Int. J. Nonlinear Anal. Appl. 13 (2022) 2, 753–762
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6458

Survey on distributed denial of service attack detection using deep learning: A review

Manal Dawood Jassem[a,*], Amer Abdulmajeed Abdulrahman[a]

[a]Department of Computer Science, College of Science, University of Baghdad, Iraq

(Communicated by Madjid Eshaghi Gordji)

Abstract

Distributed Denial of Service (DDoS) attacks on Web-based services have grown in both number and sophistication with the rise of advanced wireless technology and modern computing paradigms. Detecting these attacks in the sea of communication packets is very important. There were a lot of DDoS attacks that were directed at the network and transport layers at first. During the past few years, attackers have changed their strategies to try to get into the application layer. The application layer attacks could be more harmful and stealthier because the attack traffic and the normal traffic flows cannot be told apart. Distributed attacks are hard to fight because they can affect real computing resources as well as network bandwidth. DDoS attacks can also be made with smart devices that connect to the Internet, which can be infected and used as botnets. They use Deep Learning (D.L.) techniques like Convolutional Neural Network (C.N.N.) and variants of Recurrent Neural Networks (R.N.N.), such as Long Short-Term Memory (L.S.T.M.), Bidirectional L.S.T.M., Stacked L.S.T.M., and the Gat G.R.U.. These techniques have been used to detect (DDoS) attacks. The Portmap.csv file from the most recent DDoS dataset, CICDDoS2019, has been used to test D.L. approaches. Before giving the data to the D.L. approaches, the data is cleaned up. The pre-processed dataset is used to train and test the D.L. approaches. In the paper, we show how the D.L. approach works with multiple models and how they compare to each other.

Keywords: Deep Learning, Convolutional Neural Network, Recurrent Neural Network, Artificial Neural Network, Gated Recurrent Unit, Long Short-Term Memory
2020 MSC: 68T07

1 Introduction

One of the issues is security concerns stemming from the usage of the internet and computer systems. Security has been attacked in a variety of ways. (DDoS) [17] is one of the most popular internet attacks because of limitations on the attacked device, such as memory or bandwidth. To allow people to connect, the victims must first access the system. Cyber attackers employ these channels to exhaust the victims' resources to the point where they can no longer be utilized. The victim's gadgets are then rendered inoperable and unable to serve users. A computer network's abnormalities may be classified using computer network traffic logs, which contain both normal data and varied network attack data. There are several features in each attack feature that may be utilized to detect abnormalities in the system.

*Corresponding author
Email addresses: manaldawd85@gmail.com (Manal Dawood Jassem), amer.abdulrahman@sc.uobaghdad.edu.iq (Amer Abdulmajeed Abdulrahman)

Received: February 2022 *Accepted:* April 2022

To launch a DDoS attack on a system, two essential stages are required. The first phase is an attacker sending malicious packets to victims' workstations in order to disrupt protocols or operating programs, i.e., a vulnerability attack that results in zombies [19]. Typically, Trojan horses, backdoors, or worms are employed to recruit zombies [15, 18]. The attacker then employs these zombies to launch flooding attacks by depleting server or network resources such as bandwidth, memory, router processing capacity, and disk/database space [8]. The DDoS attack interrupts the attacked system as well as the services that the system provides to legitimate users. DDoS attacks are conducted using a network of zombie botnet computers that are remotely controlled, well-organized, and extensively distributed. Many traffic or service requests are sent to the target system at the same time or in a continuous stream. Because of the attack, the target system becomes useless, responds slowly, or crashes completely [13].

The defense methods have a difficult time identifying the original attackers because the attackers have spoofed IP addresses and are hidden among the zombies under their control [19]. In 2009, several zombies were utilized to overwhelm a target via a DDoS attack, causing network services for big websites such as Facebook, Live Journal, Twitter, and Amazon to be disrupted [2].

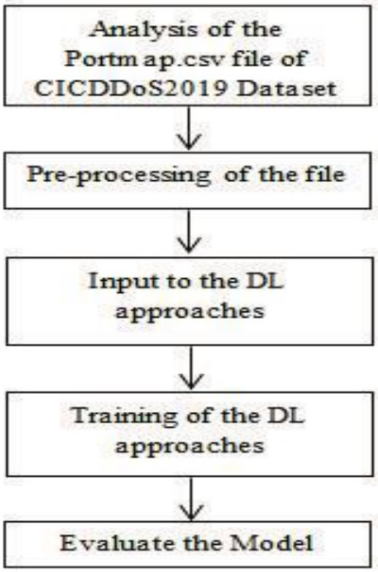

Figure 1: The steps of detection are shown in [37]

2 Background

To properly comprehend a DDoS attack, we must first understand the attack's roots and the historical significance of these occurrences. A denial-of-service attack, according to [28] "It interrupts or degrades network services (by depleting network bandwidth or router processing capacity) or victim resources (by draining disk or database bandwidth, file descriptors, buffers, sockets, CPU cycles, RAM) and prevents legitimate users from accessing a certain Internet service. The DDoS attack, like the DDOS attack, seeks to utilize many devices to impede or halt the user's connection or to take a service down. The use of various forms of botnets has allowed DDoS to develop in capabilities and take down larger targets. These sorts of attacks may happen to anyone, regardless of how prominent a user's reach in technology is. Denial of service (DoS) attacks, which take advantage of sluggish equipment, are widely encountered in the gaming industry on a single-person experience. Where a DDoS-attack has effectively brought down significant organizations. Such attacks have rendered thousands of beneficiaries unable to connect to the internet until mitigation or rectification procedures have been implemented to assist their connections [28].

3 DDOS method

DDoS attacks have evolved into a global threat to today's Internet. These attacks are deft in nature and employ the same strategies as ordinary DoS attacks, with the difference being that the former is carried out on a larger scale than the latter via botnets [10]. A botnet chain is made up of hundreds or thousands of compromised (bots, zombies, or slave agents) that are controlled remotely by one or more attackers assaulting a victim. For attackers, any computer

linked to the Internet represents an appealing chance to generate zombies, often without the users' awareness. Zombies are recruited via worms, Trojan horses, or backdoors by offering an enticing URL, e-mail content, or a trustworthy sender address to susceptible PCs [25].

In general, an individual attacker or a gang of attackers uses various hacking techniques to exploit the vulnerabilities and weaknesses of computer computers linked to the Internet. As a result, malicious code is planted on these computers, putting them in a vulnerable position and gaining control of them [25]. Some of these devices are set up as "handlers," while others are set up as "zombies." The handlers are controlled by the attackers, while the zombies are controlled by the handlers' software. Before launching the attack, the attackers seek to get control of as many computer machines as possible. The number of zombies may number in the hundreds or perhaps thousands. As demonstrated in Figure 2, massive groups of zombies construct a "botnet" of attacks one after the other. The size of the botnet dictates the degree and extent of the attack. A massive botnet launches devastating and severe attacks [16]. A single zombie only supplies a tiny quantity of information. The aggregate traffic from several zombies that appear on end users' computers is massive, and so exhausts resources. Low-rate DDoS attacks are risky and difficult to detect since the traffic that may be controlled by a specific link appears regular [39]. As a result, current detection methods may result in a quick surge in high-rate DDoS attacks. DDoS attacks are now deployed in the form of link and packet flooding. These attacks are classified as Net DDoS and App-DDoS flooding attacks based on the protocol level that is affected. DDoS attacks must be classified correctly in order to be identified. Figure 3 depicts a classification of DDoS attack methods.

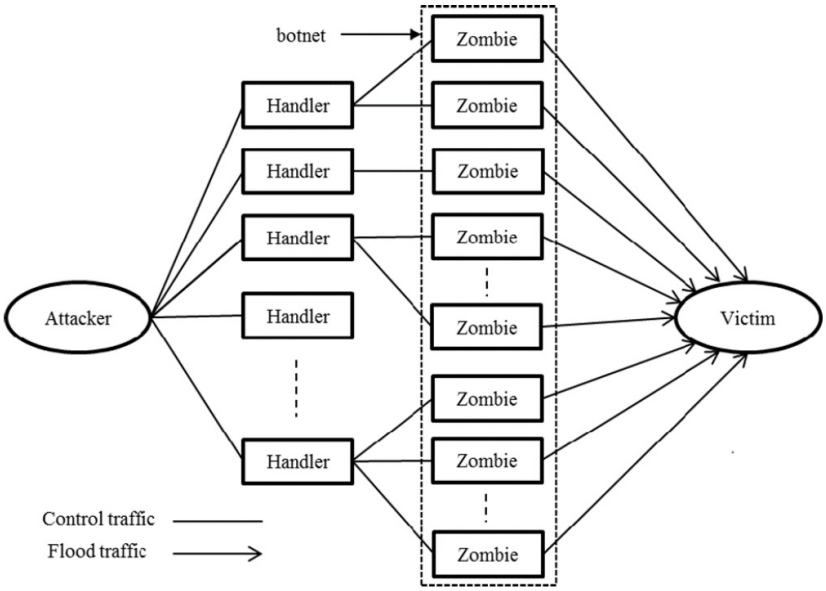

Figure 2: The DDos attack architecture [4]

4 DL-based DDOS detection approaches

A wide range of techniques based on (D.L.) have found tremendous success in a variety of applications, including face recognition, image processing, and natural language translation. D.L. can extract raw features from data without the need for human interaction. It can meet the high-performance rate by automatically detecting correlations in raw data. As a result, with the introduction of D.L.-based models, the accuracy in identifying attacks has risen even more. However, network traffic temporal correlations frequently yield time-series data [32] and training the simplest form of D.L. algorithms with sequential traffic can result in data loss. For our suggested solution, we employ the R.N.N. approach to cover this problem and avoid any loss. At every input stage, the R.N.N. compares prior computations to the current events. A D.L. strategy based on R.N.N.-autoencoder for the detection of DDoS attacks on Software-defined networking may maintain all data information with little loss while training the model with such methods (S.D.N.). When compared to several classical methodologies, the suggested model has the greatest performance in terms of precision, recall, F1-score, and accuracy.

The description steps are followed:

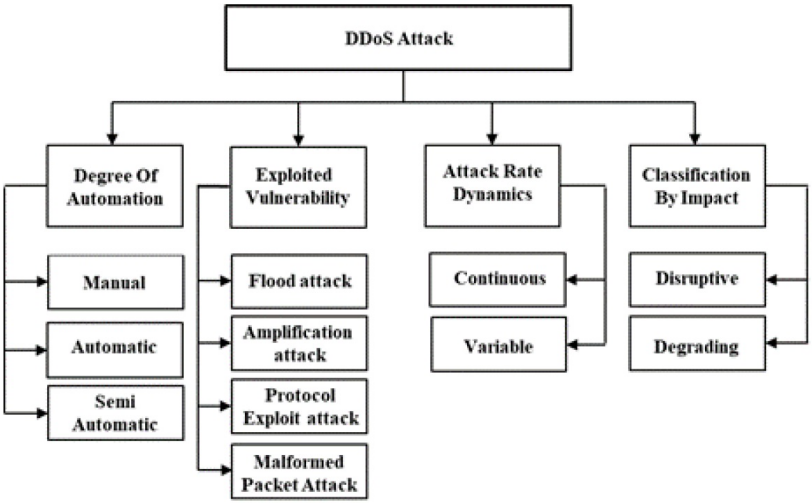

Figure 3: The classification of the DDoS attack [4]

1. Examination of the CICDDoS2019 dataset's Portmap.csv file: The Portmap.csv file was examined to check if it included Non-a-Number (N.a.N.), null, infinite, numerical, binary, or categorical data. That data is then transformed into the same format.
2. File pre-processing: The N.a.N. and infinity values are removed, and the numerical values are normalized if necessary. Label Encoder is used to transform the class values.
3. Pre-processed input to the D.L. techniques: The pre-processed input is fed into the D.L. approaches. The Conv1D has been utilized to transform the input according to [37] and deliver it to the R.N.N.-based techniques and the input to the C.N.N..
4. D.L. approach training: The D.L.-based methods were trained utilizing 80 percent of the Portmap.csv entries.
5. Test the model: The trained model has been tested on 20 percent of the records.

The description of D.L.-based DDoS detection approaches is described as follows:

Conventional feed-forward neural networks have a limitation that data is considered independent [24]. Figure 4 depicts the R.N.N.'s construction.

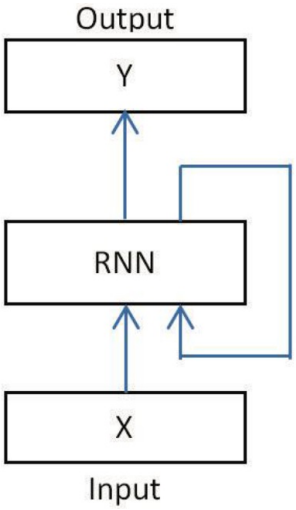

Figure 4: Structure of R.N.N. [24]

It's a little hard for R.N.N. to remember long-term memories [12]. It means that it doesn't work well when there are a lot of long sequences [22]. R.N.N. also has vanishing gradient and exploding gradient, which make it useless. All of these problems make it impossible to use [35]. If the multiplication of derivatives of "n" hidden layers in a

network is bigger, then the gradient will grow exponentially, or it will explode as we get closer to the beginning layers. This is defined the "exploding gradient" [26]. As we get closer to the beginning layers, the gradient will go away or be completely gone if the derivatives are very small [26]. L.S.T.M. Networks, a type of R.N.N., can help solve these problems with R.N.N.s, such as vanishing gradients and short term memory [3].

1. Long Architecture Short-Term Memory (L.S.T.M.) Network
   As defined in [24], an L.S.T.M. network is composed of distinct memory blocks referred to as cells. Figure 5 illustrates the L.S.T.M. structure. Each cell accepts the following three inputs:

   - Previous condition of the cell.
   - Previous concealed state.
   - The current time step's input.

   Following three inputs, a cell generates two states or outputs for the following cell, namely the cell state and the hidden state. Through memory blocks, L.S.T.M.s may choose which information to remember or discard. Changes to the memory are made using three mechanisms known as gates, including the forget gate, the input gate, and the output gate. The cell state and its gates are at the heart of L.S.T.M.s:

   - Forget Gate:
     The forget gate is responsible for forgetting or retaining information from the cell state. The critical information is maintained, while the irrelevant data is deleted. This gate accepts two inputs, namely the previous concealed state and the cell's current input. The sigmoid function within the forget gate returns a value of 0 or 1 depending on which information the forget gate wishes to forget about a cell state and which information it wishes to remember. This is accomplished through the use of a multiplication filter situated between the cell state and the forget gate. The mathematical expression for it is given in table 1 equation (4.1).

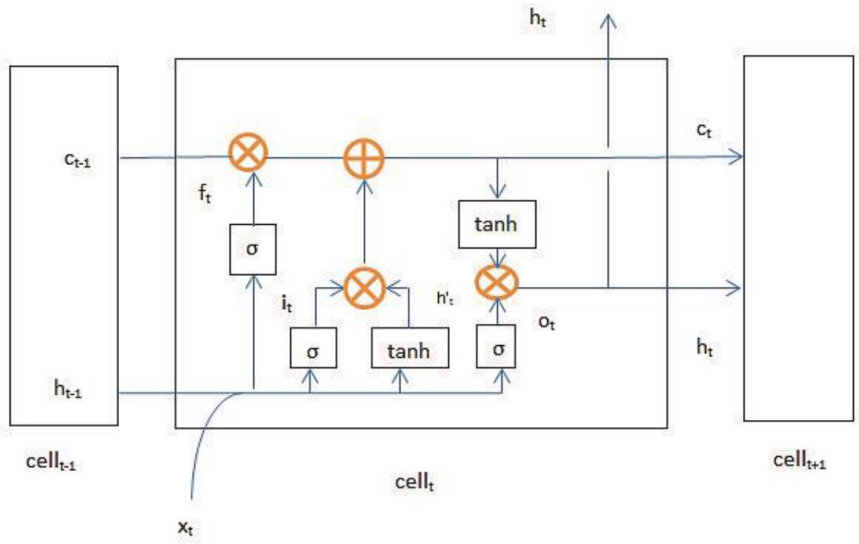

Figure 5: Structure of L.S.T.M. [24]

   - Input Gate:
     The input gate can be used to introduce fresh information to the cell state [24]. This procedure of adding new information has three-steps and may be summarized as follows:
     Step 1: A sigmoid function is used to determine how much value should be added to the cell state.
     Step 2: The tank function returns a vector containing all potential values for the cell state.
     Step 3: The product of steps 1 and 2 is then added to the current state of the cell.
     Equation illustrates the mathematical equation for this (4.2) show in table 1.
   - Output Gate:
     The output gate's job is to extract relevant information from the current cell state and output it [24]. This may be accomplished in three easy steps:

Step 1: The tank method represents the current state of the cell as an input and returns a vector.

Step 2: The sigmoid function is used to the previous concealed state's values and the current input to control the values that must be produced from the vector formed in step 1.

Step 3: Multiplying the values from steps 1 and 2, this output is used as the cell's output and also serves as the hidden state for the following cell. The mathematical expression for it is given in table 1 equation (4.4).

- Cell State:
  This is utilized by the whole network within the L.S.T.M. cell (the mathematical equation for this is presented in table 1 equation (4.5) and information is retained or deleted using gates [30]:

Table 1: L.S.T.M. equation

Equations	
$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$	(4.1)
$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$	(4.2)
$c_t^{\sim} = \tanh(W_c[h_{t-1}, x_t] + b_c)$	(4.3)
$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$	(4.4)
$c_t = f_t * c_t + i_t * c_t^{\sim}$	(4.5)
$h_t = o_t * \tanh(c_t)$	(4.6)

The current is denoted by x_t, the output or hidden state is denoted by h_t, the cell state is denoted by c_t, the forgotten gate is denoted by f_t, the output gate is denoted by O_t, the input gate is denoted by I_t and the candidate state is denoted by c. W and b denote weight matrices and bias, respectively.

2. Bidirectional long short-term memory architecture
   The Bidirectional L.S.T.M. was employed to enhance the detection model's performance. As seen in Figure 6, Bi-L.S.T.M.s train two L.S.T.M.s, the first on the input sequence and the second on a reversed version of the input sequence [11].

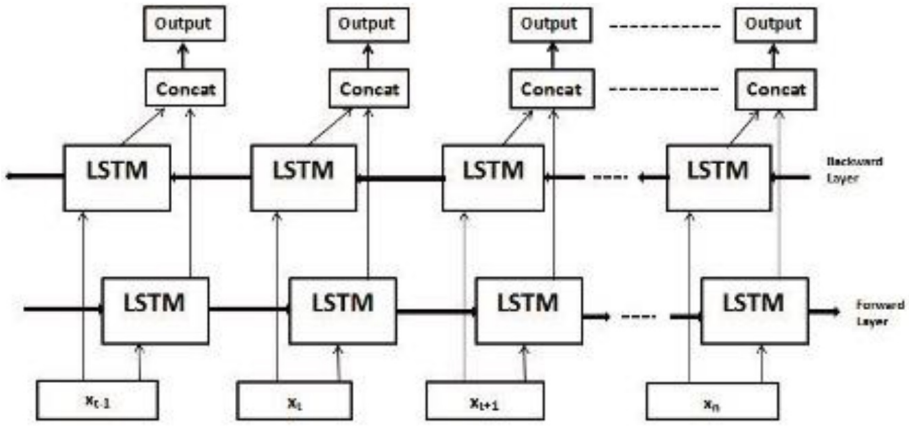

Figure 6: Bidirectional LSTM Architecture [24].

3. Stacked L.S.T.M. Architecture:
   Figure 7 shows that the Stacked-L.S.T.M. architecture has multiple L.S.T.M. layers [5]. One L.S.T.M. layer is stacked on top of the other L.S.T.M. layer. In the second L.S.T.M. layer, the output of the first layer is the sequence of output that the second layer is going to use as input.
4. Gated Recurrent Units:
   Additionally, the G.R.U. is a sort of R.N.N., and this type of R.N.N. is thought to be good and quick due to the fact that it requires less network parameters [4]. G.R.U.s combine forget and input gates into a single update gate and integrate the cell state and hidden state, as well as some additional alterations.

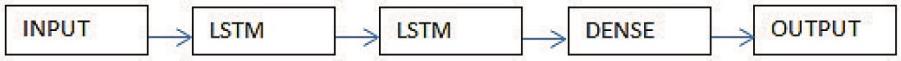

Figure 7: Stacked L.S.T.M. [24].

5. Convolutional Neural Network:
   Convolutional Neural Network (C.N.N.s) are similar to multiple-layer feed-forward neural networks. Convolutional layers are the building pieces of C.N.N. [33], which are stacked on top of one another to enable hierarchical learning of features. Convolutional layers are followed by activation layers and, in certain cases, pooling layers in a C.N.N.. C.N.N. is described in full in [7].

5 Dataset

In our studies, we use two well-known datasets: CIC-IDS2017 [20] and CIC-IDDoS2019 [31]. The Canadian Institute for Cybersecurity gathered the datasets using Wireshark in simulated settings. They are formed through the use of two distinct types of user profiles and multistage attacks such as Heartbleed, as well as a variety of DoS and DoS attacks. The CIC-Flowmeter is then used to pre-process the collected stream [31]. It has 80 network traffic features and is designed to create a variety of Dos and DoS traffic data. The generated data collection is in CSV format and comprises records of traffic features. To conform to the proposed framework's numeric nature, the non-numeric fields are changed using One-Hot encoding. After that, all fields are normalized for the purpose of rescaling their dynamic ranges.

6 Related work

In order to categorize the DDoS attack data from 4,986 records, researchers specified the number of hidden layers of the network between 30-55 layers. DNS DDoS assault, CharGen DDoS attack [29], UDP DDoS attack, and Normal were the four categories used to classify records. The results showed that an artificial neural network (A.N.N.) with 50 hidden layers can correctly recognize DDoS data 95.6 percent of the time.

Hsieh and Chan [6] employed a neural network with the Apache Spark framework for DDos detection, which has been used to manage large-scale data (Big Data) and function as a cluster. Number of Packets, Average Packet Size, Time Interval Variance, Packet Size Variance, Number of Bytes, Packet Rate, and Bit Rate were all tested in the ARPA 2000 LLDOS 1.0 series, which contains seven unique features. Normal data and assault data were used to categorize all of the data. There were 51,040 normal data points and 74,480 assault data points in total. All data samples were split into two sections, with 30 percent going to the learning series and 70 percent going to the test data. It was discovered that the accuracy rate was 94 percent.

Devaraju and Ramakrishnan [9] put three artificial neural networks to the test, each with three different ways. There were three types of neural networks: feed forward neural network (F.F.N.N.), probabilistic neural network (P.N.N.), and radial basis neural network (R.B.N.N.) (R.B.N.N.). The methodologies were utilized to evaluate the intrusion detection system's efficacy using the knowledge-discovery in databases (K.D.D.) Cup 1999 dataset [**?**], which has 41 unique features. There are four sorts of attacks: 1) DoS, which includes back, land, neptune, pod, smurf, and teardrop; 2) DoS, which includes back, land, neptune, pod, smurf, and teardrop; and 3) DoS, which includes back, land, 2) Ftp write, guess passwd, imap, multihop, phf, spy, warez-client, and warez-master are included in the Remote to Local (R2L) package. 3) User-to-Root (U2R), which includes buffer overflow, load-module, perl, and rootkit; and 4) probing, which includes ip sweep, nmap, port sweep, and satan. The information was separated into seven data. Normal class, smurf class, neptune class, saint class, mail bomb class, Apache class, and devil class were the several types of classes. The experiment was split into two parts: training and testing. There are 700 data points in each batch. The P.N.N. network was found to be the most effective based on the data of the experiments. The accuracy rate for the P.N.N., F.F.N.N., and R.B.N.N. neural networks was 97.5 percent, 94.3 percent, and 65 percent, respectively. Different machine learning approaches, such as deep learning, are being used by researchers. They classified network attack data using A.N.N., S.V.M., and A.N.N.+S.V.M. algorithms on the dataset NSL K.D.D. in [23]. The attack was separated into two classes in this experiment: 58,630 attack classes and 67,343 regular classes [38]. The respective accuracy rates were 79.56, 79.27, and 79.71 percent.

To detect DDoS attacks in the S.D.N. network, Ye et al. [36] employed the Support Vector Machine (S.V.M.) classification technique. For the training stage, the researcher used six tuple features that may be obtained from the

S.D.N. controller. The dataset samples are gathered by simulating an S.D.N. network with five virtual hosts using Mininet and flood controller. During the simulation phase, three separate DDoS scenarios are simulated, including UDP, TCP SYN, and ICMP flood traffic.

In the context of S.D.N., Rahman et al. [27] used four distinct machine learning approaches to identify DDoS attacks. To build the Training and Testing Dataset, the researchers simulated both normal and malicious traffic. The hping3 utility is used to produce two DDoS samples (TCP and ICMP floods). The findings of the trial revealed that the J48 is more accurate than the other approaches tested.

To identify flow-table overflow attacks inside the S.D.N. data plane, Abhiroop et al. [1] employed three distinct machine learning algorithms: S.V.M., Naive Bayes (N.B.), and Neural Network. To produce training data, the researchers used the open flow protocol to extract tuple features from open flow switches. The Scapy utility is used to produce three types of flood traffic: TCP, UDP, and ICMP. Five features are used in machine learning approaches, and the findings demonstrate that the S.V.M. has a lower accuracy rate than the other two classifiers.

In [14] the researchers presented two detection models for DDoS attacks on S.D.N. networks. The signature-based snort detection system was employed to gather network traffic in the first stage. S.V.M. and Deep Neural Network (DNN) algorithms are used for attack classification in the final stage. The researchers trained the two detection modules using the KDDCUP'99 dataset, which had a total of 41 features. With an accuracy rate of 92.30 percent and 74.30 percent, respectively, the DNN outperforms the S.V.M. in the trial.

Mohammed et al. [21] suggested a novel architecture for DDoS attack detection on S.D.N. . The NSL-K.D.D. dataset was used to train the N.B. classifier using 25 features. The researchers combine three distinct selection algorithms (Genetic, Ranker, and Greedy) to choose the dataset's combined features. Precision, recall, and F1-score had average values of 0.81, 0.77, and 0.77, respectively.

The majority of detection algorithms in the literature that simulated the S.D.N. network to produce the DDoS attacks dataset only evaluate a small number of malicious activities, just for IP or TCP protocols, and ignore any application layer DDoS attacks. The great resemblance of attacks and benign actions is one of the issues in detecting application layer DDoS attacks. As a result, there are few features available to define such attacks, and many detection systems are unable to detect them [34]. In addition, technologies such as Scapy and Hping3 are used to produce the simulated traffic. As a result, the generated dataset is limited and does not contain all of the traffic required to obtain reliable findings. Furthermore, present strategies for training anomaly detection systems utilizing public datasets have a number of flaws. Most databases, for example, are out of date and do not include fresh attacks or attack traffic. Furthermore, they offer only a few forms of attacks to meet all current Internet trends. The evaluation of detection algorithms and approaches systems is greatly influenced by a large and valid dataset. To test our proposed model, we used the most recent publicly accessible dataset, CICDDoS2019 [31], which contains a wide range of DDoS attacks and fills in the gaps in previous datasets.

Table 2: Compare table for previous work

Researcher	**Dataset**	**Algorithm**	**Accuracy**
CharGen et al. [29]	K.D.D. CUP 1999 dataset and NSL-K.D.D.	A.N.N.	95.6%
Hsieh and Chan et al. [6]	CICDDoS2019	neural network	94%
Rahman et al. [27]	CIC IDS 2017 dataset	four machine learning approaches	N/A
Abhiroop et al. [1]	K.D.D. CUP 1999 and NSL-K.D.D.	S.V.M., Naive Bayes and Neural Network	N/A
Mohammed et al. [21]	CICDDoS2019	Naive Bayes	77%
Scapy et al. [31]	CICDDoS2019	machine learning approaches and S.V.M	74%

7 Conclusion

DDoS is increasing at a breakneck pace, and academics have developed several ways for detecting DDoS attacks, including statistical, machine learning, and deep learning methods. However, several of the approaches have drawbacks. Thus, D.L. techniques such as R.N.N. (L.S.T.M., stacked L.S.T.M., bidirectional, and G.R.U.s) and C.N.N. were utilized to identify DDoS attacks. The Portmap.csv file from the CICDDoS2019 dataset was analyzed using Stacked

L.S.T.M., which resulted in a higher testing accuracy of 99.55 percent due to the training of two L.S.T.M.s. We will analyze R.N.N. and C.N.N. on the whole CICDDoS2019 dataset in future work.

References

[1] T. Abhiroop, S. Babu and B. Manoj, *A machine learning approach for detecting DoS attacks in SDN switches*, Proc. Twenty Fourth Nat. Conf. Commun., 2018, pp. 1–6.

[2] B. Acohido and J. Swartz, *ABC News Live Journal New York*, https://abcnews.go.com//story?id=8271907&page=1, (2009).

[3] E. Alese, *The curious case of the vanishing & exploding gradient*, https://medium.com/learn-love-ai/the-curious-case-of-the-vanishing-explodinggradient-bf58ec6822eb, 2018.

[4] M.Z. Alom, *The history began from alexnet: A comprehensive survey on deep learning approaches*, arXiv preprint arXiv:1803.01164, (2018).

[5] J. Brownlee, *Stacked long short-term memory networks*, https://machinelearningmastery.com/stacked-long-short-term-memory-networks, 10 (2017), p. 2019.

[6] C.H.T. Chan, *Detection DDoS attacks based on neural-network using apache spark*, Int. Conf. Appl. Syst. Innov., IEEE, 2016, pp. 1–4.

[7] S.K. Dasari and V. Prasad, *A novel and proposed comprehensive methodology using deep convolutional neural networks for flue cured tobacco leaves classification*, Int. J. Inf. Technol. **11** (2019), no. 1, 107–117.

[8] H. D'Cruze, P. Wang, R.O. Sbeit and A. Ray, *A software-defined networking (SDN) approach to mitigating DDoS attacks*, Inf. Technol. New Gener. 2018, pp. 141–145.

[9] S. Devaraju and S. Ramakrishnan, *Performance analysis of intrusion detection system using various neural network classifiers*, Int. Conf. Recent Trends Inf. Technol., IEEE, 2011, pp. 1033–1038.

[10] C. Douligeris and A. Mitrokotsa, *DDoS attacks and defense mechanisms: Classification and state-of-the-art*, Comput. Netw. **44** (2004), no. 5, 643–666.

[11] i2tutorials, *Deep dive into bidirectional LSTM*, https://www.i2tutorials.com/deep-dive-into-bidirectional-lstm/, (2019).

[12] G. Jain, M. Sharma and B. Agarwal, *Optimizing semantic LSTM for spam detection*, Int. J. Inf. Technol. **11** (2019), no. 2, 239–250.

[13] S. Jamali and V. Shaker, *Defense against SYN flooding attacks: A particle swarm optimization approach*, Comput. Elect. Eng. **40** (2014), no. 6, 2013–2025.

[14] B. Karan, D. Narayan and P. Hiremath, *Detection of DDoS attacks in software defined networks*, Proc. 3rd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Sol. 2018, pp. 265–270.

[15] A. Karim, R.B. Salleh, M. Shiraz, S.A.A. Shah, I. Awan and N.B. Anuar, *Botnet detection techniques: Review, future trends, and issues*, J. Zhejiang Univ. Sci. **15** (2014), no. 11, 943–983.

[16] I. Kotenko and A. Ulanov, *Agent-based simulation of DDOS attacks and defense mechanisms*, Int. J. Comput. **4** (2005), no. 2, 113–123.

[17] F. Lau, S.H. Rubin, M.H. Smith and L. Trajkovic, *Distributed denial of service attacks*, Smc 2000 Conf. Proc. IEEE Int. Conf. Syst. Man Cyber. **3** (2000), pp. 2275–2280.

[18] D.E. Levine and G.C. Kessler, *Computer security handbook*, Chapter 11-Denial of Service Attacks, Computer Security Handbook, S. Bosworth and M.E. Kabay (eds), John Wiley & Sons, 2002.

[19] S. Lim, J. Ha, H. Kim, Y. Kim and S. Yong, *A SDN-oriented DDoS blocking scheme for botnet-based attacks*, Proc. 6th Int. Conf. Ubiquitous Future Netw. (ICUFN), 2014, pp. 63–68.

[20] F.S.D. Lima Filho, F.A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar and L.F. Silveira, *Smart detection: An online approach for DoS/DDoS attack detection using machine learning*, Secur. Commun. Networks **2019** (2019).

[21] S.S. Mohammed, R. Hussain, O. Senko, B. Bimaganbetov, J. Lee, F. Hussain, C.A. Kerrache, E. Barka and M.Z.A. Bhuiyan, *A new machine learning-based collaborative DDoS mitigation mechanism in software-defined network*, 2018 14th Int. Conf. Wireless Mobile Comput. Network. Commun., 2018, pp. 1–8.

[22] M. Nguyen, *Illustrated guide to LSTM's and GRU's: A step by step explanation*, Towards Data Science, 2018.

[23] T. Omrani, A. Dallali, B.C. Rhaimi and J. Fattahi, *Fusion of ANN and SVM classifiers for network attack detection*, 2017 18th Int. Conf. Sci. Tech. Automatic Control Comput. Engin. IEEE, 2017, pp. 374–377.

[24] Pranj52, *Essentials of deep learning: Introduction to long short term memory*, https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/, 2017.

[25] K.M. Prasad, A.R. Reddy and K.V. Rao, *DoS and DDoS attacks: Defense, detection and traceback mechanisms–A survey*, Global J. Comput. Sci. Technol. **14** (2014), no. 7, 1–19.

[26] K. Pykes, *The vanishing/exploding gradient problem in deep neural networks*, https://towardsdatascience.com/the-vanishing-exploding-gradient-problem-in-deep-neural-networks-191358470c11, (2020).

[27] O. Rahman, M.A.G. Quraishi and C.-H. Lung, *DDoS attacks detection and mitigation in SDN using machine learning*, Proc. IEEE World Cong. Serv. **2642** (2019), 184–189.

[28] S.A. Riga, *Two breaches, two enforcement actions, and a Ddos attack: data security and the rise of the internet of things*, J. Internet Law **20** (2017), no. 9, 3–7.

[29] T. Roempluk and O. Surinta, *A machine learning approach for detecting distributed denial of service attacks*, 2019 Joint Int. Conf. Digital Arts, Media Technol. ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON), IEEE, 2019, pp. 146–149.

[30] S. Selvin, R. Vinayakumar, E.A. Gopalakrishnan, V.K. Menon and K.P. Soman, *Stock price prediction using LSTM, RNN and CNN- sliding window model*, Proc. Int. Conf. Adv. Comput. Commun. Inf. 2017, pp. 1643–1647.

[31] I. Sharafaldin, A.H. Lashkari, S. Hakak and A.A. Ghorbani, *Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy*, Int. Conf. Secur. Technol. IEEE, 2019, pp. 1–8.

[32] T.A. Tang, L. Mhamdi, D. McLernon, S.A.R. Zaidi and M. Ghogho, *Deep recurrent neural network for intrusion detection in SDN-based networks*, Proc. 4th IEEE Conf. Network Softwarization Workshops (NetSoft), 2018, pp. 202–206.

[33] T. Wood, *Convolutional neural network definition— DeepAI.*, https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network, (2020).

[34] S. Yadav and S. Subramanian, *Detection of application layer DDoS attack by feature learning using stacked AutoEncoder*, Proc. Int. Conf. Comput. Tech. Inf. Commun. Technol. 2016, pp. 361–366.

[35] S. Yan, *Understanding LSTM and its diagrams*, https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714, 2016.

[36] J. Ye, X. Cheng, J. Zhu, L. Feng and L. Song, *A DDoS attack detection method based on SVM in software defined network*, Secur. Commun. Networks **2018** (2018).

[37] X. Yuan, C. Li and X. Li, *DeepDefense: Identifying DDoS attack via deep learning*, IEEE Int. Conf. Smart Comput. (SMARTCOMP), IEEE, 2017, pp. 1–8.

[38] S. Yuanyuan, W. Yongming, G. Lili, M. Zhongsong and J. Shan, *The comparison of optimizing SVM by GA and grid search*, 13th IEEE Int. Conf. Electron. Measur. Instrum. IEEE, 2017, pp. 354–360.

[39] C. Zhang, Z. Cai, W. Chen, X. Luo and J. Yin, *Flow level detection and filtering of low-rate DDoS*, Comput. Netw. **56** (2012), no. 15, 3417–3431.