DDoS Attack Intrusion Detection System Based on Hybridization of CNN and LSTM

Ahmet Sardar Ahmed Issa[1], Zafer Albayrak[2]

[1] Department of Computer Engineering, Karabuk University, Karabuk, Turkey, 1928126532@ogrenci.karabuk.edu.tr

[2] Department of Computer Engineering, University of Applied Sciences, Sakarya, Turkey, zaferalbayrak@subu.edu.tr

Abstract: A distributed denial-of-service (DDoS) attack is one of the most pernicious threats to network security. DDoS attacks are considered one of the most common attacks among all network attacks. These attacks cause servers to fail, causing users to be inconvenienced when requesting service from those servers. Because of that, there was a need for a powerful technique to detect DDoS attacks. Deep learning and machine learning are effective methods that researchers have used to detect DDoS attacks. So, in this study, a novel deep learning classification method was proposed by mixing two common deep learning algorithms, Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM). The NSL-KDD dataset was used to test the model. This method architecture consists of seven layers to achieve higher performance compared with traditional CNN and LSTM. The proposed model achieved the highest accuracy of 99.20% compared with previous work.

Keywords: DDoS attacks, Deep learning, CNN, LSTM, NSL-KDD

1 Introduction

Currently, networks are very important for everyone because they present many features and one of the most important is the resource sharing. A network is defined as connecting two or more nodes, regardless of which nodes may be a computer, server, mobile phone, etc. The merging of computer networks in worldwide has formed the important technology is Internet that is indispensable. Today, the Internet is becoming highly vulnerable to many forms of cyberattacks. The most dangerous kind of cyber-attack is distributed denial of service (DDoS) attack [1]. In a DDoS and Denial of Service (DoS) scenario, the attacker tries to flood the host's service, making the host unavailable to legitimate users [2]. Generally, DoS attack is initiated from a single infected device or virtual machines utilizing an Internet connection whereas DDoS attacks are initiated from many different infected devices or virtual machines to overload the target systems [3]. Even if an organization has

implemented a typical security system, it will be virtually impossible to protect against a DDoS attack because of the large number of attacks in the same time and the attack is improved very fast [4]. This is largely due to the fact that DDoS attacks try to simulate normal traffic but have increased exponentially. A DDoS attack targeted GitHub [5], NETSCOUT Arbor [6], and Amazon platform [7]. These are some of the biggest DDoS attacks in the world in recent years. This has led to huge losses in industry and government globally due to DDoS attacks in recent years [8]. These problems are caused by the devices interacting with remote applications, which allows malicious agent to control the devices. The main reasons for the increase in DDoS attacks are that implementing DDoS attacks is easy and simple, does not require a great deal of technological understanding on the part of the attacker, and there were many platforms and software that could be used to coordinate the attack [9]. In general, the attackers use many devices called botnet in the DDoS attacks quickly [10].

Figure 1 shows how the attacker controls the system by connecting to the control server [11]. An efficient server with abundant resources like memory, processing power, and bandwidth is called a control server. In addition, the handlers of Botnets, also known as Agents, are the ones who receive commands from attackers. All of the attacker's commands go to the victims through these botnets. Even if malware is already installed on the compromised computer, the owner doesn't know whether it is part of a Botnet. Proxy servers are commonly used by attackers to distribute malware, execute DDoS attacks, and carry out other attacks on their victims [12]. DDoS attacks can be separated into two types. They are the application layer and the network layer [13], or they can be divided into three types [14]. At the first, volume-based attacks include UDP floods and other spoofed-packet floods. Secondly, protocol attacks cover SYN floods, Smurf DDoS, Ping of Death, fragmented packet attacks, and different types of DDoS. Lastly, application layer attacks include some advanced techniques such as SIDDOS, HTTP GET/POST floods. Security hackers are daily developing new techniques for evading defensive measures and evading detection. Therefore, daily improvement intrusion detection systems (IDS) are needed [15]. IDS is the system that can recognize a new DDoS speedily and without the need for human assistance. To increase the adaptability and accuracy of an IDS, an IDS-based machine learning has been used over the past few decades [16]. In addition, these systems are hampered by their essential reliance on previous information, their slowness, and their failure to learn from vast volumes of data. Their ability to learn isn't always powerful, either [17]. Deep learning models have recently been deployed to recognize detecting troubles, considerably increasing their chances of success [18].

In ML, deep learning (DL) is a new field that has emerged recently, the concept of which came from neural networks that mimic the human brain [19]. It has achieved successes in many areas such as speech recognition, image processing, language translation, and the IDS field [20]. Deep learning-based IDS has been found to be more effective at recognizing than traditional machine learning in several recent studies.

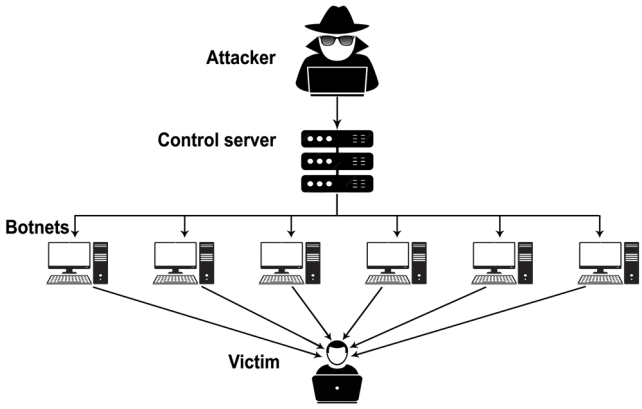

Figure 1
DDoS attack

Although deep learning algorithms analyze deeper and faster network data, none of these algorithms individually can reflect the correlation of features between multidimensional features. Another issue is that training datasets with false labels aren't taken into consideration [21].

In this paper, to solve the problems we discussed before, we proposed a new DL model that combines the Convolutional Neural Network (CNN) layers for feature extraction from input automatically [22] with the Long Short Term Memory neural network (LSTM) for predicting sequence [23]. In the proposed model design has seven layers to achieve high performance compared with each CNN and LSTM individually. The performances of in the proposed model, CNN, and LSTM were compared according to four metrics. These four metrics are accuracy, precision, recall, and F1 score. The model achieved the best accuracy among other state-of-the-arts applied to the same dataset, the NSL-KDD dataset. Other sections of the paper are arranged as follows. Sect. 2 deliberates about and concludes the related work. Sect. 3 concludes by discussing the NSL-KDD dataset and the methods used in this paper. Sect. 4 provides information on the evaluation criteria being used. Sect. 5 contains information about the experiments and the paper results. Finally, Sect. 6 is the paper's conclusion and future works.

2 Related Work

Recently, machine learning and deep learning algorithms have had great success in predicting DDoS attacks. In 2017, a feature selection approach by authors in [24] is utilized to facilitate successful intrusion detection system with machine learning. This method is the combination between DDoS Characteristic Features (DCF) and Consistency Subset Evaluation (CSE). ANN and black hole optimization approach is proposed by Kushwah and Ali [25] as a model in cloud computing for detecting

DoS attacks. Researcher in [26] proposed the Dendritic Cell Algorithm (DCA), an AIS-based algorithm for identifying most frequent denial of service attack and distributed DoS attacks that impact network communication to analyze the suggested detection method. In 2018, the researchers in [27] suggested a method based on genetic algorithm (GA) to identify DDoS attacks in cloud platform. This approach was to optimize Bernoulli Naïve Bayes BNB classifier using genetic algorithm. The H2O data mining tool was used in implementing algorithms, and a comparison of the algorithms' accuracy in DDoS attacks detection was performed [28]. Entropy estimation, co-clustering, information gain ratio (IGR) for features selection, and the Extra-Trees ensemble classifying algorithm are utilized to identify DDoS attacks; called Semi-supervised approach [29]. Network traffic data entropy is estimated and analyzed over time-based sliding windows. The second step the co-clustering algorithm divide network traffic time to three clusters when the network entropy reaches its limits. The third step is features selection represented by IGR and lastly classification algorithm is Extra-Trees ensemble. In 2019, Anjum and Shreedhara in [30] proposed an approach to improve the performance compared to the supervised and unsupervised techniques for DDoS attack detection. They proposed Semi-Supervised Machine learning technique is presented which is the combination of both supervised and unsupervised techniques. Researcher in [31] have claimed that neural networks (NN) are a good choice for DDoS detection. To develop the neural network model, the Deduct or modelling environment was employed. A single-layer perceptron for this NN model was comprised of 35 neurons (or nodes) that are (11 input neurons, 23 hidden and only one output node). A contingency table was used to evaluate the accuracy of the developed model. According to researchers in [32], they suggested to classify the incoming request as a DDoS attack and a legitimate request. A hybrid method for selecting features and classifying it is being presented. What is interesting about the work is that it relies on an available thresholding methodology with the technique of classifying, based on varied network traffic situations. This new method using the algorithm combination of Mean Absolute Deviation (MAD) thresholding and random forest (RF) classification algorithm proved to be most effective. Azizi and Hosseini in [33] have suggested a hybrid framework for DDoS detection. Processes are classified into two groups based on the outcomes. Because each group completed its own work, the speed with which work can be organized is increased as a result of this technique. Random forest appears to produce better results in both datasets under consideration (the NSL-KDD dataset and other modern dataset), however, in a particular case, any other of the algorithms may perform superior. The researcher in [15] suggested a network IDS (NIDS) that is capable of detecting a DDoS attack using ensemble classifiers and a reduced feature dataset.

The researchers in [21] addressed the major obstacles hindering the development of IoT intrusion detection systems in 2020. A unique CNN model was suggested, which uses a feature fusion method and a loss function based on cross entropy which utilizes multilayer convolution. Their solution is more advanced than current deep

learning methods, which are mostly focused on normal network intrusion problems. DDoS attacks in cloud computing can be detected and reduced using artificial immune systems (AIS) described by Prathyusha and Kannayaram [34] in 2020. According to authors in [35], a recommended architecture for DDoS classification is the auto encoder (AE) and the deep neural network (DNN) architectures developed in 2020. Initially, a naïve artificial intelligent and DNN model is generated, and hyperparameters values are randomly being used to create the model. An upgraded model is created from the baseline by enhancing it with additional algorithmic improvements. In 2020, Bagyalakshmi and Samundeeswari [36] proposed two approaches which are the filter method represented by Learning Vector Quantization (LVQ) and the dimensionality reduction method defined by Principal Component Analysis (PCA). Naïve Bayes (NB), Decision Tree (DT), and Support Vector Machine (SVM) are used to classify DDoS attacks, and these algorithms use the selected features out from each method.

3 Methodology

Deep learning is a new part of machine learning, but it has some key differences: DL needs a large amount of data to recognize the data excellently. Also, in DL, the features extracted are automatically [37]. Moreover, DL does not need to break problems down into sub-problems to solve them and gather the end result like ML, so DL directly solves the problem. Furthermore, DL takes a long time to train data in the training phase, but in the testing phase it is very quick. For these reasons, it can be summarized that deep learning has better performance than machine learning, especially with large datasets. Therefore, in the present study, two methods of deep learning were used, CNN and LSTM, and they were combined together to extract a novel method that gives better results. Figure 2 demonstrates the model of the methodology proposed in this work. In the following subsections, the dataset will be introduced as the first step. Secondly, the preprocessing technique will be implemented on the entire suggested dataset. Thirdly, the CNN and LSTM will be introduced individually. Then, the proposed model, which consists of CNN and LSTM, will be explained. Finally, in the last subsection is the learning functions and parameters.

3.1 Dataset

The NSL-KDD dataset was used to test our suggested model. Over time, the KDD'99 dataset has been refined to be more useful for algorithm performance evaluations by removing or reassigning records from classes that were previously duplicated. The NSL-KDD dataset consists of 41 features per record [38]. The NSL-KDD dataset consist of 148514 rows. In this study, the data will be divided into a training and test set. The training set is 80% and becomes 118811 rows, while the test set is 20% that becomes 29703 rows.

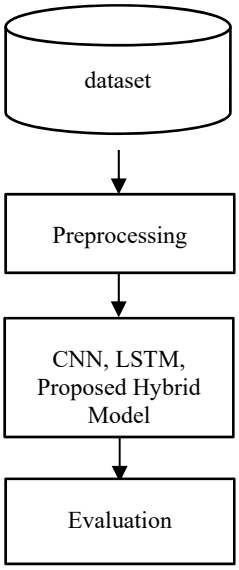

Figure 2
The proposed methodology model

3.2 Preprocessing

Data preprocessing is an important and necessary phase in the machine learning and data mining processes that involves manipulating or removing data before it is utilized for performance improvement. When dealing with a large dataset, preprocessing can be utilized to deal with multiple issues at once. Preprocessing techniques should be utilized to extract redundant data or unwanted data. Therefore, the task of preprocessing is to make the data suitable for processing in the training phase [39]. One of the preprocessing approaches used in this paper is standardizing features, which means eliminating the mean and dividing them all by the standard deviation. That is calculated as.

$$z = (x - \mu)/\delta \tag{1}$$

where μ represent the mean of the training samples. 0 will represent the mean if $with\_mean = False$. Also, δ represents the standard deviation of the training samples. But it will be 1 if $with\_std = False$. Each feature is separately centered and scaled by calculating the necessary statistics from the training set examples. By using a transform, the mean and standard deviation are stored to be used in the testing set.

3.3 Convolutional Neural Network (CNN)

This type of deep neural network, known as a "convolutional neural network," has been commonly utilized in a variety of fields due to its high performance [40].

CNNs are the most accurate multilayer neural networks; they use the same feedforward and backpropagation as other NNs' algorithms, but their architecture is unique. CNNs have the following architecture: the input layer comes first, followed by the several hidden layers, and finally the output layer [41]. Where the hidden layers are generally comprised of convolutional, pooling like maxpooling, and fully connected layers. Also, convolution process and sampling process are the two basic operations in the CNN algorithm. The convolution process applies filters to the original data or feature map that is created from the original data and then adds bias. The convolution process is conceptualized as a one-dimensional process with a specified input *I(t)* and a kernel *K(a)* The process to calculate the convolution may be summarized as follows.

$$s(t) = \sum_a I(t+a) \cdot k(a) \tag{2}$$

The core of the process is that the kernel is a considerably smaller collection of multiple points of data than the data input, but when the input is equal to the kernel, the convolution process output is greater. Moving along the network, using a technique called sampling to lessen their dependency on the precise placement of elements. Max-pooling seems to be the most widely used pooling method, and hence, it is mostly found in this layer. A technique of selecting the biggest element inside small region in the certain pooling region is known as "max-pooling". when the stride is set to two, the max-pooling layer output will be halved [42]. In the present study, CNN was comprised of five layers. Firstly, the data comes from the NSL-KDD dataset and it is preprocessed. This layer is called the input layer. After that comes the convolutional layer, which is one dimension (Conv1D). With the parameters: filter equals 10, kernal_size equals 3, and stride equals 1. Also, the activation function is a Rectified Linear Unit (ReLU) function, which will be explained afterward. The next layer is the max pooling layer, which has one dimension, and the pooling size is equal to 2. Before data was moved to a last layer, the flatten layer flattened it because the pooling size was greater than one. Softmax is the activation function utilized with the last layer (a fully connected layer). The CNN parameters are tabulated in Table 1.

Table 1
CNN parameter setting

Algorithm	*Initializer*	*Activation Function*	*Optimizer*	*Epochs*
CNN and LSTM	glorot_uniform	Relu, Softmax	Adam	500

In the table above, the term "activation function" refers to $f: R \rightarrow R$ [43]. There are many different activation functions but for these non-linear functions, the non-linear activation functions are necessary. A non-linear activation function with a finite number of possible values was published in the literature in the past. Activation functions such the Rectified Linear Unit ReLU function and Softmax function are often employed, especially because they are the most prevalent. Generally, in the output layer, the softmax function and Cross Entropy loss function are combined and utilized for multi-classification activities. The Softmax layer standardizes

outputs of the preceding layer in order to be one. The preceding layer model's units represent the un-normalized score that the input belongs to a specific class. This layer has normalized by the Softmax, therefore the output value indicates the likelihood of each class [43]. The ReLU function will return 0 as an output if the input is less than 0, while it will return the same input number if the input is higher than 0.

$$softmax(x) = \frac{e^{x1}}{\sum_{c=1}^{n} e^{xc}} \tag{3}$$

$$ReLU(x) = max(x, 0) \tag{4}$$

ReLU functions are mathematically a lot simpler because both forward and backward passes through a ReLU are simple statements. There is an enormous benefit in situations when a network has a large number of neurons because the training and assessment duration may be considerably reduced [43].

3.4 Long Short Term Memory Neural Network (LSTM)

LSTMs are a common kind of recurrent neural network (RNN) built primarily for the purpose of learning long-term reliance. An RNN and an LSTM network are both neural networks with the same structure. There is a major distinction between LSTM and RNN's basic unit since LSTM has a memory block built in. The LSTM memory blocks are called cells that are responsible for remembering things. Also, the cells are controlled by three techniques called gates: the Forget gate, the Input gate, and the Output gate. A forget gate is in charge of erasing unwanted data from the cell state. Where adding information to the cell's state is a responsibility of the input gate. At the same time, extracting valuable info from the current cell state and displaying it as an output, it is managed from the output gate side. A complete overview of LSTM is shown in Figure 3.

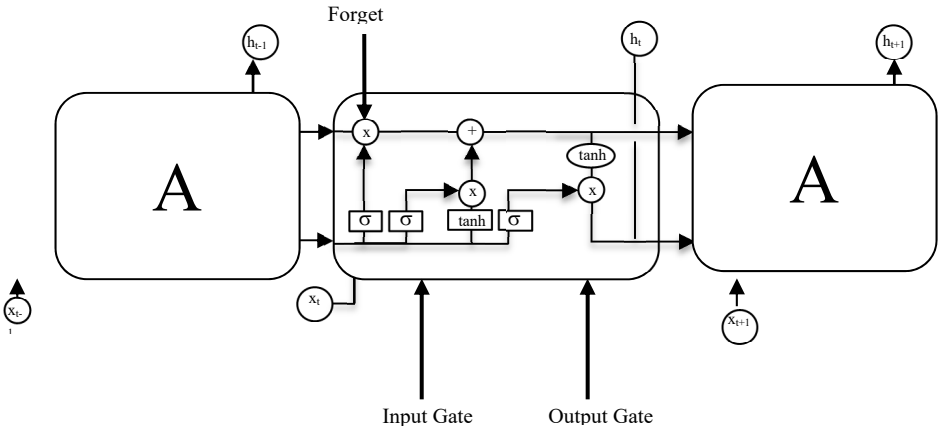

Figure 3
LSTM with its gates [44]

In the current study, LSTM was made up of three layers. Intially, the input layer is the same as CNN, which is the preprocessing layer followed by an LSTM layer. The LSTM layer has an activation function named ReLU of 41 units, and the initializer is glorot_uniform. Finally, it is a fully connected layer. Softmax is the activation function used for this layer, as CNN. The LSTM parameters are tabulated in Table 1.

3.5 Proposed Model

Proposed model is a hybrid method that combines CNN and LSTM into a single model that consists of seven layers. The present study combined CNN with LSTM in order to indicate the high quality of detecting DDoS attacks. Figure 4 illustrates the overall architecture of the suggested propose model. The figure includes seven layers. As it is mentioned in the following paragraph:

Intially, the input layer is the same as the first layer in CNN and LSTM, which is the preprocessing data followed by the convolutional layer, which is one dimension (Conv1D). With the parameters: filter equals 10, kernal_size equals 3, stride equals 1, and the activation function is a ReLU function.

The next layer is the max pooling layer, which has one dimension, and the pooling size is equal to 2. The second layer is repeated in the fourth layer, and the third layer is repeated in the fifth layer. Moreover, the next layer, the LSTM layer, has the same activation function as the second and fourth layers. The last layer in the proposed model, like the CNN and LSTM output layers, is a fully connected layer with softmax activation function.

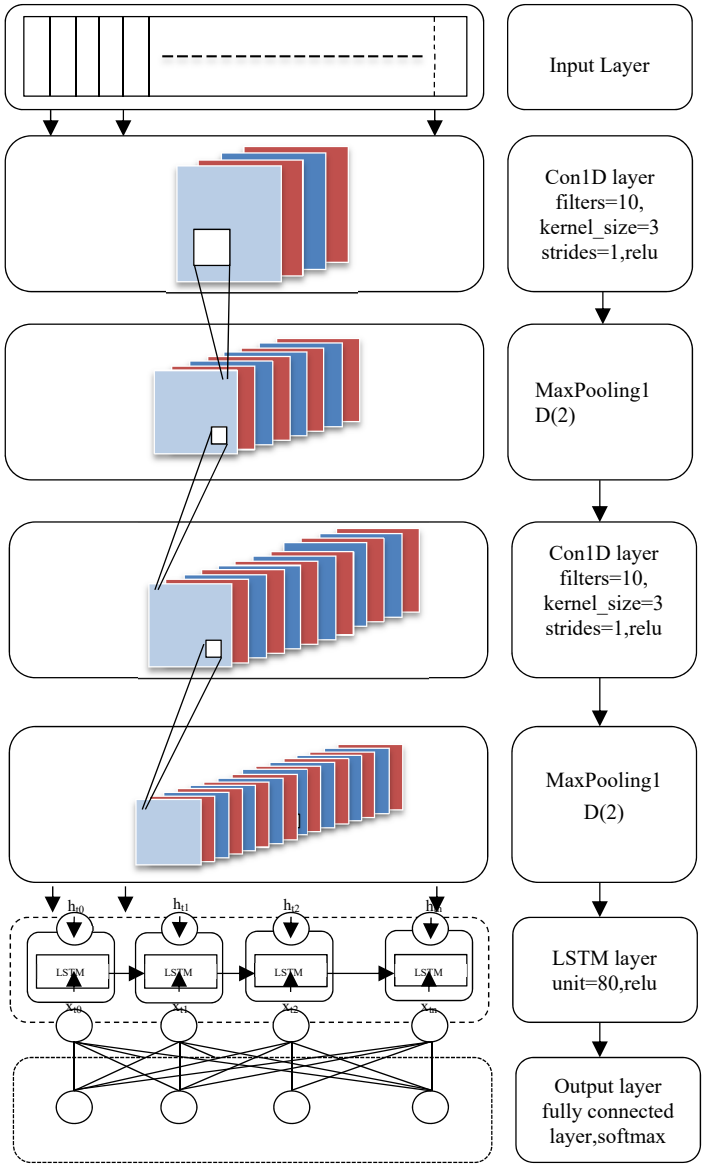

Figure 4
The General Structure of the Proposed Model

3.6 Learning

The glorot_uniform initializer was used as the kernel_initializer to initialize the weights for all CNN, LSTM, and in the proposed model methods [45]. This glorot_uniform function is useful for obtaining samples from a distribution of uniform within the bounds of two limitations. The limitation is the square root of six divided by (*fan-in + fan-out*). In the same time, the number of weight tensor input units is represented by *fan-in* and the number output weight tensor output unit is represented by *fan-out*. The weights were updated in the training phase, and in the same phase, the backpropagation technique was used. The Sparse Categorical Cross-entropy is a loss function that is utilized to compute the error, where the error is the difference between the predicted value $f(x_i,\theta)$ and the actual value y:

$$SCCE = -\sum_{i=1}^{n} y_i \, log \, f(x_i, \theta) + (1 - y_i) \, log\big(1 - f(x_i, \theta)\big) \quad (5)$$

The error will move backward across the network while the weights wait for themselves to become current. All intermediate nodes between layers are, therefore, linked, and they all will contribute their error values to forward propagation as it passes through them. The propagation mechanisms, both forward and backward, wrapped the entirety of the network [43]. In the current paper, a stochastic gradient descent optimizer known as Adaptive Moment Estimation (ADAM) [46] was employed for weight updating, with a learning rate of 0.0001. Learning rate is an important hyperparameter to minimize loss function because it controls the weight update. The learning rate must be right, not tiny or huge because the tiny learning rate makes the processing in the training phase slow, and at the same time, being too high can cause unwanted divergent action in the loss function. During processing in the training phase, the networks went through 500 epochs of repetition. Where one epoch refers to one pass forward and one pass backward of all the data in the training phase or a comprehensive training cycle of all the data. Also, the size of the batch is equal to 32.

4 Evaluation Criteria

In the present study, the evaluation criterias were applied on NSL-KDD dataset testing. The evaluation of results composed of four criterias, which were Accuracy, Precision, Recall and F1 score. The results of the present study were classified according to normality and abnormality. In each result, there were four expectations, namely: True Positive (TP) is the correct recognition of DDoS attacks; True Negative (TN) is the correct recognition of normal records; False Positive (FP) identified DDoS attacks incorrectly; and False Negative (FN) recognizes normal records incorrectly.

Accuracy: indicates the correct predicts from all predications.

$$Accuracy = \left(\frac{TP+TN}{TP+TN+FP+FN}\right) \quad (6)$$

Precision (P): is a measure of a system's ability to distinguish between an assault and what is considered normal [47].

$$Precision = \left(\frac{TP}{TP+FP}\right) \quad (7)$$

Recall or true positive rate: represent the number of predicted DDoS attacks in real DDoS attacks [48].

$$Recall = \left(\frac{TP}{TP+FN}\right) \quad (8)$$

F1 score: The F1 score can be defined as a harmonic average of recall and precision, and the F1 score result is between the worst 0 and the best 1 [49].

$$F1\ score = \left(\frac{2TP}{2TP+FP+FN}\right) \quad (9)$$

5 Experiment and Results

In the current study, the experiments were formed by Python language. Python is an efficient high-level and object-oriented programming language. A wide range of machine learning, artificial intelligence and computation libraries are available by Python, such as: NumPy, SciPy, Scikit Learn, Keras, Theano and many others [50]. The Keras library which provided by Python, was used to create and train suggested models, and it was executed on TensorFlow's framework. TensorFlow is a free and open-source framework that may be used for high-performance numerical computing. The TensorFlow is a flexible and extensible architecture that makes it possible to run computation easily on many platforms (Tensor Processing Unit, Graphics Processing Unit, Central Processing Unit), on desktops, in data centers, on mobiles, and many other devices.

In the present study, five experiments were conducted for each of the upcoming methods: CNN, LSTM, and in the proposed model to obtain comprehensive results. The mean, median, and standard deviation (SD) of accuracy, precision, recall, and F1 score for each of the aforementioned methods were indicated in order to be able to make a comparision between them, as it is shown in Table 2, Table 3, and Table 4. Table 2 illustrates the suggested CNN's performance for each fold. Shown in the fourth fold the accuracy was the highest at 97.83%.

While the precision rate was the highest in the fifth fold 98.23%. Furthermore, recall was considered as the highest rate in the first fold with the percentage of 97.92%. Moreover, in the fifth fold, F1 score was demonstrated as the highest rate by 98.00%. The mean of accuracy, precision, recall, and F1 score was 1, 2, 3, and 4 respectively. Table 3 represents the suggested LSTM's performance for each

iteration. As it is obvious in the middle table the results of the fourth fold were the highest ones among all of the folds. The accuracy, precision, recall, and F1 score in mentioned fold were 98.97%, 84.19%, 84.39%, and 84.28% respectively. Moreover, the mean of every five iterations of LSTM method for each metric (accuracy, precision, recall, and F1 score) were 97.25%, 79.55%, 78.64%, 78.65% respectively.

Table 2
The suggested CNN's performance for each fold

Fold	*Accuracy*	*Precision*	*Recall*	*F1 score*
1	97.67	97.94	**97.92**	97.92
2	97.80	93.77	83.67	83.72
3	97.74	83.80	83.65	83.72
4	**97.83**	84.16	83.55	83.85
5	97.75	**98.23**	97.78	**98.00**
Mean	97.76	91.58	89.31	89.44
Median	97.75	93.77	83.67	83.85
Standard deviation	0.061	7.160	7.793	7.776

Table 3
The suggested LSTM's performance for each fold

Fold	*Accuracy*	*Precision*	*Recall*	*F1 score*
1	97.55	83.12	81.30	82.17
2	98.57	83.87	83.66	83.75
3	98.23	79.19	83.91	81.10
4	**98.97**	**84.19**	**84.39**	**84.28**
5	92.93	67.38	59.96	61.95
Mean	97.25	79.55	78.64	78.65
Median	98.23	83.12	83.66	82.17
Standard deviation	2.471	7.092	10.513	9.421

Table 4 demonstrates the performance of the proposed model for every five iterations. As it is mentioned the second fold achieved the highest metrics. In the second fold as it is seen, accuracy, precision, recall, and F1score were 99.31%, 99.18%, 99.18%, 99.18% respectively. Furthermore, the mean of accuracy was 99.20%, while the mean of precision was 91.94%. Also the mean of recall was 93.37%, and the final mean metric was 92.41%. The current study was conducted to indicate that using the hybrid method, which consisted of CNN and LSTM, obtained better results than using them separately.

As it is clear in terms of comparison and Figure 5, proposed model was much improved than others in terms of the four metrics. Also, the mean, max, and min of every metric of proposed model were more elevated than CNN and LSTM methods, but proposed model terms of SD only recall was better than the others.

Table 4
The suggested in the Proposed Model's performance for each fold

Fold	*Accuracy*	*Precision*	*Recall*	*F1 score*
1	99.21	92.01	99.10	94.36
2	**99.31**	**99.18**	**99.18**	**99.18**
3	99.11	99.03	98.99	99.01
4	99.19	84.75	84.78	84.77
5	99.20	84.71	84.79	84.75
Mean	99.20	91.94	93.37	92.41
Median	99.20	92.01	98.99	94.36
Standard deviation	0.071	7.188	7.835	7.250

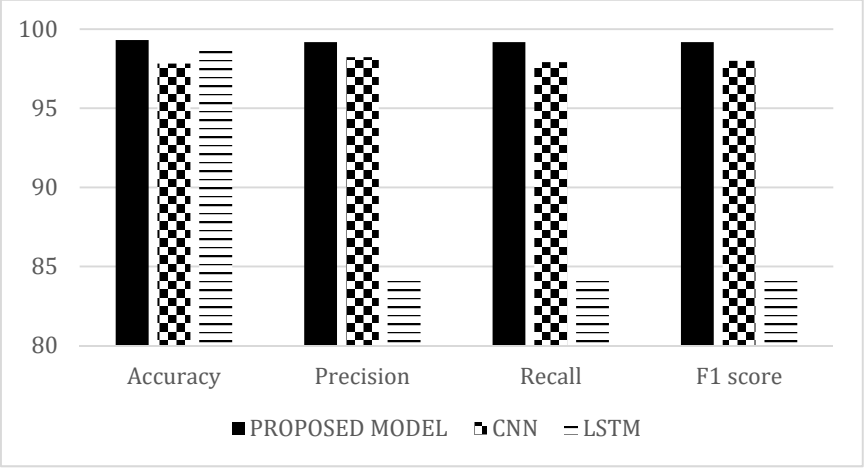

Figure 5
The performance comparison between CNN, LSTM, and the proposed model based on mean

Table 5
The comparison of proposed model with many state-of-the-art approaches in term of accuracy

No	*Name*	*Year*	*Accuracy (%)*	*Technique*
1	Our proposed model	current	99.20	Proposed Hybrid Model
2	Yusof et al. [24]	2017	91.7	DCF + CSE
3	Kushwah and Ali [25]	2017	96.3	ANN + black hole optimization algorithm
4	Igbe et al. [26]	2017	98.6	DCA
5	Derakhsh et al. [27]	2018	82.44	GA
6	Hoon et al. [28]	2018	93.26	DRF
7	Idhammad et al. [29]	2018	98.23	semi-supervised

8	Anjum and Shreedhara [30]	2019	93.26	semi-supervised
9	Mukhametzyanov et al. [31]	2019	97.94	NN
10	Verma et al. [32]	2019	98.23	MAD+RF
11	Hosseini and Azizi [33]	2019	98.9	hybrid technique
12	Das et al. [15]	2019	99.1	Ensemble technique
13	Ma et al. [21]	2020	92.99	CNN
14	P.-K.-Y.[34]	2020	96.7	AIS
15	Bhardwaj et al. [35]	2020	98.43	AE+DNN
16	B. and S. [36]	2020	98.74	LVQ+DT

Table 5 demonstrates the comparison of proposed model with many state-of-the-art approaches in terms of accuracy. As shown in the table, there are no hybrid techniques of two deep learning algorithms in the previous work on the NSL-KDD dataset but there are many good techniques such as: ensemble technique, hybrid technique, semi-supervised technique and others. By comparing the present study with them, the present study achieved the highest result and the accuracy rate was 99.20%.

From the results of the experiments, it is seen that the hybridization of two deep learning technologies, CNN and LSTM, leads to excellent results in detecting DDoS attacks depending on their architecture. In addition to that, the functions and parameters used in the learning have a magical effect to make the proposed model more accurate. This hybridization that relies on CNN as a feature extractor and LSTM as a predictor has a better accuracy when compared to each one individually. Moreover, from the comparison of proposed model and previous work of the same dataset, the NSL-KDD dataset, it is found that the current method has the best accuracy in detecting DDoS attacks. It has become apparent for the researcher that the usage of proposed model was greater than the usage of DL, and traditional ML algorithms.

Conclusion

The results obtained in the present study indicated that the proposed model has higher performance than CNN and LSTM in terms of accuracy, precision, recall, and F1 score. Also, the mean of the four metrics' accuracy, precision, recall, and F1 score rate are 99.20%, 91.94%, 93.37%, and 92.41%, respectively. Moreover, the DDoS detection in the NSL-KDD dataset achieved the highest accuracy among other previous studies. The findings of the current study indicate that the proposed model is better than using CNN and LSTM separately on this dataset. The present study can contribute to making DDoS attack detection more accurate. For future work, the present study suggests that proposed model be implemented in various sectors, not only for attack detection. Furthermore,we propose improving the architecture used from serial to parallel and introducing voting technology to it.

References

[1] Bharot, N. et al.: *DDoS Attack Detection and Clustering of Attacked and Non-attacked VMs Using SOM in Cloud Network*. In: International Conference on Advances in Computing and Data Sciences. Springer, 2019, pp. 369-378

[2] Baykara, M., Das, R.: A Novel Hybrid Approach for Detection of Web-Based Attacks in Intrusion Detection Systems. *International Journal of Computer Networks and Applications*, *4*(2), 2017, pp. 62-76

[3] ISSA, Ahmed Sardar Ahmed, and Zafer ALBAYRAK. "CLSTMNet: A Deep Learning Model for Intrusion Detection." *Journal of Physics: Conference Series*. Vol. 1973, No. 1, IOP Publishing, 2021

[4] Özalp, A. N et al.: Layer-based examination of cyber-attacks in IoT. *International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2022, pp. 1-10

[5] Hyder, H. K., Lung, C. H.: Closed-Loop DDoS Mitigation System in Software Defined Networks. *DSC 2018 - 2018 IEEE Conf. Dependable Secur. Comput.*, 2019, pp. 1-6

[6] Musotto, R., Wall, D. S.: More Amazon than Mafia: analysing a DDoS stresser service as organised cybercrime. *Trends Organ. Crime*, 2020

[7] Chen, W. et al.: *Intrusion Detection for Modern DDoS Attacks Classification Based on Convolutional Neural Networks*. In: Studies in Computational Intelligence. Springer, Cham, 2021, pp. 45-60

[8] Alabadi, Montdher, and Yuksel Celik. "Anomaly detection for cyber-security based on convolution neural network: A survey." *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2020

[9] Lima Filho, F. S. De et al.: Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning. *Secur. Commun. Networks*, 2019

[10] Atasever, S. et al.: Siber Terör ve DDoS. *Süleyman Demirel University Journal of Natural and Applied Sciences*, *23*(1), 2019, pp. 238-244

[11] Tuan, T. A. et al.: Performance evaluation of Botnet DDoS attack detection using machine learning. *Evol. Intell.*, 13 (2), 2020, pp. 283-294

[12] Beitollahi, H. et al.: ConnectionScore: A Statistical Technique to Resist Application-layer DDoS Attacks. *J. Ambient Intell. Humaniz. Comput.*, 5 (3), 2014, pp. 425-442

[13] Ajeetha, G., Madhu Priya, G.: Machine Learning Based DDoS Attack Detection. *2019 Innov. Power Adv. Comput. Technol. i-PACT 2019*, 1, 2019, pp. 1-5

[14] Yusof, M. A. M. et al.: Detection and Defense Algorithms of Different Types of DDoS Attacks Using Machine Learning. *Lect. Notes Electr. Eng.*, 488, 2018, pp. 370-379

[15] Das, S. et al.: DDoS Intrusion Detection Through Machine Learning Ensemble. *Proc. - Companion 19th IEEE Int. Conf. Softw. Qual. Reliab. Secur. QRS-C 2019*, 2019, pp. 471-477

[16] Naveen Bindra, Manu Sood: Detecting DDoS Attacks Using Machine Learning Techniques and Contemporary Intrusion Detection Dataset. *Autom. Control Comput. Sci.*, 53 (5), 2019, pp. 419-428

[17] Otoum, Y. et al.: DL-IDS: a deep learning–based intrusion detection framework for securing IoT. *Trans. Emerg. Telecommun. Technol.*, (September 2020), 2019

[18] Obaid, K. B. et al.: Deep Learning Models Based on Image Classification: A Review. *Int. J. Sci. Bus.*, 4 (11), 2020, pp. 75-81

[19] Aytaç, T. et al.: Detection DDOS Attacks Using Machine Learning Methods. *Electrica*, *20*(2), 2020, pp. 159-167

[20] Yuan, X. et al.: Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Trans. neural networks Learn. Syst.*, 30 (9), 2019, pp. 2805-2824

[21] Ma, L. et al.: A Deep Learning-Based DDoS Detection Framework for Internet of Things. *IEEE Int. Conf. Commun.*, 2020-June, 2020

[22] Tasdelen, A., Sen, B.: A hybrid CNN-LSTM model for pre-miRNA classification. *Sci. Rep.*, 11 (1), 2021, pp. 1-9

[23] Donahue, J. et al.: *Long-term Recurrent Convolutional Networks for Visual Recognition and Description*. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015, pp. 2625-2634

[24] Yusof, A. R. A. et al.: Adaptive feature selection for denial of services (DoS) attack. *2017 IEEE Conf. Appl. Inf. Netw. Secur. AINS 2017*, 2018-Janua, 2017, pp. 81-84

[25] Kushwah, G. S., Ali, S. T.: Detecting DDoS attacks in cloud computing using ANN and black hole optimization. *2nd Int. Conf. Telecommun. Networks, TEL-NET 2017*, 2017, pp. 1-5

[26] Igbe, O. et al.: Denial of service attack detection using dendritic cell algorithm. *2017 IEEE 8th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2017*, 2018-Janua (October), 2017, pp. 294-299

[27] Derakhsh, A. M. et al.: Using Genetic Algorithm to Improve Bernoulli Naïve Bayes Algorithm in Order to Detect DDoS Attacks in Cloud Computing Platform. *Int. J. Sci. Eng. Investig.*, 7 (March), 2018

[28] Hoon, K. S. et al.: Critical review of machine learning approaches to apply big data analytics in DDoS forensics. *2018 Int. Conf. Comput. Commun. Informatics, ICCCI 2018*, (1), 2018, pp. 2-6

[29] Idhammad, M. et al.: Semi-supervised machine learning approach for DDoS detection. *Appl. Intell.*, 48 (10), 2018, pp. 3193-3208

[30] Anjum, M., Shreedhara, K. S.: Performance Analysis of Semi-Supervised Machine Learning Approach for DDoS Detection. *Int. J. Innov. Res. Technol.*, 6 (2), 2019, pp. 144-147

[31] Mukhametzyanov, F. et al.: The neural network model of DDoS attacks identification for information management. *Int. J. Supply Chain Manag.*, 8 (5), 2019, pp. 214-218

[32] Verma, P. et al.: An Adaptive Threshold-Based Attribute Selection to Classify Requests Under DDoS Attack in Cloud-Based Systems. *Arab. J. Sci. Eng.*, 45 (4), 2019, pp. 2813-2834

[33] Hosseini, S., Azizi, M.: The hybrid technique for DDoS detection with supervised learning algorithms. *Comput. Networks*, 158, 2019, pp. 35-45

[34] Prathyusha, D. J., Kannayaram, G.: A cognitive mechanism for mitigating DDoS attacks using the artificial immune system in a cloud environment. *Evol. Intell.*, (0123456789), 2020, pp. 1-12

[35] Bhardwaj, A. et al.: Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of ddos attacks in cloud. *IEEE Access*, 8, 2020, pp. 181916-181929

[36] Bagyalakshmi, C., Samundeeswari, E. S.: DDoS attack classification on cloud environment using machine learning techniques with different feature selection methods. *Int. J. Adv. Trends Comput. Sci. Eng.*, 9 (5), 2020, pp. 7301-7308

[37] Barik, K. et al.: Applied Artificial Intelligence Cybersecurity Deep: Approaches, Attacks Dataset, and Comparative Study. *Applied Artificial Intelligence*, *36*(1), 2022, pp. 1-24

[38] Nandi, S. et al.: Detection of DDoS Attack and Classification Using a Hybrid Approach. *ISEA-ISAP 2020 - Proc. 3rd ISEA Int. Conf. Secur. Priv. 2020*, 2020, pp. 41-47

[39] Zulkepli, F. S. et al.: Data pre-processing techniques for publication performance analysis. *Lect. Notes Data Eng. Commun. Technol.*, 5, 2018, pp. 59-65

[40] Kesenek, Y., Özçelik, İ., & Kaya, E. (2022) A new document classification algorithm against malicious data leakage attacks. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *37*(3), 2022, pp. 1639-1654

[41] Özyurt, F.: UC-Merced Image Classification with CNN Feature Reduction Using Wavelet Entropy Optimized with Genetic Algorithm. *International Information and Engineering Technology Association*, 37(3), 2020, pp. 347-353

[42] Zeng, H. et al.: Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics*, 32 (12), 2016, pp. i121–i127

[43] Ketkar, N., Santana, E.: *Deep Learning with Python*. Springer, 2017

[44] Gudikandula, P.: *Recurrent Neural Networks and LSTM explained | by purnasai gudikandula | Medium*. no date

[45] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. *Proc. Thirteen. Int. Conf. Artif. Intell. Stat.*, 2010, pp. 249-256

[46] Kingma, D. P., Ba, J. L.: Adam: A method for stochastic optimization. *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, 2015, pp. 1-15

[47] Bhuyan, M. H. et al.: Network Anomaly Detection: Methods, Systems and Tools. *IEEE Commun. Surv. Tutorials*, 16 (1), 2014

[48] Macías, S. G. et al.: ORACLE: Collaboration of Data and Control Planes to Detect DDoS Attacks. 2020

[49] De, V. et al.: Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms. *Comput. Networks*, 186, 2021

[50] Pedregosa, F. et al.: Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12, 2011, pp. 2825-2830