Computers & Security 118 (2022) 102748

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

A new DDoS attacks intrusion detection model based on deep learning for cybersecurity

Devrim Akgun[a,1,*], Selman Hizal[b,2], Unal Cavusoglu[a,3]

[a] Sakarya University Department of Software Engineering, Esentepe Campus, Sakarya, 54187, Serdivan, TURKEY
[b] Sakarya University of Applied Sciences, Computer Engineering, Esentepe Campus, Sakarya 54187, Serdivan, TURKEY

ARTICLE INFO

Article history:
Received 27 November 2021
Revised 14 April 2022
Accepted 28 April 2022
Available online 2 May 2022

Keywords:
Intrusion detection system
Deep learning
Cloud security
DDoS
Data preprocessing

ABSTRACT

The data is exposed to many attacks during communication in the network environment. It is becoming increasingly essential to identify intrusions into network communications. Researchers use machine learning techniques to design effective intrusion detection systems. In this study, we proposed an intrusion detection system that includes preprocessing procedures and a deep learning model to detect DDoS attacks. For this purpose, various models based on Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Long Short Term Memory (LSTM) have been evaluated in terms of detection performance and real-time performance. We tested the suggested model using the CIC-DDoS2019 dataset, which is frequently used in the literature. We applied preprocess techniques such as feature elimination, random subset selection, feature selection, duplication removal, and normalization to the CIC-DDoS2019 dataset. As a result, better recognition performance was obtained for the training and testing evaluations. According to the test results, 99.99% for binary and 99.30% for multiclass accuracy using the CNN-based inception like model gave the best results among the proposed models. Also, the inference time of the proposed model for various sizes of test data looks promising compared to baseline models with a smaller number of trainable parameters. The proposed IDS system, together with the preprocessing methods, provides better results when compared to state-of-the-art studies.

1. Introduction

The protection of personal data and the security of cloud systems have become much more critical due to the number and types of attacks on cloud systems increasing over time. Various intrusion detection systems are being developed using machine learning methods to counter these attacks. Especially for cloud systems, Distributed Denial of Service (DDoS) attacks are hazardous and can cause severe damage to systems. Many financial losses occur due to the service being blocked, and the confidence of service recipients is damaged. It is necessary to ensure the continuous and adequate security of cloud computing infrastructures provided by the military, health, education, banking and finance, public institutions, and organizations by susceptible and expert systems. These organizations have been providing their security with a network structure that only they use for years. Corporations provide a method and network security through firewall solutions and the supervision of network security specialists. Today, hosting, maintaining, and securing information systems and infrastructures within the institution is very costly. Instead, many companies and organizations now receive centralized service through service providers such as Google Cloud, Microsoft Azure, and Amazon, which provide cloud computing services in different locations around the world. Since these cloud systems are in a centralized structure, ensuring security has become much more critical. For this reason, there are many different methods to ensure safety.

The aim of this article is to develop a Deep Learning based system that detects DDoS attack types with high accuracy and false-positive rates. For this purpose, the performances of different deep learning methods using Dense Neural Network (DNN), Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) architectures for detecting DDoS attacks were examined. Furthermore, balanced sub-datasets were produced in the study by using CIC-DDoS2019, which is widely used in the literature. Random sampling of each attack type was performed for this purpose by cleaning the repetitive data and identifying the attributes that have little or no effect on the detection system. In addition, the most effective deep learning model was given, taking into

* Corresponding author.
E-mail address: dakgun@sakarya.edu.tr (D. Akgun).
[1] [orcid=0000-0002-0770-599X]
[2] [orcid=0000-0001-6345-0066]
[3] [orcid=0000-0002-5794-6919]

https://doi.org/10.1016/j.cose.2022.102748
0167-4048/© 2022 Elsevier Ltd. All rights reserved.

account the accuracy and inference performance of the model. In the study, using min-max and logarithmic normalization methods on sub-datasets produced by random sampling, two separate sub-datasets were obtained and their effects on performance were examined. An inspection of the literature shows that very different machine learning methods are used for DDoS attacks. One of the most popular of these methods is deep learning models. Due to a large number of records in the CIC-DDoS2019 dataset, researchers generally obtained sub-datasets by randomly dividing the original dataset. However, when the studies on this data set were analyzed, it was seen that repetitive samples were generally not taken into account in the data sets obtained after feature selection. As a result, the presence of the same samples in the training and testing processes leads to high-performance results.

The main contribution of the article as follows:

- In the preprocessing section, redundant data such as zero, null, and duplicates are deleted and extracted from the CIC-DDoS2019 original dataset, which many studies in the literature ignore.
- Different feature selection techniques have been on the obtained cleaned dataset and defined a sub-dataset with the Info Gain Attribute Evaluation technique.
- A new convolutional deep learning model based on inception-like blocks has been designed. Also, the input features are separated into bytes for better processing of features with large maximum values.
- The evaluation results model has been compared with results from the baseline models and literature to show the success of the proposed model.

In this study, testing and training processes were carried out on the sub-dataset obtained by cleaning the redundant data, which is different from the studies in the literature. Thus, testing the deep learning model with data not used in its training is guaranteed, and more accurate results are obtained compared to the literature. The remaining part of this paper is organized as follows: Section 2 covers essential topics such as DDoS, Intrusion Detection Systems (IDS), Deep neural networks, and DDoS Evaluation Dataset (CIC-DDoS2019) in the field that is the subject of our study. Section 3 presents our proposed model, and Section 4 describes the experiments; Section 5 is the conclusion and future works.

2. Related work

Machine learning and deep learning-based methods have been widely used in recent studies on intrusion detection. With the developments in machine learning algorithms and deep learning algorithms emerging with access to big data, more effective intrusion detection systems are being developed in this area. We presented some of the current studies on DDoS attacks in cloud-based systems in the literature in this section.

Sharafaldin et al. reviewed the current DDoS datasets and discussed their shortcomings. They defined a new testbed by designing and implementing Attack Network and Victim Network and proposed a new dataset for DDoS attacks to evaluate IDS and IPS methods and systems. Their dataset, which they called CIC-DDoS2019, provides improvements over the limitations of the current datasets (Sharafaldin et al., 2019). De Assis et al. proposed a defense system for Software Defined Network (SDN), commonly used with the Internet of Things (IoT), to prevent DDoS attacks. They classified the attacks as DDoS and Normal and evaluated the machine learning architectures such as MLP, CNN, D-MLP, and LR. In addition to generating SDN traffic for training and testing, simulated IP flows were collected from the CIC-DDoS2019 data set (Assis et al., 2021a). Elsayed et al. proposed a deep learning model for the detection of DDoS attacks against SDN. A Recurrent Neural Network (RNN) in the form of an autoencoder was utilized to develop the deep learning model. They evaluated their deep learning model using the CIC-DDoS2019 dataset and compared it with the machine learning methods (Elsayed et al., 2020).

Li et al. propose a method consisting of 3 main parts to protect against DDoS attacks against Internet of Things (IoT) devices. The first is to speed up the entropy calculation, the second to perform early detection, and the last to optimize a detection result. They conducted experimental studies with the 1999 DARPA Intrusion Detection Evaluation Data Set, DARPA DDoS Dataset, and the UNB CIC DDoS 2019 Evaluation Dataset. The proposed method has low latency and good performance, and it is stated that it can be integrated into real-time IOT defense systems (Li et al., 2020). Alamri and Thayananthan have created a scheme to detect DDoS attacks against SDN. Threshold violations are detected thanks to its bandwidth control mechanism, and an adaptive control mechanism triggers the Extreme Gradient Boosting (XGBoost) Algorithm. Thus, it can be determined whether the traffic is normal or harmful. They presented a schema. They performed performance analyzes with CIC-DDoS2019, NSL-KDD, and CAIDA datasets (Alamri and Thayananthan, 2020). Shurman et al. proposed hybrid and deep learning-based methods to detect DoS and DrDoS attacks in IoT networks. They integrated the signature-based and anomaly-based methods to produce an efficient technique. All their implemented models include LSTM based networks, and they trained and tested their models using the CIC-DDoS2019 dataset (Shurman et al., 2020).

Jia, et al. presented convolution and LSTM based models for detecting DDoS attacks depending on traffic variations in IoT networks. They combined the CIC-DDoS2019 data set with generating the dataset using BoNeSi and SlowHTTPTest simulators for DDoS. They evaluated the performances of the developed model and discussed the suitability for IoT networks (Jia et al., 2020a). Hussain et al. proposed a method to detect DoS and DDoS attacks in IoT environments using ResNet18 deep learning architecture. They converted the network traffic data features into image representations and used the converted dataset to train ResNet18. They trained the deep learning model to classify 11 types of attacks and benign traffic (Hussain et al., 2020). Pontes et al. proposed an energy-based method for classifying various types of intrusion attacks trained using CIDDS-001, CICIDS17, and CICDDoS19 datasets. They trained a statistical model based on inverse potts using only benign samples. According to energy values, they classified traffic flow as benign or malicious (Pontes et al., 2021). Ferrag et al. proposed deep learning intrusion detection models in the context of Agriculture 4.0. They evaluated the performance of their networks which involve DNN, CNN, and RNN structures based on binary and multiclass classifications. They trained their models using TON_IoT and CIC-DDoS2019 for various DDoS attacks (Ferrag et al., 2021).

Khempetch et al. presented the CIC-DDoS2019 dataset to fix the flaws and introduce a new taxonomy for DDoS attacks, including a new categorization based on flows networks. The DNN and LSTM architecture are used to identify DDoS attacks (Khempetch and Wuttidittachotti, 2021). Amaizu et al. developed a 5G and B5G DDoS attack detection system that is both composite and efficient. The suggested detection framework consists of a composite multilayer perceptron paired with an effective feature extraction method designed to detect and return the type of DDoS assault it faced. Results indicated that the proposed framework could identify DDoS assaults with a high accuracy score of 99.66% and a loss of 0.011 after the simulations and after testing it using an industry-recognized dataset. Furthermore, the suggested detection framework's findings were compared to those of other researchers (Amaizu et al., 2021). Cil et al. proposed the DNN based model for detecting DDoS attacks on a sample of packets collected from network traffic. Because it incorporates feature extraction and classification algorithms in its structure and layers that update themselves as it is trained, the DNN model can perform successfully using the CIC-DDoS2019 dataset (Cil et al., 2021). Odumuyiwa et al. identified DDoS attacks in IoT networks by employing unsupervised machine learning techniques to categorize incoming network packets at the transport layer. Two deep learning algorithms and two clustering algorithms were developed independently for mitigating DDoS attacks in this study. Exploitation-based DDoS attacks, such as Transmission Control Protocol SYN-Flood assaults and UDP-Lag attacks, were highlighted. During the experimental phase, the algorithms were trained using the Mirai, BASHLITE, and CIC-DDoS2019 datasets. The autoencoder scored the best overall, with the highest accuracy across all datasets, according to findings (Odumuyiwa and Alabi, 2021).

Rajagopal et al. proposed a meta-classification technique based on decision jungle to conduct binary and multiclass classifications. Using Azure machine learning, the proposed model configured an ideal set of hyper-parameters together with important feature subsets to demonstrate the resilience of this method. Using three recent datasets, UNSW NB-15, CIC-IDS2017, and CIC-DDoS2019, they validate the suggested approach's efficacy (Rajagopal et al., 2021). Javeed et al. presented an SDN-enabled architecture that utilizes hybrid deep learning detection algorithms to detect cyber threats and assaults while addressing resource-constrained IoT devices using CICDDoS 2019 dataset (Javeed et al., 2021). Babi et al. combined three well-known IDS methods and the Triple Modular Redundancy (TMR) methodology with asymmetric optimization. An improved threshold can be calculated using a proposed approach as opposed to a single way. TMR enables IDS software to be dynamically threshold adjusted, reducing false alarms and undiscovered assaults. As a consequence, IDS software is more effective and may achieve better outcomes. The IDS software approach presented uses the CIC-DDoS2019 dataset (Babić et al., 2021).

Nie et al. proposed a deep learning-based intrusion detection system against unauthorized attempts to internet of things (IoT)-based edge networks. In their models, which consist of three stages, first feature selection is made on the data coming from the traffic of the edge network. Then, a generative adverse network (GAN) based attack detection method is used to detect a single attack type. By combining the methods used in this model, they also detected multiple attacks. They used CIC-DDoS2019 and CSE-CIC-IDS2018 datasets in their studies. They compared the performance of their proposed GAN-based intrusion detection framework with existing deep learning models. They achieved better results than other models in both single-class and multi-class intrusion detection (Nie et al., 2021). Shieh et al. conducted research on the effect of the Open Set Recognition OSR problem in machine learning or deep learning-based DDoS intrusion detection systems. They proposed a new deep learning-based intrusion detection framework in which previously encountered and unknown DDoS attacks are tagged by experts and sent back to the training dataset as feedback. In the performance tests they performed on the CIC-IDS2017 and CIC-DDoS2019 datasets (Shieh et al., 2021).

Nashat and Hussain propose an intrusion detection system that is adaptable and based on Multifractal Detrended Fluctuation Analysis (MFDFA) against SYN flooding. In their proposed method, relevant features are selected from TCP/IP packets to create time series. Then, Local Root-Mean-Square variations (RMS) are calculated for these time series. By comparing the local fluctuation value and the adaptive threshold value in this time series, it is determined whether it is an attack or normal traffic. Experimental studies were carried out on ESynFlood, January CIC-DDoS2019, March CIC-DDoS2019, NDSec-1, and BOUN TCP SYN datasets. The method they propose has simplicity and low computational overhead, and high accuracy rates (Nashat and Hussain, 2021). Vuong, Thi, and Ha presented a multi-layered machine learning method for the detection of malicious attacks. The method proposed by the authors consists of two main phases. In the first phase, the data from the CIC-DDoS2019 data set is first pre-processed, and then feature selection is made. In the second phase, there are training and test datasets, intrusion detection model, and evaluation. They used the CIC-DDoS2019 data set in their study and in their experiments (Vuong et al., 2021).

Kasim suggested a deep classification strategy that combines an AE model with an SVM classifier to improve anomaly identification. By comparing several SVM models, the efficiency of deep learning in anomaly detection systems has been demonstrated. The efficiency of the proposed approach was tested using the CICIDS and NSL-KDD datasets in a large-scale experiment. According to the experiment, the suggested scheme minimizes the high false-positive rate in DDoS traffic (Kasim, 2020). Gupta et al. offer a two-layer Anomaly based NIDS (A-NIDS) based on an LSTM classifier and an Improved Onevs-One approach for network intrusion detection. An LSTM classifier and numerous machine learning techniques are used in this framework to identify various attacks. At the second layer of the suggested system, the paper also introduces the Improved One-vs-One (I-OVO) approach for performing multi-class classification. The performance of this system was evaluated using the NSL-KDD, and CIC-IDS2017 datasets (Gupta et al., 2021). Ge et al. offer a new intrusion detection method for the IoT through a customized deep learning algorithm. Denial of service, distributed denial of service, data collecting, and data theft attacks are all included in a cutting-edge IoT dataset that includes IoT traces and actual attack traffic. A feed-forward neural network model with embedding layers is designed for multi-class classification (Ge et al., 2021). Wei et al. (2021) suggested a hybrid model that incorporates two deep learning-based algorithms for successful DDoS attack detection and classification. The Autoencoder component of proposed model performs successful feature extraction by automatically identifying the most important feature sets. The suggested model's Multi-layer Perceptron Network part employs compressed and reduced feature sets to address the performance overhead for multiple DDoS attack categories. According to the test results of the proposed method, it has been shown that it achieves high accuracy and F1-score over 98%.

Kozik et al. (2018) presented a distributed machine learning approach named Extreme Learning Machine (ELM) to detect attacks where EDGE computing is used to perform the complex and expensive processing tasks like classification and inspection. The data is collected on EDGE nodes and stored on Cloud servers. The authors have shown that, the complex training, learning, classification processes could be performed on cloud with the help of EDGE computing. However, the system demands more resources to ensure parallel processing while handling large data sets. Di Mauro et al. (2020) presented a review of neural-based approaches for network intrusion management where the authors compiled neural-based, deep-learning and weightless neural networks based intrusion detection techniques and evaluate them by using a dataset. They also find out the time complexity and performance of the reviewed method and presented a comparative analysis. Zhong et al. (2020) utilized a deep learning system for intrusion detection named Big Data based Hierarchical Deep Learning System (BDHDLS) which is targeted to increase the performance of Intrusion Detection Systems (IDS). To find out the characteristics and information from the traffic the propose method analyzed the behavior and content features of the data. By deploying multiple machines and parallel training techniques the system reduces the construction time of the proposed BDHDLS.

Martinez et al. (2021) presented a comparative study of different decision tree-based machine learning techniques combined with feature selection techniques to obtain higher accuracy results in their work. Algorithms such as Random Forest and XGBoost are used for feature selection. According to different attack types, feature selection processes were performed and the selected features were compared. It has been shown to detect high performance using fewer attributes, with little dropout. Kamalov et al. (2021) applied a fusion machine learning method to develop new IDS model. They used the orthogonal variance decomposition technique to determine related attirubutes in dataset. A deep neural network for intrusion detection is built using the given characteristics. It is claimed in the article the suggested technique achieves a detection accuracy of 100% in accurately detecting DDoS attacks. Di Mauro et al. (2021) examined the feature selection techniques and their performances on the datasets used recently in the literature. In this study, brief information about many different techniques is given. The data set of DDoS attacks and the selection results of different algorithms were examined using correlation maps. Feature selection time and training time were compared on the same data set. Performance evaluation is presented for different attack types on different datasets obtained as a result of feature extraction from the DDoS dataset.

In this section, IDS system designs in the literature are reviewed. When the studies in the literature are examined, it is seen that many different machine learning methods are used to detect different types of attacks. When the performances of these methods are analyzed, it has been determined that they have advantages and disadvantages compared to each other. In addition to system designs using a single algorithm, it has been seen that hybrid approaches are frequently preferred in designs. In the preprocessing process, feature selection techniques are presented as a component in the proposed systems to alleviate the workload and increase performance. It has been seen that there are many different algorithms as a feature extraction method and tests are carried out to determine the method suitable for the structure of the data set, and as a result, the method to be used is determined.

3. Background

3.1. Intrusion detection systems (IDS)

IDS is a system that consists of a software application or hardware that analyzes network traffic for malicious behavior or rule violations and provides warnings when such activity is detected. Any malicious behavior or breach is often reported to a system administrator or gathered and reported centrally via the security information and event management system (SIEM). SIEM incorporates data from numerous sources and uses alarm filtering mechanisms to distinguish between malicious and false alarms. Intrusion detection systems come in various forms, ranging from antivirus software to tiered monitoring systems that monitor the whole network's traffic. Network intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS) are the most general categories. NIDS watches and analyzes incoming network data, whereas HIDS watches and analyzes essential operating system files. IDS can also be classified based on its detecting method by signature-based or anomaly-based. Signature-based IDS uses network traffic patterns to identify threats. Signatures are a term used to describe the patterns found by an IDS. Signature-based IDS can quickly detect assaults whose pattern (signature) already exists in the system, but it is difficult to detect new malware attacks since their pattern (signature) is unknown. As new malware is generated quickly, anomaly-based intrusion detection systems (IDS) were established to detect unknown malware threats. Machine learning is used in anomaly-based IDS to develop a trustful activity model, and anything that comes in is compared to that model, and it is considered suspicious if it is not found in the model. In comparison to signature-based IDS, machine learning-based methods have more excellent generic properties because these models may be trained based on application and hardware configurations.

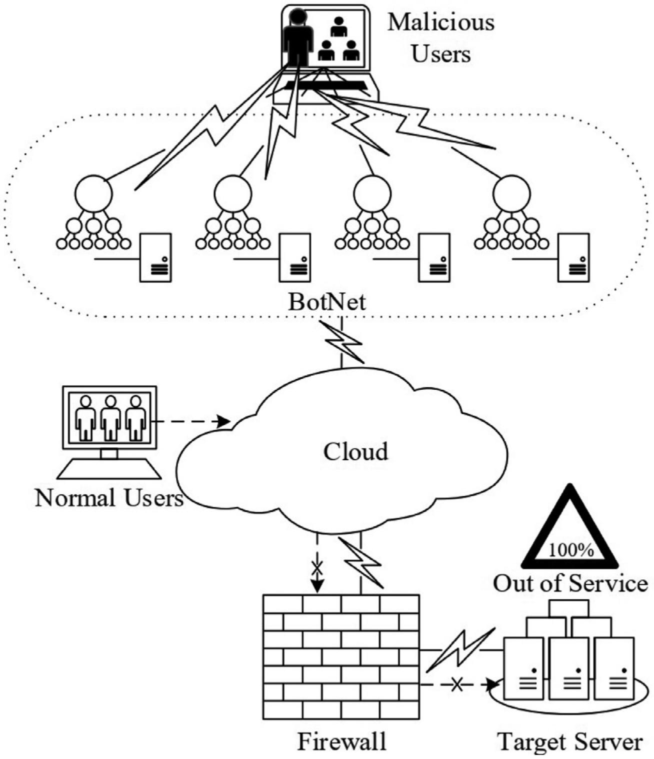


Fig. 1. An example IDS for DDoS attacks.

3.2. Distributed denial of service (DDoS)

DDoS is a pure accessibility attack, unlike attacks by malicious users such as data leakage or account hijacking. Fig. 1 shows an example DDoS attack diagram. These types of attacks are carried out using vulnerabilities of systems, applications, or protocols. For any service to be provided in the IT sector, certain values are foreseen for parameters such as the number of users, bandwidth, number of instant requests. Server and infrastructure design is carried out to handle a load slightly above these values. DDoS is a dangerous attack usually carried out on a large scale, which causes the services it provides to be completely stopped by sending instant requests and consuming resources that are far above the load that the service provider can handle.

When exposed to a DDoS attack, symptoms such as long-term service interruptions, instantaneous congestion in server resources, serious aggravation of services, and backlog in data loads due to UDP, SYN, and GET/POST are seen. It is essential to develop systems with maximum sensitivity against these attacks, which are very dangerous and costly, especially for companies providing services on cloud systems. Separating DDoS attacks from instantaneous and average performance increases/decreases requires the right technology and expertise. From the point of view of businesses, the network infrastructure should be well designed. In addition, the high level of system and TCP/IP knowledge of the relevant expert personnel is one of the primary protection measures. Today, automated intrusion detection systems using machine learning or deep learning models are becoming increasingly common.

3.3. Deep neural network

Feed-Forward Neural networks are a vital part of machine learning methods. In the case of deep learning, where it handles

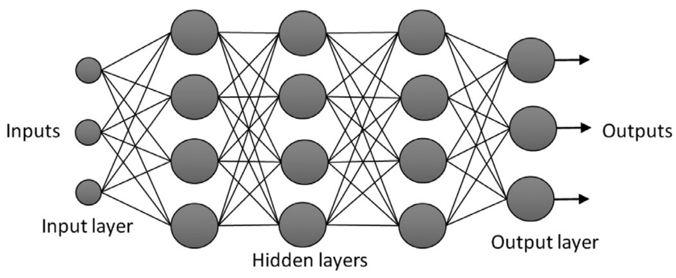


Fig. 2. The basic structure of DNN.

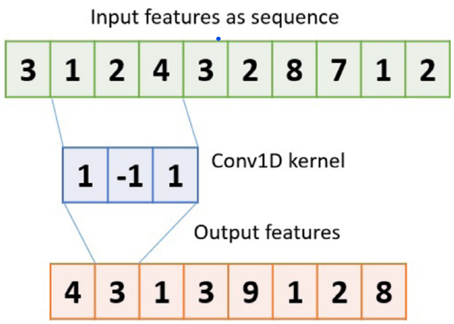


Fig. 3. The working principle of a single convolution unit.

feature extraction with the layers of neural networks and datasets large enough to provide generality, it is called DNN. Fig. 2 shows a basic structure of DNN where input, output, and hidden layers form the network. A more advanced type of neural network is CNN, where the convolution operations extract the input features before the fully connected or dense layer. CNN is also a type of feed-forward neural network and partially contains DNN for classification. The trainable weights of the convolutional neural networks are the kernel weights for each convolution. Fig. 3 shows the basic idea of a single convolution operation on a one-dimensional array. In this example, trainable weights are multiplied with the input array for each of the output elements. The kernel weights are updated by backpropagation according to loss function value.

Another widely used neural network type is the Recurrent neural network with connections fed back to inputs through trainable weights. One particular kind of RNN is the LSTM network, and it is more successful than RNN in remembering the long-term dependencies in the data. An LSTM unit generally consists of an input gate, and a forget gate and an output gate. Eq. (1) shows the fundamental relations to implement an LSTM where X stands for inputs, W and b stands for trainable weights and biases, respectively. The weights represented with U stands for weights for recurrence, and V stands for weights for the carry. An LSTM contains approximately four times more trainable weights than RNN, which is costly to operate.

$$
\begin{aligned}
output_t &= f(state_t.Uo + X.Wo + C_t.Vo + bo) \\
i_t &= f(state_t.Ui + X.Wi) + bi) \\
f_t &= f(state_t.Uf + X.Wf) + bf) \\
k_t &= f(state_t.Uk + S.Wk) + bk) \\
c_{t+1} &= i_t \times k_t + c_t \times f_t
\end{aligned}
\tag{1}
$$

3.4. DDoS evaluation dataset (CIC-DDoS2019)

In this study, the CIC-DDoS2019 dataset, the most up-to-date version published in the Canadian Institute for Cybersecurity, which has been creating datasets on cyber attacks for years and conducting statistical studies on them, was preferred. The authors obtained this dataset as a result of real-world tests using CICFlowMeter-V3 (Lashkari, 2021). This dataset can be accessed as PCAP files or files prepared in CSV format. The dataset contains the source and destination IP addresses, ports, protocols, and classes labeled by experts. The authors who created the dataset first performed classification processes in 2 main groups: reflection-based and exploitation-based. In their sub-categories, labeling was carried out in 13 different classes on a protocol basis, as shown in Fig. 4. The dataset was created between 9:43 - 17:35 on the first day for the test dataset of 7 DDoS attack types such as PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, and SYN. On the second day, between 10:35 and 17:15, 12 DDoS attack types, namely NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, and TFTP, were recorded. The authors did not consider the WebDDoS attack for their model because there was very little WebDDoS attack.

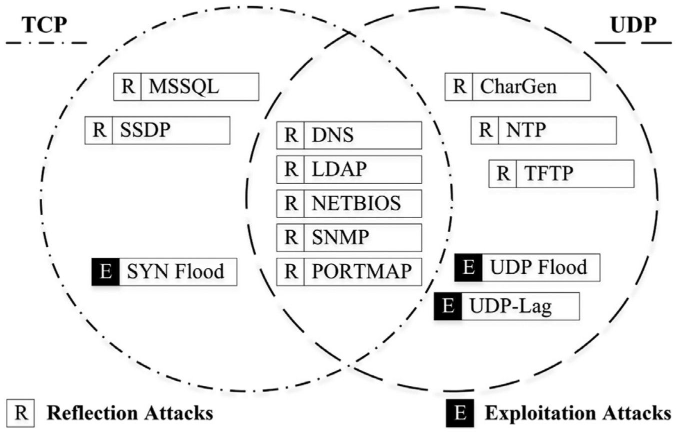


Fig. 4. DDoS attack taxonomy based on protocol.

Table 1
The number of instances of CIC-DDoS original dataset.

Class (Label)	Number of Instances	Number of Instances (Duplicates removed)
Benign	56,863	56,025
Attack	50,006,249	47,875,647
Total	50,063,112	47,931,672

4. Proposed IDS for DDoS attacks

In this section, we explained the preprocessing operations performed on the dataset. Then, we presented the proposed deep learning models.

4.1. Preprocessing operations

Fig. 5 shows the preprocessing operation to obtain the dataset to be used in the IDS. Initially, some features which we consider to be unimportant in the decision process are eliminated. Then, a random selection is applied to reduce the size of the dataset. Then duplicate records are removed to improve training. Finally, min-max and logarithmic normalization were performed to obtain two new sub-datasets for evaluations.

There are 50,063,112 records of 13 attack types in total in the CIC-DDoS2019 data set. When the data set is analyzed, it is evaluated that there are some records with all values of 0. There are very large differences between the minimum and maximum values, and when it is evaluated in terms of network traffic, some features are not important for attack detection. In the Table 1, the sample numbers of the attack types in the 88-featured data set are presented. CIC-DDoS2019 dataset contains 50,063,112 instances that includes 50,006,249 DDoS attacks and 56,863 benign (normal)

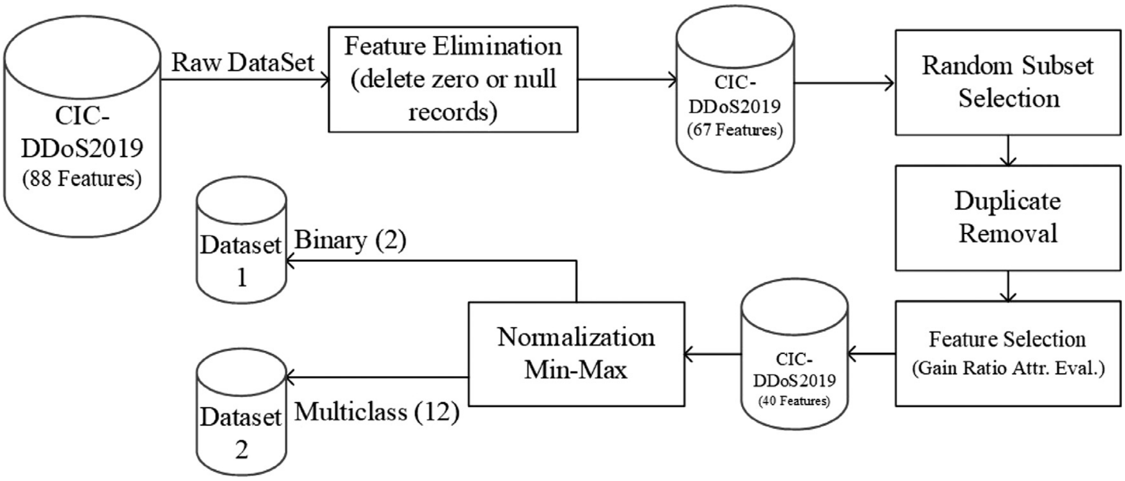


Fig. 5. Preprocessing of raw dataset for training and testing.

Table 2
The number of instances after feature selection operation .

Class (Label)	Number of Instances	Number of Instances (Duplicates removed)
Benign	56,025	51,453
Attack	616,714	609,168
Total	672,739	660,621

traffic as shown in Table 2. Duplicate records in this dataset have been removed. The total numbers after cleaning are also given in Table 1 comparatively.

The preprocessing operations performed on the data set are given below, respectively:

- **Feature elimination:** First of all, It has been determined that the values of some attributes in the dataset are all zero or null. These attributes were first cleared from the dataset. As a result of these evaluations, a new dataset with 88 features and 67 was obtained.
- **Random selection process:** Then, the same number of samples from all attack types were randomly selected based on the number of benign samples in the data set, which is 56,025. Since the number of WebDDoS samples in the dataset is 439, all of the samples are included in the dataset for this attack type. In total, a data set consisting of 672,739 samples was created.
- **Duplicate Removal:** In the original data set, 47,931,672 samples were obtained in the data set after the duplicate data were deleted. When the Table 2 is examined, it is seen that there are big differences between the number of examples of attack types. For example, while the number of samples for the WeB-DDoS attack type is 439, the sample number for the TFTP attack type is 20,082,580, and for other attack types, the number of samples varies between 56,000 and 50,000,000. Finally, duplicate samples on the 40-attributes dataset were deleted in this process. A data set consisting of 660,621 samples was obtained.
- **Feature Selection (Info Gain Attribute Evaluations):** Filtering-based feature extraction algorithms perform a sorting to detect useful features that work independently of any algorithm. These algorithms are generally lighter in computational load and produce results quickly. In filtering algorithms, the attributes that make up the data set are sorted, the attributes that fall below a certain threshold value as a result of the calculation are eliminated from the data set, and a new data set is created. In this article, the Info gain attribute evaluation algorithm (Sourceforge, 2022) is used to reduce the size of the dataset and create a smaller dataset with effective and high performance results. According to the gain ratio values given in Table 3, 40 feature selection processes were performed. Tests have been made on other algorithms in the literature, but this algorithm has been preferred because higher performance results are obtained with this algorithm compared to other selection methods. The Info gain atrribute evaluation algorithm used for feature selection processes is explained (Al Janabi and Kadhim, 2018). In this algorithm, firstly, the IG values of the attributes are calculated for segmentation. By choosing the attribute with the highest IG value, the attribute to be partitioned for N nodes is determined. Operations on this selected feature represent the minimum information requirement for classification. This approach optimizes the performance of the algorithm and guarantees a simple tree finding.The following information provides the expected information required to categorize an item in partition D. where p_i is the probability that an item in D belongs to class C_i and is computed by $|Ci, D|/|D|$. Info(D) indicates the average amount of information required to determine an object's class label in partition D.

$$\text{Info}(D) = -\sum_{i=1}^{m} p_i \log(p_i) \tag{2}$$

It is necessary to split the items in D based on some feature A with v different values $(a1, a2, a3, \dots, an)$; the expected information based on attribute A division is presented by:

$$\text{Info}_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{Info}(D_j) \tag{3}$$

Where $|D_j|/|D|$ represents the j_{th} partition's weight. As seen in Eq. (4), IG is described as the variation between the original data prior dividing and the new IG obtained following partitioning on A.

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D) \tag{4}$$

As a consequence, Gain(A) here represents the amount of profit generated by the branching process.
As the splitting attribute, the attribute with the biggest gain ratio is chosen. The non-leaf nodes of the resulting decision tree are regarded as significant features (Han et al., 2011). As a result of feature operations, a new data set with 40 features was obtained.
- **Normalization:** Data sets to be used in training validation and testing processes were obtained by applying min-max normalization processes on randomly selected sub-datasets from all

Table 3
CIC-DDoS 2019 selected 40 features using Info Gain Attr. Eval.

Features	Gain Value	Description	Min-Max Range
Packet Length Mean	2.587840	Mean length of a packet	0 - 4023.945
Average Packet Size	2.586634	Average size of packet	0 - 4023.779
Max Packet Length	2.577663	Maximum length of a packet	0 - 37,960
Avg Fwd Segment Size	2.570492	Average size observed in the forward direction	0 - 3015.91
Fwd Packet Length Mean	2.569949	Mean size of packet in forward direction	0 - 3015.91
Fwd Packet Length Max	2.569601	Maximum size of packet in forward direction	0 - 32,120
Fwd Packet Length Min	2.560382	Minimum size of packet in forward direction	0 - 2021
Subflow Fwd Bytes	2.559334	The average number of bytes in a sub flow in the forward direction	0 - 15,266,416
Total Length of Fwd Packets	2.559334	Total size of packet in forward direction	0 - 15,266,416
Min Packet Length	2.555892	Minimum length of a packet	0 - 1472
Source Port	1.470888	Source Port	0 - 65,534
act_data_pkt_fwd	1.154258	Act data in forward direction	0 - 5043
Flow Duration	1.134145	Duration of the flow in Microsecond	0 - 119,999,986
Fwd Packets s	1.099233	Number of forward packets per second	0 - 4,000,000
Flow IAT Mean	1.093613	Mean time between two packets sent in the flow	0 - 39307794.333
Flow IAT Max	1.071818	Maximum time between two packets sent in the flow	0 - 119,954,412
Fwd IAT Total	1.041866	Total time between two packets sent in the forward direction	0 - 119,999,986
Fwd IAT Mean	1.001968	Mean time between two packets sent in the forward direction	0 - 30505593.333
Fwd IAT Max	0.98594	Maximum time between two packets sent in the forward direction	0 - 119,954,412
Flow IAT Std	0.971157	Standard deviation time between two packets sent in the forward direction	0 - 68082925.446
Fwd Header Length	0.910916	Total bytes used for headers in the forward direction	-21,254,3795000 - 129,536
Fwd IAT Std	0.849177	Standard deviation time between two packets sent in the forward direction	0 - 56020637.667
Init_Win_bytes_forward	0.71427	The total number of bytes sent in initial window in the forward direction	-1 - 65,535
Total Fwd Packets	0.650225	Total packets in the forward direction	1 - 100,129
Subflow Fwd Packets	0.650225	The average number of packets in a sub flow in the forward direction	1 - 100,129
Protocol	0.603988	Protocol	0 - 17
ACK Flag Count	0.573889	Number of packets with ACK	0 - 1
Packet Length Variance	0.484914	Variance length of a packet	0 - 43778893.573
Packet Length Std	0.484914	Standard deviation length of a packet	0 - 6616.562
min_seg_size_forward	0.483161	Minimum segment size observed in the forward direction	-1408237563 - 1480
Fwd Packet Length Std	0.401855	Standard deviation size of packet in forward direction	0 - 221.556
Fwd IAT Min	0.369574	Minimum time between two packets sent in the forward direction	0 - 15,407,833
Flow IAT Min	0.369067	Minimum time between two packets sent in the flow	0 - 15,407,833
Bwd Packets s	0.247753	Number of backward packets per second	0 - 4,000,000
Destination Port	0.214434	Destination Port	0 - 65,535
Bwd Header Length	0.208336	Total bytes used for headers in the backward direction	-2125437950 - 147,280
Total Backward Packets	0.199864	Total packets in the backward direction	0 - 4602
Subflow Bwd Packets	0.199864	The average number of packets in a sub flow in the backward direction	0 - 4602
Bwd IAT Max	0.199205	Maximum time between two packets sent in the backward direction	0 - 118,159,264
Bwd IAT Mean	0.19727	Mean time between two packets sent in the backward direction	0 - 58,961,594

Table 4
Number of instances for multiclass classification using 88 and 40 features.

Class (Label)	Num. of Instances (88)	Unique Instances (88)	Num. of Instances (40)	Unique Instances (40)
WebDDoS	439	439	439	438
BENIGN	56,863	56,025	56,025	55,939
UDP_Lag	366,461	366,452	56,025	56,023
NTP	1,202,642	1,202,518	56,025	56,024
Syn	1,582,289	1,582,049	56,025	56,025
SSDP	2,610,611	2,610,580	56,025	56,025
UDP	3,134,645	3,134,598	56,025	56,025
NetBIOS	4,093,279	3,923,062	56,025	56,025
MSSQL	4,522,492	4,519,697	56,025	56,025
DNS	5,071,011	4,935,090	56,025	56,025
SNMP	5,159,870	5,048,055	56,025	56,025
TFTP	20,082,580	18,410,498	56,025	56,025
Total	47,883,182	16,267,889	616,714	616,624

attack types. As a result of this process, the final data set was obtained to be used in the training and testing operations.

Summarization of the preprocessing operations is as follows: Based on the benign sample number of 56,025 on non-duplicate data set with 88 attributes, the same number of samples from all attack types were randomly selected. After this process, a sub-dataset of 672,739 samples with an equal number of samples from each attack type was obtained. By applying the remove duplicates operation on this data set, a data set consisting of 660,621 unique data to be used in the operations in the article was created as seen in Table 2. After the remove duplication operation, the difference between samples in the data set is very small. Table 3 shows the attributes and descriptions of the 40-featured dataset used in this article. Table 4 shows the attribute numbers of the original dataset with 12 class and 88 features and the sub-dataset of 12 class and 67/40 featured dataset. Due to the similarities between the DNS and LDAP classes we excluded the LDAP class in the evaluations.

4.2. Proposed IDS models

We made performance comparisons on deep learning models with three different base-line models structure and the proposed model using the sub-datasets obtained from the CIC-IDDosS2019

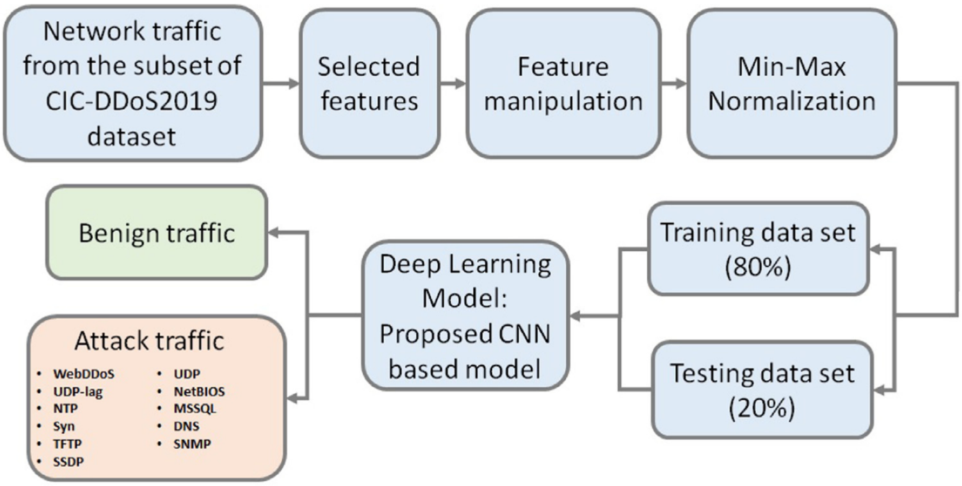


Fig. 6. Working principle of the proposed deep learning based IDS.

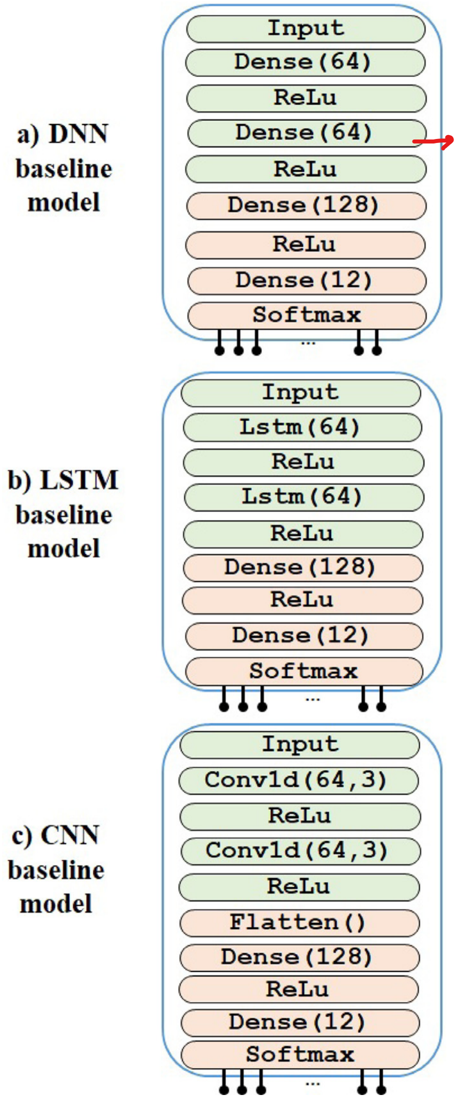
Table 5
The Confusion Matrix.

		Predicted	
		Intrusion	Normal
Actual	Intrusion	TP	FN
	Normal	FP	TN

Table 6
Performance results for DENSE-based model.

Traffic type	TPR	FPR	Precision	F_1 Score
WebDDoS	0.977273	0.000771	0.475138	0.639405
BENIGN	0.999108	0.000071	0.999286	0.999197
UDP-lag	0.968496	0.000250	0.997426	0.982748
NTP	0.996252	0.001507	0.985086	0.990638
Syn	0.999018	0.003728	0.964003	0.981198
SSDP	0.939581	0.005413	0.945487	0.942525
UDP	0.993217	0.008302	0.922803	0.956716
NetBIOS	0.966087	0.003576	0.964279	0.965182
MSSQL	0.866934	0.001757	0.980123	0.920061
SNMP	0.993039	0.003879	0.962377	0.977467
TFTP	0.979027	0.005948	0.942683	0.960511
DNS	0.943597	0.000276	0.997077	0.969600

data set studies. Fig. 6 shows the working principle of the proposed system.The selected features are divided into training and testing datasets using random selection. In addition to the proposed model, we examined various baseline deep learning models. In deep learning models, selecting proper hyperparameters includes network size, layer types, and activation functions. Especially the layer types and the size of layers significantly affect the performance of the deep learning models. Hence we experimented with different layer types in the proposed model, such as dense layer, convolutional layer and LSTM layer. Fig. 7a shows the first class of DNN based model where all four layers densely connected the neural network. The second model, as shown by Fig. 7b was implemented using a convolutional layer together with densely connected layers. Conv1D layers can be used with one-dimensional sequences to extract better features before the classification with dense layers. Similarly, LSTM based networks can also be used to process one-dimensional features together with dense layers. Hence our third model is based on the LSTM network as shown by Fig. 7c. The last two layers have been selected as dense layers for both CNN-based and LSTM based models to classify extracted features. Fig. 8 shows our proposed model based on various numbers of one dimensional convolutional (Conv1d) layers which



Dense Layer

Fig. 7. Baseline models using DNN, CNN and LSTM layers.

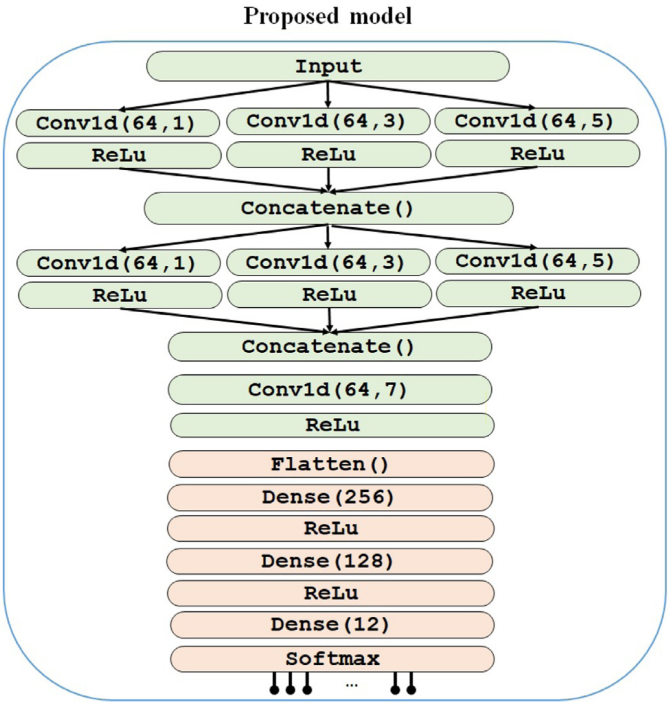


Fig. 8. Proposed model using CNN layers with inception like connections.

Table 7
Performance results for CNN-based model.

Traffic type	TPR	FPR	Precision	F_1 Score
WebDDoS	0.988636	0.000714	0.497143	0.661597
BENIGN	0.999554	0.000009	0.999911	0.999732
UDP-lag	0.971977	0.001284	0.986951	0.979406
NTP	0.993039	0.000383	0.996150	0.994592
Syn	0.998394	0.000901	0.991052	0.994709
SSDP	0.963677	0.003175	0.968083	0.965875
UDP	0.988577	0.001338	0.986639	0.987607
NetBIOS	0.991432	0.002515	0.975244	0.983271
MSSQL	0.965462	0.002060	0.979093	0.972230
SNMP	0.975814	0.002568	0.974336	0.975075
TFTP	0.985542	0.001703	0.982998	0.984268
DNS	0.983668	0.001561	0.984371	0.984019

Table 8
Performance results for LSTM-based model.

Traffic type	TPR	FPR	Precision	F_1 Score
WebDDoS	0.897727	0.000682	0.484663	0.629482
BENIGN	0.994734	0.000259	0.997405	0.996068
UDP-lag	0.962695	0.004102	0.959100	0.960894
NTP	0.996519	0.000490	0.995098	0.995808
Syn	0.980277	0.002390	0.976182	0.978225
SSDP	0.604105	0.005252	0.919951	0.729300
UDP	0.972959	0.031194	0.757083	0.851552
NetBIOS	0.940295	0.000383	0.995935	0.967315
MSSQL	0.966801	0.010451	0.902374	0.933477
SNMP	0.980544	0.006456	0.938178	0.958893
TFTP	0.977064	0.000339	0.996541	0.986706
DNS	0.973583	0.003005	0.970034	0.971805

Table 9
Performance results for the proposed CNN-based model.

Traffic type	TPR	FPR	Precision	F_1 Score
WebDDoS	0.988636	0.000527	0.572368	0.725000
BENIGN	0.999911	0.000009	0.999911	0.999911
UDP-lag	0.978313	0.000883	0.991050	0.984640
NTP	0.998037	0.000045	0.999553	0.998794
Syn	0.998751	0.000000	1.000000	0.999375
SSDP	0.985364	0.001480	0.985188	0.985276
UDP	0.992236	0.000999	0.990027	0.991130
NetBIOS	0.996341	0.000178	0.998212	0.997275
MSSQL	0.986078	0.001204	0.987929	0.987003
SNMP	0.998394	0.001186	0.988251	0.993296
TFTP	0.992503	0.000919	0.990823	0.991663
DNS	0.997233	0.000205	0.997946	0.997590
DNS	0.997947	0.000089	0.999107	0.998527

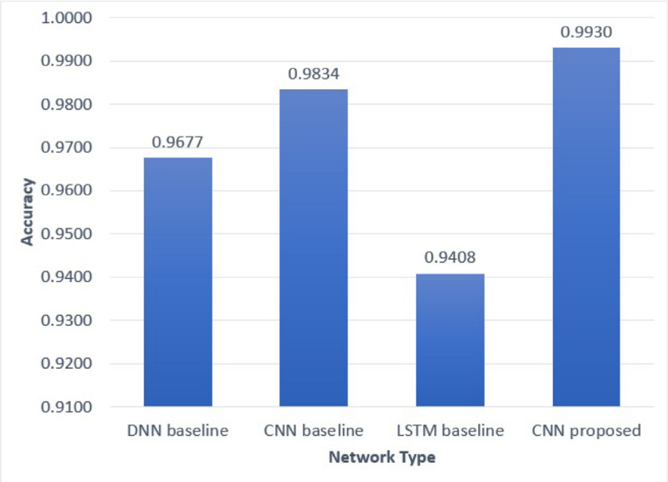


Fig. 9. Comparison of the accuracy evaluations.

determined after a number of trials. We included inception like two blocks made up of three Conv1d layers to increase the network's performance. Features obtained from various sizes of convolutions are merged with concatenate layers at the end of each inception-like block.The stride size of each Conv1D layer in the mode has been set equal to the kernel size. For example, if kernel size is 1, the stride is set to 1, and if kernel size is 5, the stride is set to 5. We also manipulated the input features by converting them into bytes based on their maximum values. The conversion helps extend some features varying large to be elaborated better. Therefore there are a total of four models for experimental evaluations. This variety also helps compare the inference times of models apart from the attack detection performances. It gives an idea about the limitations of the developed models when they are put into real-time operation.

5. Experimental results

In this section, we evaluated the proposed models using the sub dataset given in Table 14. We realized experiments on a computer with a processor AMD Ryzen 2700 with eight cores, the main memory of 32 GB, and a video card NVIDIA® GeForce GTX 1080 8 GB. The experimental measurements were done in the Ubuntu 20.04 (Source, 2021c) operating system using Python 3.8 (Source, 2021b) with Keras 2.5.0 (Source, 2021a) framework to implement and train the proposed deep learning models. We selected Adam as the optimization method, and the number of epochs, the learning rate, and batch size is 500, 3e-4, and 32, respectively. The most successful networks were saved while testing all experimented hyperparameters using the Keras library's ModelCheckpoint class.

Table 5 shows the confusion matrix. TPR, FPR, Precision and F_1 score metrics are computed according to Eqs. (5)–(8) respectively. These metrics given by Tables 6–9 also show the experimented models' success for each of the attack classes and Benign class.

$$TPR = \frac{TP}{TP + FN} \quad (5)$$

Table 10
Confusion matrix for DNN based model.

	WebDDoS	BENIGN	UDP-lag	NTP	Syn	SSDP	UDP	NetBIOS	MSSQL	SNMP	TFTP	DNS
WebDDoS	**86**	0	0	0	0	0	0	0	0	0	0	2
BENIGN	3	**11195**	1	1	3	0	0	0	0	0	0	2
UDP-lag	1	0	**10852**	0	202	68	67	0	11	1	2	1
NTP	21	1	0	**11163**	2	0	0	9	0	5	1	3
Syn	0	0	7	0	**11194**	0	1	1	0	0	0	2
SSDP	28	3	4	1	0	**10528**	365	1	130	135	10	0
UDP	0	0	0	0	0	2	**11129**	1	43	2	27	1
NetBIOS	4	0	0	36	1	37	24	**10825**	13	241	21	3
MSSQL	0	1	0	0	0	124	473	237	**9714**	43	597	16
SNMP	0	1	0	2	0	70	0	2	0	**11127**	3	0
TFTP	8	0	16	1	205	1	1	0	0	2	**10970**	1
DNS	30	2	0	128	5	305	0	150	0	6	6	**10573**

Table 11
Confusion matrix for CNN based model.

	WebDDoS	BENIGN	UDP-lag	NTP	Syn	SSDP	UDP	NetBIOS	MSSQL	SNMP	TFTP	DNS
WebDDoS	**87**	0	0	0	0	0	0	0	0	0	0	1
BENIGN	0	**11200**	0	1	0	1	0	0	0	0	0	3
UDP-lag	5	0	**10891**	0	84	112	49	0	10	1	51	2
NTP	23	0	0	**11127**	1	0	0	10	0	0	2	42
Syn	0	0	18	0	**11187**	0	0	0	0	0	0	0
SSDP	26	1	1	0	0	**10798**	0	4	90	237	1	47
UDP	0	0	0	0	1	24	**11077**	1	99	0	3	0
NetBIOS	2	0	0	3	0	9	8	**11109**	27	0	3	44
MSSQL	0	0	2	0	0	128	92	0	**10818**	42	122	1
SNMP	1	0	0	1	0	39	0	203	2	**10934**	0	25
TFTP	10	0	123	5	10	0	0	0	1	3	**11043**	10
DNS	21	0	0	33	5	43	1	64	2	5	9	**11022**

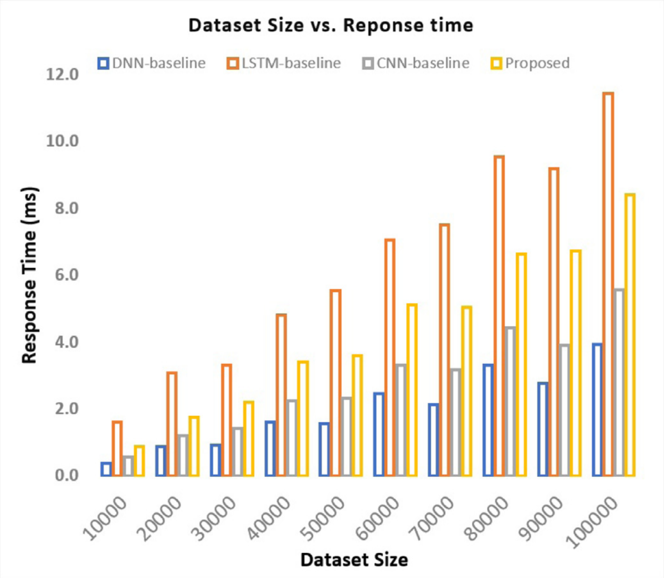


Fig. 10. Comparison of the response times vd. dataset size.

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$F_1 Score = \frac{TP}{TP + 1/2(FP + FN)} \tag{8}$$

Fig. 9 shows the training results using test datasets for all experimented networks. According to results the DNN based model produced about 96.77% accuracy. The CNN baseline model where the number of neurons are set to 64 for Conv1D layers produced accuracy about 98.34%. On the other hand LSTM baseline model with two Lstm layers produced 94.08% accuracy. CNN-based models produced more promising results even in the simplest case. Especially the proposed model where input features are decomposed into bytes produced best result with 99.30% for multi-class evaluation. Also we used the same model to obtain the binary classification result. In practice, it is desired to reduce the computation time of the detection model so that more packages can be analyzed per second. Hence analyzing the inference times of the candidate models help determine the real-time limitations of the networks. Fig. 10 shows the prediction times of the evaluated models. DNN and CNN models produce the detection results in shorter times compared to LSTM models. A model's total training time depends on the number of epochs and computational hardware. On the experimental hardware, the Dense, CNN, and LSTM based networks' training times per epoch take approximately 7 s, 11 s, and 35 s, respectively.

Tables 10–13 shows the confusion matrix for the proposed model, which is obtained using a test dataset. Table 14 compares the performance of the network to the state-of-the-art results. Our proposed model with inception like convolutional layers provides the best performances with approximately 99.30% accuracy. We replaced the last layer with a single output dense layer with a sigmoid activation function for binary class comparisons. According to the results, the proposed model produced 99.99% accuracy for the binary classification.

Table 12
Confusion matrix for LSTM based model.

	WebDDoS	BENIGN	UDP-lag	NTP	Syn	SSDP	UDP	NetBIOS	MSSQL	SNMP	TFTP	DNS
WebDDoS	**79**	1	2	2	0	0	0	0	0	0	0	4
BENIGN	8	**11146**	16	3	3	3	2	0	11	0	3	10
UDP-lag	5	7	**10787**	2	219	52	108	0	9	12	3	1
NTP	20	3	0	**11166**	0	9	0	1	0	0	0	6
Syn	0	2	215	0	**10984**	0	0	0	4	0	0	0
SSDP	23	5	13	18	0	**6769**	3317	1	256	616	3	184
UDP	0	2	1	2	0	107	**10902**	0	189	0	2	0
NetBIOS	5	2	0	4	0	9	5	**10536**	586	21	2	35
MSSQL	0	1	13	6	0	134	66	4	**10833**	64	23	61
SNMP	1	1	3	1	0	117	0	25	36	**10987**	0	34
TFTP	4	0	197	1	45	6	0	0	2	0	**10948**	2
DNS	18	5	0	16	1	152	0	12	79	11	2	**10909**

Table 13
Confusion matrix for the proposed CNN based model.

	WebDDoS	BENIGN	UDP-lag	NTP	Syn	SSDP	UDP	NetBIOS	MSSQL	SNMP	TFTP	DNS
WebDDoS	**87**	0	0	0	0	0	0	0	0	0	0	1
BENIGN	0	**11204**	0	0	0	0	0	0	1	0	0	0
UDP-lag	1	0	**10962**	0	0	88	58	0	0	1	95	0
NTP	17	1	0	**11183**	0	0	0	0	0	1	0	3
Syn	0	0	12	0	**11191**	0	0	0	0	0	0	2
SSDP	25	0	9	0	0	**11041**	1	1	53	71	0	4
UDP	0	0	2	1	0	10	**11118**	5	68	0	1	0
NetBIOS	3	0	1	0	0	10	0	**11164**	13	9	0	5
MSSQL	0	0	1	0	0	48	53	0	**11049**	48	6	0
SNMP	0	0	0	0	0	8	0	1	0	**11187**	1	8
TFTP	6	0	74	3	0	0	0	0	0	1	**11121**	0
DNS	13	0	0	1	0	2	0	13	0	2	0	**11174**

Table 14
The comparison of the proposed model to the state of the art results.

Author	Year	Area	Method	Accuracy	Classification Type	Remove Duplcation
Jia et al. (2020)	2020	IoT application	LSTM	98.9%	Binary	No
Li et al. (2020)	2020	IoT application	LSTM	Not Available	Multi (7)	No
de Assis et al. (2020)	2020	SDN environments in IoT networks	CNN	95.4%	Binary	No
Alamri and Thayananthan (2020)	2020	SDN environments in IoT networks	Extreme gradient boosting algorithm	91.26%	Multi (11)	No
Pontes et al. (2021)	2021	Not Available	Energy-based flow classifier	98.1%	Binary	No
Assis et al. (2021a)	2021	SDN environments	Gated Recurrent	99%	Binary	No
Javeed et al. (2021)	2021	SDN environments in IoT networks	LSTM and GRU	99.74%	Multi (8)	No
Nie et al. (2021)	2021	IoT application	Generative adversarial network	98.35%	Multi (8)	No
Ferrag et al. (2021)	2021	Cloud Computing,NFV, and SDN	CNN, RNN, DNN	95.12%, 94.88%, 93.88%	Multi (13)	Yes
Our Model (Multi)	2021	IoT application, Cloud Computing	CNN	**99.30**%	Multi (12)	Yes
Our Model (Binary)	2021	IoT application, Cloud Computing	CNN	**99.99**%	Binary	Yes

6. Conclusion

In this paper, we provide a new intrusion detection system that is based on deep learning models for DDoS attacks. We used CIC-DDoS 2019 dataset, which contains 12 classes, including a benign class. We tested various deep learning models such as DNN, CNN, and LSTM for various units per layer. We also improved the system using preprocessing techniques such as feature elimination and selection where we selected 40 important features out of 88. We obtained a new homogeneous data set by selecting an equal number of samples from each attack type with random subset selection. Afterward, we removed duplicate records to obtain a clean, non-repetitive data set which most relevant studies ignored. In this respect, this study presents two new data sets to the literature that directly affect the performance of the training processes produced from the CIC-DDoS 2019 data set. Finally, we applied min-max normalization processes to examine their effect on the performance. Therefore we produced applicable data by obtaining a normalized set containing an equal number of preprocessed samples from each attack type without duplicates. According to the test results, the suggested model achieved accuracy values of 99.30% for multi class and 99.99% for binary class as can be seen in Table 16. We observed that proposed CNN-based method which contains one-dimensional convolution layers achieve more successful results and have higher accuracy than many current studies as shown in Table 14. In addition, we tested real-time packet processing speed on the examined models. The proposed model has a reasonable inference time when compared with the baseline models. As a result, the proposed model successfully detects various types of attacks for multi-class and binary-class classification.

Data Availability

The CIC-DDoS2019 dataset, which was utilized to back up the study's conclusions, is available at: https://www.unb.ca/cic/datasets/ddos-2019.html

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Devrim Akgun: Conceptualization, Software, Investigation, Writing – review & editing, Supervision, Project administration. **Selman Hizal:** Conceptualization, Validation, Data curation, Writing – review & editing, Software. **Unal Cavusoglu:** Methodology, Validation, Investigation, Writing – review & editing, Software.

References

Al Janabi, K., Kadhim, R., 2018. Data reduction techniques: a comparative study for attribute selection methods. Int. J. Adv. Comput. Sci. Technol. 8 (1), 1–13.

Alamri, H.A., Thayananthan, V., 2020. Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks. IEEE Access 8, 194269–194288. doi:10.1109/ACCESS.2020.3033942.

Amaizu, G., Nwakanma, C., Bhardwaj, S., Lee, J., Kim, D., 2021. Composite and efficient DDoS attack detection framework for B5G networks. Comput. Netw. 188, 107871. doi:10.1016/j.comnet.2021.107871.

Assis, M.V., Carvalho, L.F., Lloret, J., Proença, M.L., 2021. A GRU deep learning system against attacks in software defined networks. J. Netw. Comput. Appl. 177, 102942. doi:10.1016/j.jnca.2020.102942.

de Assis, M.V., Carvalho, L.F., Rodrigues, J.J., Lloret, J., Proença Jr, M.L., 2020. Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. Comput. Electr. Eng. 86, 106738. doi:10.1016/j.compeleceng.2020.106738.

Babić, I., Miljković, A., Čabarkapa, M., Nikolić, V., Aorević, A., Ranelović, M., Ranelović, D., 2021. Triple modular redundancy optimization for threshold determination in intrusion detection systems. Symmetry 13 (4), 557. doi:10.3390/sym13040557.

Cil, A.E., Yildiz, K., Buldu, A., 2021. Detection of DDoS attacks with feed forward based deep neural network model. Expert Syst. Appl. 169, 114520. doi:10.1016/j.eswa.2020.114520.

Di Mauro, M., Galatro, G., Fortino, G., Liotta, A., 2021. Supervised feature selection techniques in network intrusion detection: a critical review. Eng. Appl. Artif. Intell. 101, 104216.

Di Mauro, M., Galatro, G., Liotta, A., 2020. Experimental review of neural-based approaches for network intrusion management. IEEE Trans. Netw. Serv. Manage. 17 (4), 2480–2495.

Elsayed, M.S., Le-Khac, N.-A., Dev, S., Jurcut, A.D., 2020. DDoSNet: a deep-learning model for detecting network attacks. In: 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM). IEEE, pp. 391–396. doi:10.1109/WoWMoM49955.2020.00072.

Ferrag, M.A., Shu, L., Djallel, H., Choo, K.-K.R., 2021. Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. Electronics 10 (11), 1257. doi:10.3390/electronics10111257.

Ge, M., Syed, N.F., Fu, X., Baig, Z., Robles-Kelly, A., 2021. Towards a deep learning–driven intrusion detection approach for internet of things. Comput. Netw. 186, 107784.

Gupta, N., Jindal, V., Bedi, P., 2021. LIO-IDS: handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system. Comput. Netw. 192, 108076.

Han, J., Pei, J., Kamber, M., 2011. Data Mining: Concepts and Techniques. Elsevier.

Hussain, F., Abbas, S.G., Husnain, M., Fayyaz, U.U., Shahzad, F., Shah, G.A., 2020. IoT DoS and DDoS attack detection using ResNet. In: 2020 IEEE 23rd International Multitopic Conference (INMIC). IEEE, pp. 1–6. doi:10.1109/INMIC50486.2020.9318216.

Javeed, D., Gao, T., Khan, M.T., 2021. SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT. Electronics 10 (8), 918. doi:10.3390/electronics10080918.

Jia, Y., Zhong, F., Alrawais, A., Gong, B., Cheng, X., 2020. FlowGuard: an intelligent edge defense mechanism against IoT DDoS attacks. IEEE Internet Things J. 7 (10), 9552–9562. doi:10.1109/JIOT.2020.2993782.

Kamalov, F., Moussa, S., El Khatib, Z., Mnaouer, A.B., 2021. Orthogonal variance-based feature selection for intrusion detection systems. In: 2021 International Symposium on Networks, Computers and Communications (ISNCC). IEEE, pp. 1–5.

Kasim, Ö., 2020. An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks. Comput. Netw. 180, 107390.

Khempetch, T., Wuttidittachotti, P., 2021. DDoS attack detection using deep learning. IAES Int. J. Artif. Intell. (IJ-AI) 10 (2), 382. doi:10.11591/ijai.v10.i2.pp382-388.

Kozik, R., Choraś, M., Ficco, M., Palmieri, F., 2018. A scalable distributed machine learning approach for attack detection in edge computing environments. J. Parallel Distrib. Comput. 119, 18–26.

Lashkari, A. H., 2021. Cicflowmeter-v3.0. https://github.com/ahlashkari/CICFlowMeter.

Li, J., Liu, M., Xue, Z., Fan, X., He, X., 2020. RTVD: a real-time volumetric detection scheme for DDoS in the internet of things. IEEE Access 8, 36191–36201. doi:10.1109/ACCESS.2020.2974293.

Martinez, V., Salas, R., Tessini, O., Torres, R., 2021. Machine learning techniques for behavioral feature selection in network intrusion detection systems.

Nashat, D., Hussain, F.A., 2021. Multifractal detrended fluctuation analysis based detection for SYN flooding attack. Comput. Secur. 107, 102315. doi:10.1016/j.cose.2021.102315.

Nie, L., Wu, Y., Wang, X., Guo, L., Wang, G., Gao, X., Li, S., 2021. Intrusion detection for secure social internet of things based on collaborative edge computing: A Generative adversarial network-Based approach. IEEE Trans. Comput. Social Syst. 1–12. doi:10.1109/TCSS.2021.3063538.

Odumuyiwa, V., Alabi, R., 2021. DDOS detection on internet of things using unsupervised algorithms. J. Cyber Secur. Mobility doi:10.13052/jcsm2245-1439.1034.

Pontes, C.F.T., de Souza, M.M.C., Gondim, J.J.C., Bishop, M., Marotta, M.A., 2021. A new method for flow-based network intrusion detection using the inverse potts model. IEEE Trans. Netw. Serv. Manage. 18 (2), 1125–1136. doi:10.1109/TNSM.2021.3075503.

Rajagopal, S., Kundapur, P.P., K. S, H., 2021. Towards effective network intrusion detection: from concept to creation on azure cloud. IEEE Access 9, 19723–19742. doi:10.1109/ACCESS.2021.3054688.

Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A., 2019. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST). IEEE, pp. 1–8. doi:10.1109/CCST.2019.8888419.

Shieh, C.-S., Lin, W.-W., Nguyen, T.-T., Chen, C.-H., Horng, M.-F., Miu, D., 2021. Detection of unknown DDoS attacks with deep learning and Gaussian mixture model. Appl. Sci. 11 (11), 5213. doi:10.3390/app11115213.

Shurman, M., Khrais, R., Yateem, A., 2020. DoS and DDoS attack detection using deep learning and IDS. Int. Arab J. Inf.Technol. 17 (4A), 655–661. doi:10.34028/iajit/17/4A/10.

Source, O., 2021a. Keras v2.5. https://keras.io/.

Source, O., 2021b. Phyton programming language v3.8. https://www.python.org/.

Source, O., 2021c. Ubuntu operating system v20.04. https://ubuntu.com/download/desktop?version=20.04&architecture=amd64.

Sourceforge, W., 2022. Class infogainattributeeval. https://weka.sourceforge.io/doc.dev/weka/attributeSelection/InfoGainAttributeEval.html.

Vuong, T.-H., Thi, C.-V.N., Ha, Q.-T., 2021. N-tier machine learning-based architecture for DDoS attack detection, pp. 375–385. doi:10.1007/978-3-030-73280-6_30.

Wei, Y., Jang-Jaccard, J., Sabrina, F., Singh, A., Xu, W., Camtepe, S., 2021. AE-MLP: a hybrid deep learning approach for DDoS detection and classification. IEEE Access 9, 146810–146821.

Zhong, W., Yu, N., Ai, C., 2020. Applying big data based deep learning system to intrusion detection. Big Data Min. Anal. 3 (3), 181–195.

Devrim Akgun received his PhD degree in the Department of Electrics-Electronics Engineering from Sakarya University in 2008. He is currently working as an associate professor at Sakarya University of Software Engineering department. His current research interests include deep learning, GPU programming, and parallel programming.

Selman Hizal received PhD in the Department of Electrical and Electronics Engineering, Sakarya University. His research topics include software engineering, information systems, communications, parallel and distributed simulation. His main research interest lies in parallel and distributed simulation and routing protocols for wireless Ad Hoc networks.

Unal Cavusoglu received PhD in the Department of Electrical and Electronics Engineering, Sakarya University. His research topics include software engineering, information systems, communications, distributed systems management and Wireless Ad Hoc networks.