Gradient Techniques to Predict Distributed Denial-Of-Service Attack

Roheen Qamar^{1,*}

¹Department of Computer Science, Quaid-e-Awam University of Engineering, Science and Technology, Nawabshah, Pakistan

*Corresponding Author: Roheen Qamar

DOI: <https://doi.org/10.52866/ijcsm.2022.02.01.006>

Received October 2021; Accepted January 2022; Available online March 2022

ABSTRACT: A distributed denial-of-service (DDoS) attack attempts to prevent people from accessing a server. A website may become inaccessible due to a DDoS attack because the server is inundated with fake requests and cannot handle real ones. A DDoS attack affects a large number of computers. Attackers employ a zombie network, which is a collection of infected machines on which the attacker has hidden the denial-of-service attacking application to carry out a DDoS attack. The MATLAB 2018a simulator was used in this study for training. Additionally, during design, the knowledge discovery dataset (KDD) was cleaned and the values of attacks were incorporated. A neural network model was subsequently developed, and the KDD was trained using a recursive artificial neural network. This network was developed using five distinct training algorithms: 1) Fletcher–Powell conjugate gradient, 2) Polak–Ribière conjugate gradient of, 3) resilient backpropagation, 4) gradient conjugation with Powell/Beale restarts, and 5) gradient descent algorithm with variable learning rate. The artificial neural network toolset in MATLAB was used to investigate the detection of DDoS attacks. The conjugate gradient with Powell/Beale restart algorithm had a success rate of 99.9% and a training time of 00:53. This inquiry uses the KDD-CUP99 dataset. Has a better level of accuracy, according to the results.

Keywords: Knowledge Discovery Data set (KDD); Artificial Neural Network (ANN); Recursive Neural Network (RNN); Distributed Denial of Service (DDoS) Attacks.

1. INTRODUCTION

Distributed denial of service (DDoS) attacks are designed to take down networks by denying them access to the resources they require to function. A DDoS assault occurs when a denial of service (DoS) attack begins with many source addresses. A huge number of compromised systems assault a single target, resulting in a DoS for the target system's users. Incoming message volume shuts down the target computer, preventing genuine users from accessing it. Decentralized DoS attacks have recently been developed in the realm of cyber-attacks. These attacks entail taking a server down and causing it to crash. Decentralized DoS attacks essentially shut down a server, forcing it to fail or preventing it from servicing genuine users. DDoS assaults are prevalent [1]. Fig 1 shows the complete architecture of a DDoS attack.

1.1 ZOMBIES IN THE DDOS CONTEXT

A “zombie” called a botnet is a collection of seized Web-based devices. These devices are infected with malware to manipulate them from a remote region outside the control of their valid proprietors. These botnet devices are used for illicit purposes, such as for junk mail or DDoS attacks [2]. DDoS can likewise be applied as a smokescreen for malicious activities and attack safety apparatuses, attacking the target's safety. Botnets are a huge range of frameworks infected with malware used for DDoS attacks. These bots or zombie frameworks are applied to attempt attacks toward the target

FIGURE 1. Structure of DDOS Attack

frameworks, regularly overpowering their information switch and preparation abilities [3]. Fig 2 shows the architecture of a botnet attack.

FIGURE 2. Architecture of Zombie Attack

1.2 DDOS AND ITS OPERATION

Unexpected increments in regular gridlocks could make a webpage stack progressively for actual clients. During certain instances, the number of visitors is sufficient to completely shut down a website. This type of attack is known as a dispersed forswearing of administration DDoS attack. Some particularly dangerous botnets use uncorrupted PCs as an element of the attack [4].

1.3 PROCESS OF DDOS PROTECTION

DDoS safety works by constructing calculations and propelled programming to show incoming visitors to the website. Any site visitor that is not an actual user is denied access, and actual visitors are filtered, allowing them to access the website. DDoS coverage considers the largest number of elements to protect against as wide a range of attacks as possible [5].

2. NEURAL NETWORKS

The artificial neural network (ANN), developed by Warren McCulloch and Walter Pitts in 1943, is one of the widely used supervised learning approaches that are influenced by the structural complexity of the human brain. An ANN is a computational version based on the form and factors of organic neural systems [6]. ANN is a statistics processing paradigm this is often stimulated through the style of organic neural systems like mind procedure statistics. ANN comprises a huge number of interconnected processing factors to solve particular problems. An ANN is a precise utility that includes include sample popularity by an aiming to know manner. getting to know is an organic device adjust to connection among neurons, and “a trained neural network can be thought of as an ‘expert’ in the category of information it has been given to analyze” [7].

2.1 BENEFITS OF NEURAL NETWORKS

1. Enhanced Learning

A capacity to examine mission based totally at the given for education and preliminary experience.

2. Specific Organization

Artificial Neural Network (ANN) can create its very own businesses or illustration of it's acquired at some point of studying time.

3. Real-Time Operation

Parallel Artificial Neural Network (ANN) processing is also used, and special hardware devices are being established to keep hold of this capability [8]. The architecture of an ANN is visualized in Fig 3.

FIGURE 3. Neural Network - Based Model

ANNs handle nonlinear statistics that are used to explore trends in complex relationships between inputs and outputs. ANNs are also a set of deep learning technologies that are also part of artificial intelligence. The commercial purpose of these applications is regularly focused on solving complex symbols or serving as cognitive examples. The most common form of ANN has three device units: an “input” connected to the “hidden” unit layer related to the unit class “output.” The data are in the input and layer streams that use the network before completing the output [9].

2.2 RECURSIVE NEURAL NETWORK

Recursive neural networks (RNN) use a similar hierarchy as do ANNs, but the input collection is time-limited. In RNNs, the inputs that maintain the hierarchy in a tree style will use the recursive air community. An example of such a method is as follows: the parse tree of a statement is explored by recursively retrieving the output of an operation completed in a small byte of text [10].

RNN is structured to apply the same set of weights to variable-sized input structures, create predictions, or traverse specified structures in topological order, resulting in a neural network that produces predictions [11]. What are dynamic recursive neural networks? Recurrent neural networks (RvNN) are used for natural language processing sequences and tree structures, with particular emphasis on learning syntax and sentence sequence representations based on word embedding [12]. RNNs were first used to learn distributed representations of structures containing logical words. Additional models and rich frameworks have been developed for RNNs.

FIGURE 4. (a) Recursive Neural Network

Generally, RNNs are developed by applying recursively at the same weight established at the structured input. For the same goal, this process happens in any node. RNNs refer to organized inputs, particularly architectures that work with induced non-cyclical graphics. RNNs can be used to solve it, and they are described as a deep tree [12].

FIGURE 5. (b) Recursive Neural Network

We use an RNN when we need to parse an entire sentence. Branching relationships and hierarchical structures are possible with a tree-like topology. How are RNNs different from RvNNs? To answer this question, we can complete the following tasks [13].

Inner approach – In this approach, recursion is typically conducted within an underlying graph, and the goal is usually accomplished by moving forward slowly along the graph's edges [13].

Outer Approach – In this approach, recursion is normally performed outside of the underlying graph, and data are aggregated over increasingly longer distances in a rectangular direction.

RNNs are used to predict hierarchical outputs from variable-size input structures and provide scalar predictions by traversing a structure in topological order. RNNs are also capable of responding not only to input but also to context. They process each time series input individually. RNNs were first introduced in 2000 [13].

FIGURE 6. (A) Recursive Neural Network Inner Approach

FIGURE 7. (B) Recursive Neural Network Outer Approach

3. RELATED WORK

Bogdanoski et al. [14] investigated vulnerabilities in systems that are attacked by a segment when the synchronize (SYN) flag is set to the transmission control protocol (TCP). The authors described a TCP protocol vulnerability that leads to a SYN flood attack, which is a type of DoS attack. They used OPNET Modeler simulation tools used in 802.11e environments to see how SYN flood attacks and DDoS SYN flood attacks affect other parameters of different types of traffic (voice, video, and data). The result is as expected: during a SYN flood attack, the situation with all network parameters worsened, but even these results were significantly worse than the results obtained in previous case simulations, including DDoS SYN floods. The last successful anomaly detection algorithm was used to handle the SYN flood attack. The authors further showed how to minimize the effects of SYN flood DoS attacks on Linux and Windows systems.

Asad et al. [15] presented a unique deep neural network-based detection technique that used feed-forward backpropagation to correctly identify numerous application layer DDoS assaults. On a state-of-the-art dataset encompassing several types of DDoS attacks, the proposed neural network architecture could detect and use the most relevant high-level aspects of packet flows with an accuracy of 98%.

Mane et al. [16] focused on intrusion, which refers to unauthorized access to or addition to a device. Intrusion detection systems (IDS) are used to identify intrusion signatures and protect the system from various network attacks such as DoS, U2R, and R2L probes. IDS collect information from networks and other parts of a system. The authors introduced an anomaly intrusion detection system that scans and detects various network attacks. They used a directed neural network, namely, a backpropagation artificial neural network algorithm, to classify specific attacks and secure complete data. The authors tested a neural network using only the most important features (17 out of 41) of the knowledge discovery dataset (KDD) 1999 dataset. The results of the system were investigated. IDS use supervision and neural networks.

Roempluk et al. [17] proposed a strategy to detect DDoS attacks. KDD Cup 1999 and NSL-KDD were used as benchmark datasets. After examining the dataset and removing duplicate data, the number of records in the KDD Cup 1999 dataset was reduced from 4,898,431 to 529,655, and that in the NSL-KDD dataset was reduced from 125,373 to 12,354. The records were consistently reduced due to the peculiarities of DDoS assaults, which deliver repeated data to the victims' servers. The researchers used K-nearest neighbor (KNN), multi-layer perceptron, and support vector machine (SVM) to train on alphabet data, finding that KNN was the most effective strategy to identify DDoS attacks.

Ali et al. [18] investigated the information collected by an IDS from different parts of the network and the system. The authors described an abnormal IDS that can detect various network attacks. They used supervised neural networks (i.e., backpropagation ANN algorithms) to identify such attacks and make all of the data safe. The proposed technique uses Bayesian regularization to successfully detect DoS/DDoS attacks with an accuracy rate of 99.6%. Depending on the result, the accuracy rate for using scaled conjugate gradient descent is 97.7%. The authors used the CICIDS2017 dataset to create an accurate and practical intrusion detection model. Their method had a detection accuracy of 99.6%, with possible increases by using more suitable training data.

Myint Oo et al. [19] proposed a detection method for DDoS attacks based on SDN that will disrupt legitimate user activities as little as possible. They also proposed an advanced SVM (ASVM) technique to enhance the extant SVM algorithm to detect DDoS attacks. The ASVM methodology is a multiclass classification method with three classes. They used two critical characteristics, namely, volumetric and asymmetric features, to shorten training and testing time in their detection approach, ultimately attaining an accuracy of approximately 97% and the shortest training and testing time.

Prakash et al. [20] developed a deep learning-based defense mechanism, which is a lightweight DoS detection system, to identify and isolate assaults in the data forwarding phase (DFP). The system detects DoS assaults like depletion, jamming, homing, and flooding. Extensive simulation studies were run to precisely separate attackers and make the system more resistant to DoS assaults. Their results demonstrate a high detection rate, throughput, packet delivery ratio, and accuracy. Energy usage and the number of false alarms were also minimized.

Kumar et al. [21] coined the term "attack amplification factor," which refers to the degree of amplification experienced by a real attacker. Specifically, a fully distributed DoS attack based on Smurf is an amplification attack in which the attacker uses an unprotected intermediate network to increase the load of attack traffic and direct it to the victim device.

Mahbod et al. [22] found that KDD 99 is the primary broadly used dataset for the assessment of those systems. To clear up those issues, they proposed a new dataset, NSL-KDD, which includes the information from the entire KDD and is no longer afflicted by any notable shortcomings. The first critical deficiency with the KDD information set is the considerable amount of redundant information, resulting in the need for further algorithms.

Boche et al. [23] investigated the detectability of DoS assaults using a detection system based on Turing machines. Turing machines have no constraints on computational complexity, processing capacity, or storage. They can also replicate any given method, making them a basic yet extremely powerful model of computation. Turing machines define the fundamental performance boundaries of contemporary digital computers. No Turing computer can determine whether a DoS assault is conceivable for a particular channel in both cases of a jammer with partial information and complete knowledge,

making the relevant decision problem is undecidable. By contrast, for both situations, algorithmically describing the channels that are not vulnerable to a DoS assault is possible, implying that circumstances in which the jammer is unable to impede communication can be detected. For all other channels, the equivalent Turing machine does not stop and continues to operate indefinitely, rendering this decision problem semi- decidable. Extra coordinating resources, such as shared randomness, make communication more resilient to DoS attacks.

Premkumar , M., and T. V. P. Sundararajan Ali et al. [24]. In this research, a novel lightweight DoS detection system Deep Learning-based Defense Mechanism (DLDM) is suggested to identify and isolate assaults in the Data Forwarding Phase (DFP). This study offers a novel technique for detecting DoS assaults such as depletion, jamming, homing, and flooding. We do comprehensive simulation trials that correctly segregate the attackers and make it more resistant to DoS assaults. Our suggested simulation result demonstrates that it can achieve a high detection rate, throughput, packet delivery ratio, and accuracy. This also minimizes energy usage and the likelihood of false alarms.

Youssef et al. [25] provided an overview of contemporary detection and mitigation algorithms for four types of DDoS attacks: UDP flood, TCP SYN flood, ping of death, and Smurf attack. They presented a detection and security algorithm and tested it in comparison to a modern intrusion detection and prevention method to determine its feasibility in mitigating DDoS attacks in a community environment. The proposed algorithm's false positive and detection accuracy were also evaluated. The authors further examined the four types of DDoS assaults and their outcomes in relation to a variety of current DDoS detection and protection algorithms.

Sambangi et al. [26] investigated DDoS attack detection in a cloud setting using the prominent CICIDS 2017 benchmark dataset and multiple regression analysis to develop a machine learning (ML) model to predict DDoS and bot assaults using a Friday afternoon traffic logfile. They developed an ML model that is an ensemble of feature selection using information gain and regression analysis. Friday morning and afternoon logfiles, which contain benign, bot, and DDoS classifications, were specifically considered. For the Friday morning logfile, the ensemble model achieved a prediction accuracy of 97.86%. For the Friday afternoon logfile, the prediction accuracy was 73.79% for 16 characteristics derived using information gain-based feature selection and regression analysis-based ML.

Kumar et al. [27] investigated distributed P2P networks that are widely used for document sharing. They evaluated their proposed device is evaluated on a community simulation platform called Network Simulator Version 2. They found that detection strategies can accurately detect DDoS attacks and protection mechanisms can properly manage attack visitors and maintain carrier benefits for valid site visitors.

Rani et al. [28] investigated zombie assaults used in DDoS attacks, finding that large-scale cooperative assaults released from a huge quantity of compromised hosts (zombies) are a prime risk to Internet services.

4. DESIGN OF PROPOSED ARCHITECTURE

FIGURE 8. Proposed Plan

1. Collecting the Information

First, the KDD was gathered from a reliable source. A wide range of recognized patterns and ML algorithms were used to develop the KDD1999 dataset, which was launched for the KDD Cup competition in 1999. The standard KDD1999 database, which contains connecting records of attacks and interference in a network, was used for testing and training [29].

2. Data Pre-processing

Before the training and pre-processing, the KDD dataset was cleaned, and protocol, attack, and flag values were assigned [29].

3. Training Algorithm

The proposed network version was built using the data collected from the data-gathering stage [29].

4. Recursive Neural Network

Neural network training is a significant part of DDoS attack simulation. In this phase, an RNN was trained to analyze the predefined behavior of proposed algorithms and compare the results with better accuracy and shorter timing [29].

5. Final Result

The better training algorithm was determined, enabling a DDoS attack to be identified by using the KDD dataset [29].

4.1 SYSTEM PERFORMANCE

MATLAB 2018a was used in this study. First, the KDD statistics set was cleaned and the values of protocols, assaults, and flags were supplied. Then, the neural network was expanded and trained on the KDD data, enabling the use of the ANN. Finally, the outcomes of DDoS assaults were provided.

The Traincgf, Traincgp, Trainrp, Traincgb, and Traingdx algorithms were used to train a neural network that updates weight and bias values. In this training, 10 attacks were used. The factors' "convention type" were modified with the following characteristics: TCP=0, SMTP=1, UDP=2, FTP_data=3, Domain=4, Telenet=5, NTP_u=6, Private=7, ICMP=8, and POP3=9. A total of 41 inputs and 20 neurons were in the hidden layer, and five neurons were in the output layer of the neural network. The DDoS attacks are divided into five classes for simplicity [30].

FIGURE 9. Artificial Neural Network

In This Research, Five Algorithms Have Been Used And Trained

Table 1. OF ALGORITHMS

S:no	Algorithm
1.	Traincgf Fletcher-Powell Conjugate Gradient Algorithm.
2.	Traincgp Polak-Ribière Conjugate Gradient Algorithm
3.	Trainrp Resilient Back propagation Algorithm
4.	Traincgb Conjugate Gradient with Powell/Beale Restarts Algorithm
5.	Traingdx Variable Learning Rate Gradient Descent Algorithm

The results show the high overall accuracy of the neural networks by using different algorithms. The dataset has more than 20 classes, which are sub-divided into 5 classes for simplicity. The first four classes are individual classes in the data set, and the fifth class includes all other classes [30].

4.2 TRAINED ALGORITHMS

4.2.1. Fletcher-Powell Conjugate Gradient Training Algorithm Result

The Fletcher-Reeves conjugate gradient training algorithm is a variation of the Fletcher and Reeves conjugate gradient system (1964). If the weight, net input, and transfer function have derivatives, the algorithm can train all networks. According to Sandhu and Chhabra (2011), the search path for all pairs of tilt algorithms is occasionally reset to negative tilt. If the number of network weights and biases is the same as the number of iterations, then the general reset point is reached. By contrast, other reset methods may increase training performance. One such method is the Powell/Beale restart strategy (Powell, 1977; Beale, 1972). If minimal orthogonally is observed between the present and previous gradients, this technique restarts [31].

The Fletcher-Reeves algorithm entails establishing the optimum distance between searches so that they are conjugated by the parameter β [32].

$$X_{K+1} = X_K + \beta_k p_k$$

$$p_k = -g_k + \beta_k p_{k-1}$$

$$\beta_k = \frac{\Delta g_k^T g_k}{g_k^T g_{k-1}}$$

where β_k is the Fletcher-Reeves parameter.

Fig 8 shows the classification value of the Fletcher-Powell conjugate gradient training algorithm, which is 99.8%. Its misclassification value is 0.2%.

Output Class	1	2	3	4	5	
1	97102 19.7%	15 0.0%	9 0.0%	2 0.0%	465 0.1%	99.5% 0.5%
2	23 0.0%	280771 56.8%	0 0.0%	0 0.0%	0 0.0%	100.0% 0.0%
3	1 0.0%	0 0.0%	107188 21.7%	0 0.0%	1 0.0%	100.0% 0.0%
4	4 0.0%	0 0.0%	0 0.0%	977 0.2%	88 0.0%	91.4% 8.6%
5	148 0.0%	4 0.0%	4 0.0%	0 0.0%	7219 1.5%	97.9% 2.1%
	99.8% 0.2%	100.0% 0.0%	100.0% 0.0%	99.8% 0.2%	92.9% 7.1%	99.8% 0.2%
	1	2	3	4	5	
	Target Class					

FIGURE 10. Confusion Matrix of Fletcher-Powell Conjugate Gradient

The neural network training state plot is shown in Fig 9. The figure shows that validation took 0.00072259 seconds over 870 epochs. The output demonstrates that validation is focused on lowering the threshold. The performance curve developed during network training, testing, and validation is depicted in the diagram. Fig 9 also shows the performance curve produced during training, testing, and validation of the network.

4.2.2. Polak-Ribière Conjugate Gradient Training Algorithm

The trainCGP algorithm is another version of the conjugate gradient method proposed by Polak and Ribière in 1969 [33]. Fig 10 shows the classification value of the Polak-Ribière conjugate gradient training algorithm, which is 99.5%. Its misclassification value is 0.5%.

The algorithm is in the first order for the minimum function. Deep learning neural networks are trained using the Polak-Ribière Conjugate Gradient optimization algorithm. The learning rate is a hyperparameter that determines how much to modify the model in response to the predicted error every time the weights of the model are adjusted. The amount

FIGURE 11. Performance of Fletcher-Powell Conjugate Gradient

Recursive Neural Networks Confusion Matrix						
Output Class	1	2	3	4	5	
	19099 19.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	197 0.2%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	1513 1.5%	0 0.0%	6 0.0%	99.6% 0.4%
	0 0.0%	0 0.0%	0 0.0%	20481 20.5%	0 0.0%	100% 0.0%
	5 0.0%	2 0.0%	489 0.5%	1 0.0%	58207 58.2%	99.2% 0.8%
						Target Class
						1
						2
						3
						4
						5

FIGURE 12. Confusion Matrix of Polak-Ribière Conjugate Gradient

of weights that are changed during training is referred to as the step size or the “learning rate” [34]. The Polak-Ribiere algorithm is Fletcher–Reeves but with a different β value.

$$\beta_k = \frac{\Delta g_k^T g_k}{g_k^T - 1 g_k - 1}$$

After complete training of the neural network, the best validation performance is 0.0023231 at epoch 248 and time 00:03:29, as shown in Fig 11.

FIGURE 13. Performance of Polak-Ribière Conjugate Gradient

4.2.3. Resilient Backpropagation Training Algorithm Result

A supervised algorithm with a linear convergence rate called scaled conjugate gradient is implemented. The algorithm is based on an optimization class known as conjugate gradient methods in numerical analysis [34].

The neuron transfer function is used to compact the input into a fixed output range in a multi-layer neural network. Owing to partial derivatives, only minimal changes in weights occur, and the network does not converge. Resilient backpropagation addresses this problem by calculating only the direction in which partial derivatives are used to change weights, resulting in less memory usage. [35].

$$\begin{aligned} dX &= \text{deltaX} \cdot \text{sign}(gX) \\ \text{where } \text{deltaX} &= \text{derivative of performance, } gX = \text{gradient,} \\ dX &= \text{change in weight} \end{aligned}$$

Fig 12 shows the classification value of the resilient backpropagation training algorithm, which is 99.4%. Its misclassification value is 0.6%.

$$|g_{k-1}^T g_k| \geq 0.2 \|g_k\|^2$$

The complete training of the neural network is shown in Fig 13. The epoch value is 907 iterations, the total time is 00:27:20. The best performance is at epoch 1000, and the complete validation performance is 0.0020672.

4.2.4. Conjugate Gradient with Powell/Beale Restart Training Algorithm

The search path for all combined gradient algorithms is sometimes reset with the negative gradient, according to Sandhu and Chabra (2011). Normal reset points are achieved when the number of network weights and components corresponds

Recursive Neural Networks Confusion Matrix						
Output Class	1	2	3	4	5	
	97001 19.6%	98 0.0%	22 0.0%	7 0.0%	2730 0.6%	97.1% 2.9%
	17 0.0%	280692 56.8%	0 0.0%	0 0.0%	0 0.0%	100.0% 0.0%
	25 0.0%	0 0.0%	107179 21.7%	0 0.0%	52 0.0%	99.9% 0.1%
	0 0.0%	0 0.0%	0 0.0%	972 0.2%	0 0.0%	100% 0.0%
	235 0.0%	0 0.0%	0 0.0%	0 0.0%	4991 1.0%	95.5% 4.5%
Target Class						
	1	2	3	4	5	
	99.7% 0.3%	100.0% 0.0%	100.0% 0.0%	99.3% 0.7%	64.2% 35.8%	99.4% 0.6%

FIGURE 14. Confusion Matrix of Resilient Back Propagation

FIGURE 15. Performance of Resilient Back propagation

to the number of repetitions. Other reset methods, however, can increase the training performance. One such combined gradient algorithm is used in the present study. The path of the slope to the set must be specified in predetermined periods. A restart is implemented if the online delivery of the gradients changes slightly, according to this technique [25, 26]. The case of search orientation should be reset to the sound level of the gradient slope, and equation (19) is used. One method is the Powell/Beale restart method (Powell, 1977; Beale, 1972). This technique starts when minimal regulations exist between the current gradient and the previous slope. The direction of the gradient to the ministry must be determined in predefined intervals in the combined gradient algorithms. This strategy only shows the restart when the gradient output does not change minimally [35, 36]. The recognition (19) is used to determine when the search direction must be reset to the sound level of the slope.

$$|g_{k-1}^T g_k| \geq 0.2 \|g_k\|^2$$

The k th iteration's gradient is denoted by g_k . If this condition is met, the search direction is reset to the negative of the gradient. Any network with derivative functions for its weight, net input, and transfer functions can be trained using this algorithm.

Figure 14 shows the classification value of the conjugate gradient with Powell/Beale restart training algorithm, which is 99.5%. Its misclassification value is 0.5%.

Output Class	1	2	3	4	5	
1	97008 19.6%	71 0.0%	18 0.0%	8 0.0%	2054 0.4%	97.8% 2.2%
2	19 0.0%	280716 56.8%	0 0.0%	0 0.0%	0 0.0%	100.0% 0.0%
3	5 0.0%	0 0.0%	107180 21.7%	0 0.0%	30 0.0%	100.0% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	970 0.2%	0 0.0%	100% 0.0%
5	246 0.0%	3 0.0%	3 0.0%	1 0.0%	5689 1.2%	95.7% 4.3%
	99.7% 0.3%	100.0% 0.0%	100.0% 0.0%	99.1% 0.9%	73.2% 26.8%	99.5% 0.5%
	1	2	3	4	5	
	Target Class					

FIGURE 16. Confusion Matrix Conjugate Gradient with Powell/Beale Restarts

Fig 15 shows the performance curve produced during training, testing, and validation of the network. The best validation performance of 0.001167 is obtained at 556 epochs.

4.2.5. Variable Learning Rate Gradient Descent

The trainGDX method combines adaptive learning rate and momentum training into one algorithm. TrainGDX is similar to trainGDA, but with the inclusion of the momentum coefficient as a training parameter. This algorithm can be used to train any network that has derivative functions for its weight, net input, and transfer functions. This algorithm uses gradient descent with momentum to keep networks from becoming stuck in shallow local minimums.

$$dX = m_e \Delta X_{k-1} + lr * m_c * \frac{\text{delta}X}{dX}$$

where m_c = momentum coefficient, lr =learning rate. Delta X = derivative of performance [37, 38]. Fig 16 shows the classification value of the variable learning rate gradient descent training algorithm, which is 99.7%. Its misclassification value is 2.4%.

Figure 17 shows the performance curve produced during training, testing, and validation of the network. The best validation performance of 0.0088901 is obtained at 162 epochs.

FIGURE 17. Conjugate Gradient with Powell/Beale Restarts Training Algorithm Result

Recursive Neural Networks Confusion Matrix						
Output Class	1	2	3	4	5	
	19037 19.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	0 0.0%	0 0.0%	0 0.0%	20482 20.5%	176 0.2%	99.1% 0.9%
	67 0.1%	199 0.2%	2002 2.0%	0 0.0%	58037 58.0%	96.2% 3.8%
						Target Class
	1	2	3	4	5	
	99.6% 0.4%	0.0% 100%	0.0% 100%	100% 0.0%	99.7% 0.3%	97.6% 2.4%

FIGURE 18. Confusion Matrix of Variable Learning Rate Gradient Descent

FIGURE 19. Performance Variable Learning Rate Gradient Descent

Table 2 shows the RNN training of three different algorithms. The variable learning rate gradient descent algorithm gives better results than do two other selected algorithms. The success rate (accuracy) is 99.9% and training is done in less time, namely, 2 minutes and 29 seconds.

Table 2. Success Rates and Timing of All Stages

S#	Algorithm	Success Rate	Timing
1.	Fletcher-Powell Conjugate Gradient Algorithm.	99.8%	34.11
2.	Polak-Ribière Conjugate Gradient Algorithm.	99.5%	24:21
3.	Resilient Back propagation Algorithm	99.4%	09:42
4.	Conjugate Gradient with Powell/Beale Restarts Algorithm	99.5%	17:53
5.	Variable Learning Rate Gradient Descent Algorithm	99.7%	21:47

5. RESULTS AND CONCLUSION

DDoS attacks present a substantial threat to the proper operation of modern networks. The goal of a DDoS attack is to completely destroy the target's capital. Legitimate users are refused service because the victim's system's communication memory and CPU resources are being consumed due to high traffic. DDoS attacks cause considerable critical security difficulties, such as an inability to access resources, for Internet users. In this study, RNNs were used to train and identify DDoS attacks. Five well-known algorithms were chosen: Fletcher-Powell conjugate gradient algorithm, Polak-Ribière conjugate gradient algorithm, resilient backpropagation algorithm, conjugate gradient with Powell/Beale restart algorithm, and variable learning rate gradient descent algorithm. The goal was to find the best algorithm in terms of performance and training duration.

An RNN was created to evaluate the accuracy and detection of DDoS attacks. In comparison with the other four algorithms, the resilient backpropagation algorithm gives superior results in a short training period, with an excellent precision output of 99.4% accuracy with a training time of 9 minutes and 42 seconds. Future studies on deep learning and ML may use a range of approaches, models, and neural networks. Other algorithms may be evaluated to determine which neural network or approach is most effective in detecting DDoS attacks.

6. ACKNOWLEDGMENTS

The first author thanks the reviewers for their useful suggestions that improved the presentation of this paper.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

- [1] Rajakumaran, Gayathri, Neelanarayanan Venkataraman, and Raghava Rao Mukkamala. "Denial of Service Attack Prediction Using Gradient Descent Algorithm." *SN Computer Science* vol. 1, no. 1, p. 1-8, 2020
- [2] Kaur Chahal, Jasmeen, Abhinav Bhandari, and Sunny Behal. "Distributed denial of service attacks: a threat or challenge." *New Review of Information Networking* 24, no. 1, 31-103, 2019
- [3] H. Monowar, H. J. Bhuyan, D. K. Kashyap, Bhattacharyya1 and J. K. Kalita. Received 00 July 2012; revised 00 December 2012 Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions PP 1- 12
- [4] X. Zang, A. Tangpong, G. Kesidis, and D. J. Miller, "Botnet Detection through Fine Flow Classification", no. 0915552, pp. 117, 2011.
- [5] Q. Li, L. Meng, Y. Zhang, and J. Yan. "DDoS attacks detection using machine learning algorithms". In *International Forum on Digital TV and Wireless Multimedia Communications* (pp. 205-216). Springer, Singapore. 2018, September
- [6] B. Yegnanarayana. "Artificial Neural Networks, PHI Learning Pvt. Ltd.", p. 461, 2009
- [7] Brij Bhooshan Gupta, Ramesh Chand Joshi, and Manoj Misra . Nov. 2011 "ANN Based Scheme to Predict Number of Zombies in a DDoS Attack", *International Journal of Network Security*, Vol.13, No.3, PP.216225
- [8] Cil, Abdullah Emir, Kazim Yildiz, and Ali Buldu. "Detection of DDoS attacks with feed forward based deep neural network model." *Expert Systems with Applications* vol. 169, 114520, 2021
- [9] R. Paulus, R. Socher, and C. D. "Manning. Global belief recursive neural networks". *Advances in Neural Information Processing Systems*, 27, 2888-2896, 2014
- [10] Q. Guo, Z. Yu, Y. Wu, D. Liang, H. Qin, and J. Yan. "Dynamic recursive neural network". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5147-5156), 2019
- [11] Muhammad, Arif Wirawan, Cik Feresa Mohd Foozy, and Kamaruddin Malik bin Mohammed. "Multischeme feedforward artificial neural network architecture for DDoS attack detection." *Bulletin of Electrical Engineering and Informatics* 10, no. 1, 458-465, 2021
- [12] S. Liu, N. Yang, M. Li, and M. Zhou. "A recursive recurrent neural network for statistical machine translation". 2014
- [13] Socher, Richard; Perelygin, Alex; Y. Wu, Jean; Chuang, Jason; D. Manning, Christopher; Y. Ng, Andrew; Potts, Christopher. "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank" (PDF). *Emnlp* 2013.
- [14] M. Bogdanoski, T. Suminoski, and A. Risteski. "Analysis of the SYN flood DoS attack". *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 5, no. (8), 1-11, 2013
- [15] Asad, Muhammad, Muhammad Asim, Talha Javed, Mirza O. Beg, Hasan Mujtaba, and Sohail Abbas. "Deepdetect: detection of distributed denial of service attacks using deep learning." *The Computer Journal* 63, no. 7, 983-994. 2020
- [16] V. D. Mane, and S. N. Pawar. "Anomaly based ids using backpropagation neural network". *International Journal of Computer Applications*, vol. 136, no (10), 29-34, 2016
- [17] Roempluk, Tanaphon, and Olarik Surinta. "A machine learning approach for detecting distributed denial of service attacks." In *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*, pp. 146-149. IEEE, 2019.
- [18] O. Ali, and P. Cotae. "Towards DoS/DDoS Attack Detection Using Artificial Neural Networks". In *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 229-234). IEEE. 2018, November
- [19] Myint Oo, Myo, Sinchai Kamolphiwong, Thossaporn Kamolphiwong, and Sangsuee Vasupongayya. "Advanced support vector machine-(ASVM-) based detection for distributed denial of service (DDoS) attack on software defined networking (SDN)." *Journal of Computer Networks and Communications* 2019 (2019).
- [20] Prakash, Aditya, and Rojalina Priyadarshini. "An intelligent software defined network controller for preventing distributed denial of service attack." In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 585-589. IEEE, 2018.
- [21] Sanjeev Kumar, Senior Member, "IEEE Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet" PP 1-7
- [22] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. "IEEE 2009 A Detailed Analysis of the KDD CUP 99 Data Set". PP 1-7
- [23] Boche, Holger, Rafael F. Schaefer, and H. Vincent Poor. "Denial-of-service attacks on communication systems: Detectability and jammer knowledge." *IEEE Transactions on Signal Processing* vol. 68: 3754-3768, 2020
- [24] M. Premkumar, and T. V. P. Sundararajan. "DLDM: Deep learning-based defense mechanism for denial of service attacks in wireless sensor networks." *Microprocessors and Microsystems* vol. 79: 103278, 2020
- [25] M. A. M. Yusof, F. H. M. Ali, and M. Y. Darus. "Detection and defense algorithms of different types of DDoS attacks". *International Journal of Engineering and Technology*, vol. 9, no. (5), 410, 2017
- [26] Sambangi, Swathi, and Lakshmeeswari Gondi. "A Machine Learning Approach for DDoS (Distributed Denial of Service) Attack Detection Using Multiple Linear Regression." *Multidisciplinary Digital Publishing Institute Proceedings* vol. 63, no. 1: p. 51, 2020
- [27] Vooka Pavan Kumar, Abhinava Sundaram., "Munnaluri Bharath Kumar, ANALYSIS OF DDoS ATTACKS IN DISTRIBUTED PEER TO PEER NETWORKS". *Journal of Global Research in Computer Science* Volume 2, No. 7, July 2011
- [28] By Sumathi Rani Manukonda & Dr. "Koppula Srinivas Rao Double Blind Effective Detection and Prevention of Ddos based on Big Data Mapreduce". *Peer Reviewed International Research Journal*. Publisher: Global Journals Inc. (USA). Volume 15 Issue 6 Version 1.0 Year 2015 PP 1-7
- [29] Mukhopadhyay, 2, S Polle, 3, P Naskar. "(May-Jun. 2014) Simulation of Denial of Service (DoS) Attack using Matlab and Xilinx IOSR Journal of Computer Engineering (IOSR-JCE)" Volume 16 PP 119-125
- [30] D.M.Choudhari 1Associate Prof, "Department of Computer Science and Engineering ISSUE2", 2014DDOS Attack Detection Based On Ensemble of Neural Classifier VOLUME1, PP.1-7
- [31] M. F. Moller. "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning", *Neural Networks*, Vol. 6, No. 4, pp. 525-533. 1993
- [32] R. Fletcher, and C.M. Reeves, "Function minimization by conjugate gradients", *Computer Journal*, Vol. 7, pp. 149154, 1964
- [33] P. S. Sandhu and S. Chhabra. "A Comparative Analysis of Conjugate Gradient Algorithms and PSO Based Neural Network Approaches for Reusability Evaluation of Procedure Based Software Systems", *Chiang Mai Journal of Science*, Vol. 38, No. 2, pp.123-135. 2011
- [34] M. Riedmiller, and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm", *Proceedings of the IEEE International Conference on Neural Networks*, 1993.

- [35] Arshi, M., M. D. Nasreen, and Karanam Madhavi. "A survey of DDOS attacks using machine learning techniques." In E3S Web of Conferences, vol. 184, p. 01052. EDP Sciences, 2020.
- [36] Adeshina, Qozeem Adeniyi. "Machine learning based approach for detecting Distributed Denial of Service attack." (2021).
- [37] Beale, E.M.L., A derivation of conjugate gradients, in F.A. Lootsma, Ed., Numerical methods for nonlinear optimization, London: Academic Press, 1972.
- [38] Ugwu, Chukwuemeka Christian, Olumide Olayinka Obe, Olugbemiga Solomon Popola, and Adebayo Olusla Adetunmbi. "A Distributed Denial of Service Attack Detection System using Long Short Term Memory with Singular Value Decomposition." In 2020 IEEE 2nd International Conference on Cyberspac (CYBER NIGERIA), pp. 112-118. IEEE, 2021.