final layer obtains the outcome of the BiSLTM layer as an input and employs SoftMax to determine the likelihood of correctly guessing the target labels (i.e., the court decisions). As shown in Equation (5), the cumulative input was computed:

$$t_i = \sum w_i l_i + b \tag{5}$$

where 'w' represents the weight vector, 'l' represents the input vector, and 'b' represents the bias. Equation (6) describes the SoftMax computation:

$$softmax\ (t_i) = \frac{exp^{t_i}}{\sum_{n=1}^{m} exp^{t_n}} \tag{6}$$

3.5. Applied Example

This section describes the various calculations that were used to forecast DDoS attacks based on the historical data provided. There is a proper explanation of the functions conducted by the suggested hybrid model.

The ultimate BiSLTM outcome is provided as input, and the SoftMax function is used to determine the probability of each tag: "t1", "t2", etc. Equation (6) was used to compute net input:

For DDoS normal/attack-1 (target variable 1 class label) = "t1"

The net input of the first class (t_1) of DDoS attacks is calculated as follows:

$$t_1 = l_1 \times w_2 + l_2 \times w_2 + b$$

$$t_1 = 0.2 \times 2.2 + 0.5 \times 1.8 + 0.8$$

$$t_1 = 0.44 + 0.9 + 0.8 = 2.14$$

The net input of the second classes of DDoS attacks is calculated in the same way:

$$t_2 = 0.921$$

The softmax activation function was used using Equation (7) to compute the probability of each label (t_1,t_2):

$$softmax\ (t_1) = \frac{exp^{t_1}}{exp^{t_1} + exp^{t_2}} \tag{7}$$

$$softmax\ (t_1) = \frac{exp^{2.14}}{exp^{2.14} + exp^{0.921}}$$

$$softmax\ (t_1) = \frac{8.499}{11.01} = 0.77$$

The SoftMax functions for the other DDoS attack/normal classes were derived in the same way:

$$softmax\ (t_2) = 2.512/11.011 = 0.23$$

The T1 DDoS traffic(normal) had the highest probability, according to this calculation. As a result, the projected DDoS attack decision was "A" based on the presented historical traffic data (Figure 3).

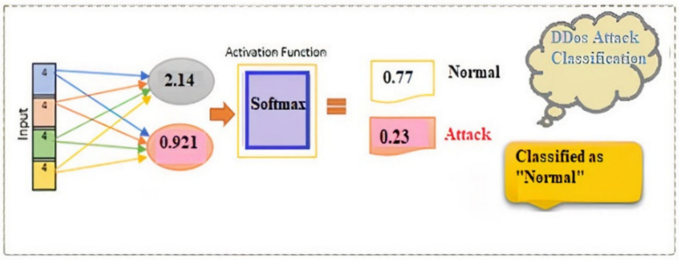


Figure 3. Classification using the softmax function.

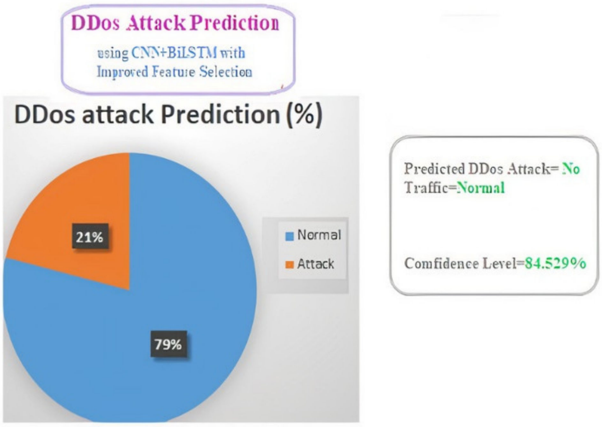
Algorithm 1 shows the pseudocode processes of the suggested model for predicting DDoS attacks.

Algorithm 1. Pseudocode of the proposed DDoS attack prediction model.

```
I.    Input: CICDDoS2019 (after preprocessing) labeled dataset D as csv file.
II.   Spilt into train ($S_{train}$, $NR_{train}$)-test ($S_{test}$, $NS_{test}$) using Scikit learn.
III.  Build the vocabulary to map integer to CICDDoS2019
IV.   Transform each CICDDoS2019 data stream into sequence of integers.
V.    Procedure CNN+ BiLSTM model ( $S_{train}$, $NS_{train}$)
# Initialization of Sequential function
Model=Sequential()
# using Embedding Layer to map integers to low dimensional vectors
Model.add(Embedding())

#Dropout Layer for preventing overfitting
Model.add(Dropout(0.5))

#—Applying Bi-LSTM layer for context information extraction
Model.add (Bidirectional (LSTM()))

#Prediction of DDoS Attacks using Softmax function
Model.add(Dense(3, activation='softmax')).

# Compilation Function
```

3.6. User Interface of the DDoS Attack Prediction Model

To predict DDoS attacks from provided benchmark data, a user-friendly online application was created. The DL model was trained using the Keras package and a Python-based Flask environment [10]. With a projected confidence rate, the model forecasts either T1 (normal) or T2 (attack). Figure 4 depicts the predicted DDoS attack for the supplied data, which was T1 (normal).



Predicted DDos Attack= No
Traffic=Normal

Confidence Level=84.529%

Figure 4. DDoS attack prediction output.

4. Experiments and Their Outcomes

To assess the efficiency and effectiveness of the suggested framework, experiments were carried out. The experiments were performed on an Intel Core i7-7700 CPU with 32 GB RAM, using Python on the TensorFlow 2.0 and Keras platforms.

4.1. Answer to First RQ

RQ1: "*How can the CNN + BiLSTM hybrid DL model predict DDoS attacks based on benchmark data set?*" can be answered by applying different parameters to the DL model and utilizing various CNN + BiLSTM models to forecast DDoS attacks using benchmark data. We performed some tests using various parameters. The parameters for the proposed CNN-BiLSTM model are given in Table 3.

Table 3. CNN-BiLSTM model parameterization.

Parameter	Value	Parameter	Value
vocabulary size	1000	Emdedding dimension	128
Input vector size	53 to 60	Unit size of BiLSTM	65, 45, 40, 35, 20, 15, 10
# of convolutional layers	1	# of hidden layers	4
# of filters	6, 9, 10, 16	size of filters	7, 8, 10
Dropout	0.9	Activation function	Softmax
# of epochs	7	size of batch	8, 16

Table 4 shows the parameter settings for 10 CNN-BiLSTM variants.

Table 4. Parameterization for 10 different CNN + BiLSTM models.

Model Name	# of Filters	Unit Size of BiLSTM	Size of Filter	Model Name	# of Filters	Unit Size of BiLSTM	Size of Filter
CNN-BiLSTM (1)	6	20	7	CNN-BiLSTM (2)	6	45	7
CNN-BiLSTM (3)	6	40	8	CNN-BiLSTM (4)	10	20	8
CNN-BiLSTM (5)	10	40	10	CNN-BiLSTM (6)	10	35	10
CNN-BiLSTM (7)	10	60	10	CNN-BiLSTM (8)	10	45	10
CNN-BiLSTM (9)	9	20	10	CNN-BiLSTM (10)	16	10	8

In accordance with Table 5, it is noticed that the CNN-BiLSTM model "CNN-BiLSTM (10)" with a filter size of 8 × 8, a filter count of 16, and a BiLSTM unit size of 10 (neurons) outperforms all other models with a 76 percent accuracy. The various models are listed in ascending order of their test accuracy, which ranges from 85% to 94%. After completing several tests on various CNN-BiLSTM models with variable parameters, we recorded the test accuracy, test loss, and training time. Table 5 summarizes the training duration, test accuracy, and test loss for all 10 trials using various parameter values in the CNN + BiLSTM model.

Table 5. The tested CNN + BiLSTM models' test accuracy, test loss, and training time.

Model Name	Test Accuracy	Test Loss	Unit Size (BiLSTM)	Training Time (s)
CNN + BiLSTM-1	85.11%	0.81	65	7 s
CNN + BiLSTM-2	86.01%	0.86	45	13 s
CNN + BiLSTM-3	86.52%	1.06	40	21 s
CNN + BiLSTM-4	87.46%	1.16	40	10 s
CNN + BiLSTM-5	87.77%	0.95	35	11 s
CNN + BiLSTM-6	91.46%	0.93	35	6 s
CNN + BiLSTM-7	92.00%	1.13	20	13 s
CNN + BiLSTM-8	92.05%	0.82	15	10 s
CNN + BiLSTM-9	93.10%	0.92	10	13 s
CNN + BiLSTM-10	94.52%	0.81	10	29 s

By varying the parameters of the DL model, we discovered that reducing the unit size of the BiLSTM model leads to increased accuracy. In other words, the CNN + BiLSTM-10 model with feature selection performed best (91.52 percent) with smaller unit sizes.

During testing, it was discovered that the CNN-BiLSTM (10) model, which had a total of 16 filters, an average filter size of 8, and a BiLSTM unit size of 10 (neurons), outperformed all other models by 76 percent. The training time of the model is enhanced by reducing the filter size.

4.2. Performance Measures

The confusion matrix is utilized in this section to construct the model's learning requirements. False positive (FP), false negative (FN), true positive (TP), and true negative (TN) are the constituents of the confusion matrix (FN). We present the confusion matrix to demonstrate our model's classification efficiency. The confusion matrix highlights which predictions were right and which were incorrect. According to Table 6, our model has a detection rate of 0.95 for all attack and normal categories.

Table 6. Confusion matrix based on our suggested technique for four unique occurrences (TP, FP, TN, and FN).

Actual **Predicted**	**Attack**	**Normal**
Attack	0.95	0.05
Normal	0.05	0.95

Additionally, we assess our proposed model using the different metrics that are widely utilized in intrusion detection systems. The mathematical equations of precision, recall, and f-score are presented below (Equations (8)–(10)).

Accuracy: Equation (8)'s accuracy reveals the model's accurate predictive performance. Accuracy is a measure that quantifies the total percent of detected and erroneous alarms generated by an IDS model; it represents the total rate of success of any IDS and is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$TP = true\ positive,\ TN = true\ negative,\ FP = false\ positive,\ and\ FN = false\ negative$$

Precision: the false negative rate (FNR), sometimes referred to as precision, is the proportion of miscategorized attacks to the overall number of attack occurrences. The precision produced from Equation (9) indicates how many positive forecasts are predicted exactly:

$$Precision(p) = \frac{TP}{FP + TP} \tag{9}$$

$$p = precision,\ TP = true\ positive,\ FP = false\ postive,\ and\ FN = false\ negative$$

Recall: the detection rate (DR), also known as the true positive rate (TPR) or recall, is the percentage of properly identified malicious occurrences in relation to the total number of malicious vectors. Equation (10), which calculates recall, reveals how many true positives are successfully forecasted:

$$Recall(r) = \frac{TP}{FN + TP} \tag{10}$$

$$r = recall,\ TP = true\ positive,\ and\ FN = false\ negative$$

F-score: The F1 score is critical since it provides further information about the IDS's performance. It takes into account false positives and negatives. The F1 score is advantageous in particular when the distribution of class labels is unequal or unbalanced. The F-score, which can be calculated using Equation (11), demonstrates the consistency of recall and sensitivity:

$$F_{score} = 2x\frac{P \text{ x } R}{P + R} \tag{11}$$

$$\text{R} = \text{Recall},\ \text{P} = \text{Precision}$$

Table 7 summarizes the accuracy, recall, and F1-score of the different CNN + BiLSTM models.

Table 7. Evaluation of the performance of CNN-BiLSTM models with and without feature selection [FS(no) = without selection of features, FS(yes) indicates with selection of features].

Model Name	Accuracy (%)		Precision (%)		Recall (%)		F1-Score (%)	
	FS(No)	**FS(Yes)**	**FS(No)**	**FS(Yes)**	**FS(No)**	**FS(Yes)**	**FS(No)**	**FS(Yes)**
CNN+ BiLSTM-1	73	85.11	75	85	74	85	74	85
CNN+ BiLSTM-2	76	86.01	78	86	76	86	74	86
CNN+ BiLSTM-3	69	86.52	71	86	70	86	68	86
CNN+ BiLSTM-4	64	87.56	69	87	64	87	62	87
CNN+ BiLSTM-5	73	87.77	76	87	72	87	73	87
CNN+ BiLSTM-6	72	91.46	77	91	73	89	74	90
CNN+ BiLSTM-7	63	92.00	69	91.71	66	91	66	90.71
CNN+ BiLSTM-8	77	92.05	81	92.47	77	91.31	77	91.16
CNN+ BiLSTM-9	71	93.10	74	93.41	71	91.92	72	92.62
CNN+ BiLSTM-10	79	94.52	80	94.74	79	92.04	79	93.44

Table 7 summarizes the accuracy, recall, and F1-score of the different CNN + BiLSTM models, with and without feature selection. The best accuracy of 94.52 percent was attained by our suggested model CNN-BiLSTM (10).

4.3. Answer to Second RQ

We assess the proposed CNN + BiLSTM model's performance in predicting DDoS attacks using benchmark data by comparing it to conventional machine learning models. The suggested word embedding-driven DL model outperforms traditional machine learning methods. Table 8 shows the results of the ML models as well as the suggested model.

Table 8. Comparison of the proposed model to ML models.

	Study/Technique	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F-Score (%)**
machine learning	Sambangi and Gondi, [1] (Multiple Linear Regression)	78	79	78	78
	XGB	76	76	76	76
	SVM	74	74	74	74
	Random Forest	75	75	75	75
	LR	64	64	64	64
	KNN	71	71	71	71
Proposed Deep learning model (without FS)	CNN + BiLSTM	83	84	83	83
Proposed Deep learning model (with FS)	CNN + BiLSTM	94.52	94.74	92.04	93.44

Accuracy, precision, recall, and the f-measue are all used to assess performance. Multiple linear regression has the best incremental learning accuracy, which is roughly 78 percent on both localhost and distant virtual hosts. However, K-neighbors had the best accuracy of 71%. Each experiment's details can be found in Table 8. The SVM algorithm produced an efficient accuracy of 74% for cloud testing.

- CNN + BiLSTM vs. Multiple Linear Regression: The purpose of this experiment was to evaluate the efficacy of the proposed CNN + BiLSTM model with research by [1], which utilized a Multiple Linear Regression classifier to predict DDoS attacks using historical traffic data. In terms of precision (78), recall (79), F1-score (78), and accuracy (78), Multiple Linear Regression classifiers provided inferior results (Table 8). The Multiple Linear Regression model's poor performance might be attributable to a variety of factors as identified by [1].
- CNN + BiLSTM vs. XGBoost: The objectives of this investigation was to evaluate the suggested CNN + BiLSTM model against an extreme gradient boosting (XGBoost) classifier. As shown in Table 8, XGBoost classifiers yielded lower precision (76), recall (76), F1-score (76), and accuracy (76 percent). XGBoost receives a poor score because it is susceptible to overfitting in the presence of noisy data, needs a longer training period, and is difficult to tweak [38].
- SVM vs. CNN + BiLSTM: The objectives of this investigation were to evaluate the efficiency of the suggested CNN + BiLSTM model to that of SVM classifier to predict DDoS attacks using historical traffic. SVM classifiers performed worse in terms of precision (74), recall (74), F1-score (74), and accuracy (74), as seen in Table 8. The SVM model's poor performance might be attributed to: (i) long training times, (ii) expensive computation, (iii) increased size requirements for training and testing, and (iv) more complexity [9].
- CNN + BiLSTM vs. Random Forest: the purpose of this experiment was to see how well the suggested CNN + BiLSTM model compared to a random forests (RF) classifier. Table 8 demonstrates that RF classifiers have lower precision (75), recall (75), F1-score (75), and accuracy (75) than the proposed system. The RF model's poor performance is due to the following factors: (i) its legitimate predictions takes time, (ii) it is unreliable for categorical attributes, and (iii) comparable sets of related attributes in the data are favored over bigger sets [39].
- CNN + BiLSTM vs. Logistic Regression: the objective of this experiment was to evaluate the suggested CNN + BiLSTM model against a logistic regression (LR) classifier. As shown in Table 8, LR classifiers provided lower precision (64), recall (64), F1-score (64), and accuracy (64) outcomes (0.64 percent). LR is rated poor because it is prone to overfitting [39] and only makes relatively brief predictions [39].
- CNN + BiLSTM vs. KNN: The objective of this investigation was to evaluate the suggested CNN + BiLSTM model against a k-nearest neighbors (KNN) classifier. Table 8 demonstrates that KNN classifiers have lower precision (71), recall (71), F1-score (71), and accuracy (71). KNN is a low-ranking algorithm because it is (i) time-consuming when working with big data sets, and (ii) sensitive to irrelevant and noisy data [39].

4.4. Answer to Third RQ

To assess the CNN + BiLSTM model's effectiveness in predicting DDoS attacks from historical traffic, it was compared to various deep learning (DL) models, including CNN, long/short-term memory (LSTM), gated recurrent unit (GRU), recurrent neural network (RNN), and BiLSTM. Table 9 summarizes the findings.

Table 9. Comparison of the proposed effectiveness of the algorithm to that of DL models.

	Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Different models of deep learning	CNN	75	74	75	73
	LSTM	72	76	72	71
	BiLSTM	73	75	73	71
	CNN + LSTM	75	76	75	75
	RNN	79	83	79	79
	CNN + RNN	70	76	70	69
Proposed model (without FS)	Proposed CNN + BiLSTM	83	84	83	83
Proposed model (without FS)	Proposed CNN + BiLSTM	94.52	94.74	92.04	93.44

- CNN + BiLSTM vs. CNN: the purpose of this experiment was to evaluate the suggested CNN + BiLSTM model against a single-layer CNN model in terms of effectiveness. As shown in Table 9, the CNN model demonstrated lower precision, recall, F1-score, and accuracy. CNN is ranked low since it (i) lacks information about the text's sequence context and (ii) needs a large data set to give an enhanced classification performance.
- CNN + BiLSTM vs. LSTM: this investigation compared the suggested CNN + BiLSTM model's effectiveness against that of an LSTM model. As shown in Table 9, the CNN model demonstrated lower precision, recall, F1-score, and accuracy. LSTM models retain only previous contextual knowledge and discard upcoming contextual information, which would aid in comprehending the meaning of the reviewed text. As such, it performed the worst of all the models.
- CNN + BiLSTM vs. BiLSTM: this investigation compared the proposed CNN + BiLSTM model against a BiLSTM model in terms of predicting court judgments from previous legal data. As shown in Table 9, the BiLSTM model performed worse in terms of precision, recall, F1-score, and accuracy. A BiLSTM model's principal aim is to store contextual information for both forward and reverse directions in a sequence. BiLSTM is ranked low due to its ineffectiveness in extracting features.
- CNN + BiLSTM vs. CNN + LSTM: the purpose of this experiment was to evaluate the suggested CNN + BiLSTM model against a CNN+LSTM model. The CNN + LSTM model underperformed in terms of precision, recall, F1-score, and accuracy, as shown in Table 9. This is because the unidirectional LSTM layer is ineffective at retaining subsequent contextual information, leading to suboptimal efficiency.
- CNN + BiLSTM vs. RNN: the aim of this experiment was to evaluate the suggested CNN + BiLSTM model against an RNN model in terms of effectiveness. As shown in Table 9, the RNN model achieved suboptimal performance in terms of precision, recall, F1-score, and accuracy. Due to the fact that RNN models are unable to manage exceptionally long-term sequencing, they would not retain information for an extended length of time. As a consequence, the RNN model produced suboptimal outcomes.
- CNN + BiLSTM vs. CNN + RNN: the purpose of this experiment was to contrast the suggested CNN + BiLSTM model against a CNN+RNN model. As seen in Table 9, the CNN + RNN model performed poorly in terms of precision, recall, F1-score, and accuracy. This is because RNN models do not retain context data over extended periods of time.

The above comparison tests show that the suggested CNN + BiLSTM model outperforms other deep learning models (LSTM, GRU, CNN, BiLSTM, and RNN) in terms of precision, recall, F1-score, and accuracy. This resulted in an increase in classification accuracy when two deep learning models, namely CNN and BiLSTM, were combined along with feature selection.

Explanation for better outcomes: our work proposes combining a BiLSTM with a CNN model. The capacity of BiLSTM models to store two-directional context data efficiently—forward (next) and backward (previous)—is the chief factor for the suggested model's enhanced performance. Its improved representation of data (input text), gained via the CNN model, allows it to gather information not only from the current input but also from previous ones, preventing information decay. This results in effective court decision prediction using past legal data, since the BiLSTM model maintains both present and previous context data, whereas the CNN model extracts just localized features. Due to the improved representation of the input text, this results in high classification results. This study makes a unique contribution by demonstrating the capability of hybrid DL with effective feature selection for anomaly detection methods. To classify incoming traffic into legitimate or malicious categories, we presented a hybrid DL method based on CNN-BiLSTM. In comparison to shallow learners, our method outperformed them in terms of accuracy, recall, F1-measure, and precision. Because of their capacity to cope with a large degree of complicated nonlinear relationships, DL methods hold promise for accurately detecting intrusions. It may be used to overcome the limitations of conventional classification approaches, which rely on classical feature encoding to detect anomalous traffic [5].

4.5. Cross Validation

The subsequent experiment is to randomize the dataset using the k-fold validation approach in order to assess the proposed method's efficiency. The dataset is split into k equal-sized subgroups. A subset of 10-fold is used in this study.

When K = 10, the data is split into 10-folds that are about the same magnitude for every fold, resulting in a total of 10 data subsets for each fold. The cross validation test is performed on each of the 10 data subsets using 9-fold for training and 1-fold for testing, as seen in Table 10.

Table 10. A sample listing of randomized 10-fold cross validation.

K-Fold (K = 10)	**D = Dataset (1–500)**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
K1	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500
K2	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500
K3	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500
K4	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500
K5	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500
K6	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500
K7	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500
K8	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500
K9	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500
K10	1–50	51–100	101–150	151–200	201–250	251–300	301–350	351–400	401–450	451–500

To evaluate a number of classifiers, we used 10-fold cross validation. Table 11 shows the mean accuracy, standard deviation of mean, mean precision marco, standard deviation of precision marco, mean recall marco, standard recall marco, and mean f-1 marco data. It is observed that the proposed model (CNN+BiLSTM) gives the best results.

Table 11. Cross validation of the proposed system contrasted with different classifiers.

Classifiers	Mean Accuracy	Standard Deviation	Mean Precision Macro	Standard Deviation	Mean Recall Macro	Standard Deviation	Mean F-I Macro	Standard Deviation
Random Forest	82	0.06	83	0.05	83	0.06	83	0.07
SVM	72	0.06	72	0.06	72	0.06	72	0.06
XGBoost	84	0.07	83	0.06	83	0.06	83	0.07
CNN	86	0.07	87	0.05	86	0.05	83	0.06
BiLSTM	85	0.06	85	0.05	83	0.05	84	0.05
CNN + BILSTM (proposed)	93.11	0.05	91	0.04	91	0.05	92	0.05

4.6. Significance Testing

To determine the study's significance, two experiments were undertaken (see Table 12). If the CNN + BiLSTM (DL) model outperformed the SVM(ML) model statistically, it did not occur randomly.

Table 12. Significant differences between the SVM (ML) and CNN + BILSTM (DL) models.

	Correct Classification with SVM	Misclassification with SVM	Total
Correct classification CNN + BILSTM	60	4	64
Misclassification with CNN + BILSTM	14	22	36
Total	74	26	100

With a two-tailed p value of 0.135 and a degree of freedom of 1, the value of chi-squared is 2.2. The alternative hypothesis is accepted, whereas the null hypothesis is rejected (both models are statistically significant).

After randomly choosing 100 records from the dataset, the individual records were classified using the CNN + BILSTM (DL) and KNN classifiers (ML). Two hypotheses were tested in the experimental context:

H_{null}: The error rates across both models are identical.

$H_{alternate}$: The error rates of both models are significantly dissimilar.

Equation (12) gives McNemar's chi-squared statistic test:

$$\chi^2 = \frac{(|x-y|-1)^2}{(x+y)} \tag{12}$$

The cells x and y were used to generate discordant test statistics, with 1 representing the degree of freedom and χ^2 representing chi-squared.

Analysis

Table 9 demonstrates the CNN + BILSTM model's utility, demonstrating a significant improvement in predicting DDoS attacks from historical traffic data, with an accuracy of 94.52 percent. The utility of the SVM model is seen in Table 8, since it scored poorly on all estimation metrics: precision, recall, F1-score, and accuracy. According to the significance test, the disparity between both the DL model (CNN+BiSLTM) and the ML model was substantial (SVM). Using continuous correction, the p-value for McNemar's statistic test was computed. The Chi-squared coefficient was 2.2, with one degree of freedom and a p-value of 0.135 for two-tailed analysis. A p-value less than 0.5 validated the alternative hypothesis and disproved the null hypothesis. As a consequence, the suggested model with word embedding outperformed the SVM model based on conventional features by a statistically significant margin. This demonstrates how the addition of word representation characteristics enhanced the CNN + BILSTM model's resistance against DDoS assaults using historical traffic data.

4.7. Comparing the Proposed System to Existing Systems and Qualitative Evaluation

This section compares the proposed approach to benchmark studies. It is challenging to do a real comparison of the stated procedures due to a variety of constraints. For example, such algorithms are evaluated on a variety of datasets, making comparison complicated. Furthermore, the participating researchers offer the methodologies in their studies at an abstract level with little information, which could render them unfeasible for future investigations.

Bearing the aforesaid difficulties in mind, we implemented the strategies outlined in the published works using two datasets. During implementation, we did our best to adhere to the original experiment and procedure described in the papers; nevertheless, owing to inadequate discussions and an absence of adequate information in certain cases, we had to remove such aspects of the technique or presume what the researchers wanted. For example, using historical traffic, Ref. [1] suggested a supervised ML model for predicting DDoS attacks. On historical traffic, an ML algorithm called Multiple Linear Regression was used. The model's performance is poor, as evidenced by the experimental results (accuracy: 75%, precision: 75%, recall: 75%, and F1-score: 75%), obtained on the given benchmark dataset. However, they did not indicate specifications on system parameters and feature engineering, which may differ from the authors.

We conducted a quantitative evaluation of the various DDoS attacks detection algorithms using two cutting-edge datasets obtained from [27]. We used an Anaconda-based Jupyter notebook to apply known methodologies [12]. The findings from our tests diverge from the stated results in a few instances owing to the use of various datasets, parameterization, and software. For example, although Sambangi and Gondi [1] claimed 85 percent accuracy, we achieved 75 percent; Refs. [9,12] reported 84 percent accuracy, while our studies generated 74 percent on the CICDDoS2019 dataset and 77 percent on the CIC-IDS2017 dataset. The variations in the claimed and tested accuracies were induced by the authors' utilization of multiple datasets. In the study on DDoS traffic data, Ref. [12] used the DL method. It was discovered that combining better feature selection approaches with a DL model might increase the model's efficacy. In another work, Ref. [11] proposed a supervised DL model for predicting DDoS attacks based on historical traffic. A deep learning method called the feed forward model was employed to analyze historical traffic. The model's performance on the provided benchmark dataset demonstrated low results in the absence of an optimal set of features. In our implementation, a hybrid deep learning model, especially CNN + BiLSTM with an improved FS, outperformed earlier methodologies, and it is recommended that more research into various combinations of deep learning models for predicting DDoS attacks will yield more remarkable results. The suggested DL-based solution for predicting DDoS attacks was based on a hybrid deep neural network model and an enhanced feature selection strategy. The experimental findings show that the suggested approach outperforms baseline research (Table 13), and that the selected predictor factors (10) have a substantial impact on the projected (target) variable.

Table 13. The suggested model and baseline results are compared (A: Accuracy, P: Precision, R: Recall, and F: F1.

Study and Technique	Performance in Our Experiments A (Accuracy), P (Precision), R (Recall)		Performance (Reported)
	CICDDoS2019 Dataset	CIC-IDS2017 Dataset	
Sambangi and Gondi [1] Machine Learning (Multiple Linear Regression)	0.75 (A), 0.75 (P), 0.75 (R), 0.75 (F)	0.78 (A), 0.78 (P), 0.78 (R), 0.78 (F)	0.85 (A), 0.85 (P), 0.85 (R), 0.85 (F)
Cheng et al. [12] DL Classifier (M.S-CNN)	0.74 (A), 0.74 (P), 0.74 (R), 0.74 (F)	0.77 (A), 0.77 (P), 0.77 (R), 0.77 (F)	0.84 (A), 0.84 (P), 0.84 (R), 0.84 (F)
Cil et al. [11] DL Classifier (Feed Forward DL)	0.79 (A), 0.78 (P), 0.79 (R), 0.78 (F)	0.82 (A), 0.81 (P), 0.82 (R), 0.81 (F)	0.89 (A), 0.88 (P), 0.89 (R), 0.88 (F)
Proposed system (CNN + BILSTM with improved FS)	94.52 (A), 94.74 (P), 92.04 (R), 93.44 (F)	93.22 (A), 93.54 (P), 91.01 (R), 92.14 (F)	N/A

5. Conclusions and Future Work

In this study, we presented a novel model based on DL for detecting DDoS attacks on SDN networks.

The goal of this research was to recognize and classify DDoS attacks using a CNN + BiLSTM hybrid deep neural network model. The following are the proposed elements for the proposed system (CNN + BiLSTM): (i) dataset acquisition, (ii) preprocessing, (iii) feature selection, and (iv) classification. We trained and evaluated the proposed model using the newly available CICDDoS2019 dataset. The collection includes the first most relevant and recent DDoS categories of attacks. Numerous further tests were performed using the data set. In the provided data collection, feature selection was used to choose just the most relevant features by ranking and choosing the top-ranking features. Finally, a CNN + BiLSTM hybrid model was used to predict DDoS attacks: *normal, or attack.* In comparison to current well-known traditional ML approaches, the assessment of our model revealed that the suggested system generates the best evaluation metrics in terms of recall, precision, F-score, and accuracy. When contrasted to the baseline techniques, the research findings are encouraging in terms of increased accuracy (94.52%), precision (94.74%), recall (92.04%), and f-score (93.44%).

5.1. Limitations

However, significant drawbacks of the proposed system include:

(i) The use of a single data set;
(ii) The use of a single statistical technique, the chi-squared measure, to identify important features (predictors);
(iii) The use of embeddings rather than a pre-trained CNN model; and
(iv) In this paper, we employed a binary classification system to categorize input traffic as normal or malicious.

5.2. Future Work

(i) To examine the use of various traffic data sets. We want to evaluate the performance of our suggested model on more datasets in the future.
(ii) To investigate alternative feature selection approaches to the chi-squared assessment, and pre-trained word embedding algorithms, such as autoencoders, Glove, or Fasttext.
(iii) However, each attack class must be classified independently. We plan to expand our research to include a multi-class categorization system.
(iv) Furthermore, we will emulate the SDN network with various sorts of settings and attack traffics in order to generate a heterogeneous dataset that accurately represents actual internet traffic.

(v) Additionally, by combining the CICDDoS2019 dataset with hybrid DL, we can give direction to other academics focusing on DDoS vulnerability scanning. When it comes to detecting intrusions and securing software-based networks, it appears that the hybrid DL model with improved FS is an excellent choice due to improved accuracy.

(vi) We intend to create a dataset similar to the CICDDoS2019 dataset in the future by capturing network activity through virtual computers and Internet of things devices. By including DNN and deep learning models in the dataset that will be created, it will be possible to identify real-time DDoS attacks and plan appropriate responses.

Author Contributions: Conceptualization, D.A. and O.B.; methodology, D.A., O.B; software, H.U. and M.Z.A.; validation, M.Z.A., and H.U.; formal analysis, D.A.; investigation, O.B.; resources, D.A., OB, data curation, O.B., and H.U; writing—M.Z.A., D.A. original draft preparation, M.Z.A., D.A; writing—D.A., H.U.; visualization, O.B., M.Z.A.; supervision, D.A.; project administration, D.A.; funding acquisition, D.A. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under Grant No. (RG-95-611-42). The authors, therefore, acknowledge with thanks to DSR technical and financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data underlying this work can be requested from the corresponding author.

Acknowledgments: This project was funded by the Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, under Grant No. (RG-95-611-42). The authors, therefore, acknowledge with thanks to DSR technical and financial support.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Sambangi, S.; Gondi, L. A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression. *Proceedings* **2020**, *63*, 51. [CrossRef]
2. Shieh, C.S.; Lin, W.W.; Nguyen, T.T.; Chen, C.H.; Horng, M.F.; Miu, D. Detection of Unknown DDoS Attacks with Deep Learning and Gaussian Mixture Model. *Appl. Sci.* **2021**, *11*, 5213. [CrossRef]
3. Genie-Networks. DDoS Attack Statistics and Trends Report for 2020. 2021. Available online: https://www.genie-networks.com/gnnews/DDoS-attack-statistics-and-trends-report-for-h1-2020/ (accessed on 6 May 2021).
4. Jonker, M.; Sperotto, A.; Pras, A. DDoS Mitigation: A measurement-based approach. In Proceedings of the NOMS 2020—2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–6.
5. Alsaeedi, A.; Bamasag, O.; Munshi, A. Real-Time DDoS flood Attack Monitoring and Detection (RT-AMD) Model for Cloud Computing. In Proceedings of the 4th International Conference on Future Networks and Distributed Systems (ICFNDS), Saint Petersburg, Russia, 26–27 November 2020; pp. 1–5.
6. Khattak, A.; Asghar, M.Z.; Ali, M.; Batool, U. An efficient deep learning technique for facial emotion recognition. *Multimed. Tools Appl.* **2021**. [CrossRef]
7. Khattak, A.; Khan, A.; Ullah, H.; Asghar, M.U.; Arif, A.; Kundi, F.M.; Asghar, M.Z. An Efficient Supervised Machine Learning Technique for Forecasting Stock Market Trends. In *Information and Knowledge in Internet of Things*; Springer: Cham, Switzerland, 2022; pp. 143–162.
8. Zubair Asghar, M.; Subhan, F.; Imran, M.; Masud Kundi, F.; Khan, A.; Shamshirband, S.; Mosavi, A.; Varkonyi-Koczy, A.R.; Csiba, P. Performance evaluation of supervised machine learning techniques for efficient detection of emotions from online content. *Comput. Mater. Contin.* **2020**, *63*, 1093–1118. [CrossRef]
9. Khan, A.; Khattak, A.M.; Asghar, M.Z.; Naeem, M.; Din, A.U. Playing First-Person Perspective Games with Deep Reinforcement Learning Using the State-of-the-Art Game-AI Research Platforms. In *Deep Learning for Unmanned Systems*; Springer: Cham, Switzerland, 2021; pp. 635–667.
10. Ahmad, S.; Asghar, M.Z.; Alotaibi, F.M.; Khan, S. Classification of poetry text into the emotional states using deep learning technique. *IEEE Access* **2020**, *8*, 73865–73878. [CrossRef]
11. Cil, A.E.; Yildiz, K.; Buldu, A. Detection of DDoS attacks with feed forward based deep neural network model. *Expert Syst. Appl.* **2021**, *169*, 114520. [CrossRef]

12. Cheng, J.; Liu, Y.; Tang, X.; Sheng, S.V.; Li, M.; Li, J. DDoS attack detection via multi-scale convolutional neural network. *Comput. Mater. Contin.* **2020**, *62*, 1317–1333. [CrossRef]
13. Ahmad, S.; Asghar, M.Z.; Alotaibi, F.M.; Awan, I. Detection and classification of social media-based extremist affiliations using sentiment analysis techniques. *Hum. Centr. Comput. Inf. Sci.* **2019**, *9*, 1–23.
14. Lima Filho, F.S.D.; Silveira, F.A.; de Medeiros Brito, A., Jr.; Vargas-Solar, G.; Silveira, L.F. Smart detection: An online approach for DoS/DDoS attack detection using machine learning. *Secur. Commun. Netw.* **2019**, *2019*, 1574749. [CrossRef]
15. Sreeram, I.; Vuppala, V.P.K. HTTP flood attack detection in application layer using machine learning metrics and bio inspired bat algorithm. *Appl. Comput. Inform.* **2019**, *15*, 59–66. [CrossRef]
16. Sahi, A.; Lai, D.; Li, Y.; Diykh, M. An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. *IEEE Access* **2017**, *5*, 6036–6048. [CrossRef]
17. Aborujilah, A.; Musa, S. Cloud-based DDoS HTTP attack detection using covariance matrix approach. *J. Comput. Netw. Commun.* **2017**, *2017*, 7674594. [CrossRef]
18. Fadlil, A.; Riadi, I.; Aji, S. Review of detection DDOS attack detection using naive bayes classifier for network forensics. *Bull. Electr. Eng. Inform.* **2017**, *6*, 140–148. [CrossRef]
19. Dincalp, U.; Güzel, M.S.; Sevine, O.; Bostanci, E.; Askerzade, I. Anomaly based distributed denial of service attack detection and prevention with machine learning. In Proceedings of the 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 19–21 October 2018; pp. 1–4.
20. Zhang, Y.L.; Li, L.; Zhou, J.; Li, X.; Zhou, Z.H. Anomaly detection with partially observed anomalies. In *Proceedings of the Companion Proceedings of the Web Conference*; Lyon, France, 23–27 April 2018, pp. 639–646.
21. Wang, N.; Zhang, Z.; Zhao, X.; Miao, Q.; Ji, R.; Gao, Y. Exploring high-order correlations for industry anomaly detection. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9682–9691. [CrossRef]
22. Krupp, J.; Backes, M.; Rossow, C. Identifying the scan and attack infrastructures behind amplification DDoS attacks. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1426–1437.
23. Yuan, Z.; Lu, Y.; Wang, Z.; Xue, Y. Droid-sec: Deep learning in android malware detection. In Proceedings of the 2014 ACM Conference on SIGCOMM, Chicago, IL, USA, 17–22 August 2014; pp. 371–372.
24. Su, X.; Zhang, D.; Li, W.; Zhao, K. A deep learning approach to android malware feature learning and detection. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; pp. 244–251.
25. Li, Y.; Lu, Y. LSTM-BA: DDoS detection approach combining LSTM and Bayes. In Proceedings of the 2019 Seventh International Conference on Advanced Cloud and Big Data (CBD), Suzhou, China, 21–22 September 2019; pp. 180–185.
26. Lin, P.; Ye, K.; Xu, C.Z. Dynamic network anomaly detection system by using deep learning techniques. In Proceedings of the International Conference on Cloud Computing, San Diego, CA, USA, 25–30 June 2019; Springer: Cham, Switzerland, 2019; pp. 161–176.
27. Li, Z.; Rios, A.L.G.; Xu, G.; Trajković, L. Machine learning techniques for classifying network anomalies and intrusions. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019; pp. 1–5.
28. Kim, J.Y.; Cho, S.B. Obfuscated Malware Detection Using Deep Generative Model based on Global/Local Features. *Comput. Secur.* **2021**, *112*, 102501. [CrossRef]
29. Gomes, H.M.; Bifet, A.; Read, J.; Barddal, J.P.; Enembreck, F.; Pfharinger, B.; Holmes, G.; Abdessalem, T. Adaptive random forests for evolving data stream classification. *Mach. Learn.* **2017**, *106*, 1469–1495. [CrossRef]
30. Ramírez-Gallego, S.; Krawczyk, B.; García, S.; Woźniak, M.; Herrera, F. A survey on data preprocessing for data stream mining: Current status and future directions. *Neurocomputing* **2017**, *239*, 39–57. [CrossRef]
31. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–8.
32. Lashkari, A.H. CICFlowMeter. 2020. Available online: https://github.com/ISCX/CICFlowMeter (accessed on 8 November 2020).
33. Li, Y.; Yan, C.; Liu, W.; Li, M. A principle component analysis-based random forest with the potential nearest neighbor method for automobile insurance fraud identification. *Appl. Soft Comput.* **2018**, *70*, 1000–1009. [CrossRef]
34. Brownlee, J. A Gentle Introduction to the Bag-of-Words Model. Available online: https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (accessed on 7 August 2019).
35. Vuong, T.H.; Thi, C.V.N.; Ha, Q.T. N-tier machine learning-based architecture for DDoS attack detection. In Proceedings of the Asian Conference on Intelligent Information and Database Systems, Phuket, Thailand, 7–10 April 2021; Springer: Cham, Switzerland, 2021; pp. 375–385.
36. Ikram, S.T.; Cherukuri, A.K. Intrusion detection model using fusion of chi-square feature selection and multi class SVM. *J. King Saud Univ. Comput. Inf. Sci.* **2017**, *29*, 462–472.
37. Asghar, J.; Akbar, S.; Asghar, M.Z.; Ahmad, B.; Al-Rakhami, M.S.; Gumaei, A. Detection and Classification of Psychopathic Personality Trait from Social Media Text Using Deep Learning Model. *Comput. Math. Methods Med.* **2021**, *2021*, 5512241. [CrossRef]
38. Khattak, A.; Asghar, M.Z.; Ishaq, Z.; Bangyal, W.H.; Hameed, I.A. Enhanced concept-level sentiment analysis system with expanded ontological relations for efficient classification of user reviews. *Egypt. Inform. J.* **2021**, in press. [CrossRef]
39. Ullah, H.; Ahmad, B.; Sana, I.; Sattar, A.; Khan, A.; Akbar, S.; Asghar, M.Z. Comparative study for machine learning classifier recommendation to predict political affiliation based on online reviews. *CAAI Trans. Intell. Technol.* **2021**, *6*, 251–264. [CrossRef]