University of Sindh Journal of Information and Communication Technology
(USJICT)
Volume 4, Issue 4, December 2020

ISSN-E: 2523-1235, ISSN-P: 2521-5582
Website: http://sujo.usindh.edu.pk/index.php/USJICT/
© Published by University of Sindh, Jamshoro

DDoS ATTACKS DETECTION USING AUTOENCODERS

Dr. Fareed Ahmed Jokhio, Dr. Abdul Wahid Memon, Dr. Irfana Memon, Engr. Aisha Jokhio, Prof. Dr. Sulleman Memon

Department of Computer Systems Engineering, QUEST Nawabshah Pakistan
fajokhio@quest.edu.pk, awam@quest.edu.pk, irfanahameed@quest.edu.pk aishajokhio@quest.edu.pk
sulleman@quest.edu.pk

Abstract: Computer networks have several issues such as insertion attacks, denial of service attacks, traffic jamming, and unauthorized access. Due to these issues network security is most important. In a network, Distributed Denial of Service (DDoS) attacks may cause significant degradation of the performance of any application. It is very challenging to detect such attacks and undetected attacks are considered as a threat. This paper describes comparative analysis of Denial of Service attacks detection using Feed Forward Neural Networks and Autoencoders which are machine learning based approaches and are usually used for feature learning.

Keywords: DDoS attacks; Neural Networks; Deep Learning; autoencoders

I. INTRODUCTION

As the networks are evaluated, the number and type of network attacks is also growing. Among these attacks, Denial of Service or distributed denial of service attacks are used to bring a system down and these attacks are real threats in networks [21].

DDoS attack is an attempt to make online services unavailable by overloading it with traffic from multiple sources. Although it is not new threat however, it is one of the major security challenges for service providers and their users. The urgency for rapid detection of DDoS attacks and proper management is dire need of networks today. In order to detect the DDoS attacks, the features selection and classification of DDoS attacks is possible through genetic algorithms and more complex algorithms such as neural networks [24]. The detection through neural networks and other machine learning techniques like, deep learning, which is a new dimension to mitigate the DDoS attacks in a networked environment. In this study, we examine the attributes of feed-forward neural networks and autoencoders that help in attack detection.

The feed-forward neural networks have multiple layers of neurons. In these neural network information travels in forward direction only. A feed-forward neural network may have input nodes and hidden layers and output nodes. In the hidden layer sigmoid neurons are used. These neural networks are simple and easy to build and are also known as Multi-layered Network of Neurons (MLN).

Autoencoders are also neural networks consisting of both encoder and decoder. Autoencoders are used for dimensionality reduction and data denoising. Autoencoders learn data patterns and extract useful features from data automatically. Auto encoders may have following types.

1) Multilayer autoencoder
2) Vanilla autoencoder
3) Convolutional autoencoder
4) Regularized autoencoder
   a) Sparse autoencoder
   b) Denoising autoencoder

In a Multilayer autoencoder there are more than one hidden layers, In Vanilla auto encoder there is only one hidden layer. Both Multilayer and Vanilla autoencoders are fully connected. Convolution autoencoders are used for images means 2D data or even higher dimensional data. These autoencoders are not fully connected instead these are with convolutions. Regularized autoencoders further have two types. First type is spare autoencoder and other is denoising autoencoder. Sparse autoencoders usually learns features from data to perform classification task. The denoising autoencoder is mainly used for removing noise from images and then learn feature of images.

The purpose of this effort is to develop an alternate method for DDoS attack detection using characteristics of neural networks to perform the feature extraction and feature selection and other machine learning techniques such as autoencoder to address safety risks. This work chose a feedforward neural network and another regularized sparse autoencoder which uses a loss function and is used for detection of DDoS attacks as well as it performs classification of these attacks.

II. RELATED WORKS

Hop-count filtering (HCF) is the mechanism to prevent Internet Protocol (IP) Spoofing attack. There is a process of frequently adding/removing network components on a cloud. However, this task of components’ up gradation in a cloud is complex. In a Platform as a Service (PaaS) layer HCF can be used to check which IP addresses are legitimate and which are not. One of the major drawbacks under this mechanism is; the attacker can build its own IP2HC mapping to avoid HCF [4].

Yang et al. [2] proposed a model for the classification of DDoS attacks in a cloud by using a traceback mechanism. This mechanism allows only legitimate requests and blocks all other requests [18]. This traceback technique has a very large number of being false negative and another drawback of this approach is; it uses reactive approach instead of a proactive approach.

Another proposed mechanism of IP Spoofing is trust-based approach in the Infrastructure as a Service (IaaS) to detect those spoofed IP addresses. These addresses are used to access routers, but the process to detecting IP spoofing is specifically in distribution routers [5].

A process of SYN cache approach to overcome the SYN flooding attack, proposed by Lemon is caused by packets flooding to the server and in response the handshaking process becomes incomplete. Increase in latency ratio can overcome this problem.

SYN attacks are detected by another SYN cookies defense approach. When this approach is applied, the performance of cloud based system is degraded [6].

Smurf and Fragile attacks use the same protocol for authentication and are referred as reflection based attacks. The attacker does not send direct request to the server, it sends forged requests. An example of forged request is Internet Connection Messaging Protocol (ICMP) echo request. In response of the forged requests, the server sends response to the victim and in this way it exhausts victim’s resources. This kind of attack is difficult to prevent but can be minimized by two techniques, first by configuring routers in Infrastructure as a Service layer to disable IP-directed broadcast command. It cannot prevent smurf attack; therefore the second technique is to configure the operating system in Platform as a Service (PaaS) layer to prevent this kind of attack [7-10].

Fu and Shi [7] highlighted three mechanisms against the attack of Buffer overflow. In the first mechanism it is checked that at what time the source code can be written in Software as a Service (SaaS) layer. In the second mechanism array bounds are checked in a SaaS layer. The third mechanism is the runtime instrumentation in the SaaS layer. All these defense mechanisms compromise time consumption.

J. Mirkovi’c [3] designed and developed a tool to detect and defend DDoS attacks. This tool is known as D-WARD. This tool restricts users to prevent attacks. It sometimes restricts legitimate users as well. Hence, it has limited adoption [12]. Due to limitations in D-WARD, it is recommended by researchers to defend at destination side by applying some mechanism which is used to protect DoS attacks [13].

Sachdeva et al, [12] discussed several problems which are available in DDoS attacks’ defense. It is indicated that if every request is checked then some cost is used to filter traffic. Sometimes, instead of detecting attacks, some legitimate users requests are also filtered which is not desired [12]. Several researchers developed other approaches to prevent DDoS attacks [25] [14-17]. However, these techniques are not able to alleviate DoS or DDoS attacks [19].

Ali et al. [26] used artificial neural networks (ANN) for denial of service attacks, distributed denial of service attacks and intrusion detection systems. The proposed method used the backpropagation algorithms. The CICID 2017 dataset is used for training neural network. Bayesian regularization detected both DoS and DDoS attacks efficiently with 99.6% accuracy while conjugate gradient descent has 97.7% accuracy.

III. DDoS Attacks

A serious threat to online organizations nowadays is DDoS attack. With these attacks legitimate users get degraded services, due to these attacks, some services are inaccessible [22]. Network congestion from source to destination, disrupts normal Internet operation and denial of services. It is still an open question to detect attack using relevant features. At present machine learning techniques are used to detect attacks. There is need to analyze fundamental features for detecting and classifying the category of attack.

In a DDoS attack, usually some computers are used by attacker to attack other computers or servers. The attacker can take control of any computer, in which multiple hosts attack on a target. Attacker can take hold of any computer and from that computer it sends a large number of requests to a server or it may send large data to several email addresses. If the attacker uses many computers, then the attack is termed as “distributed” [1], [5].

A distributed denial of service (DDoS) attack is created using Internet into furcated end nodes to attack a network. Thousands of computer systems via Internet can be turned into “zombies” and used to attack other systems or website [6], [11]. By considering this it needs to analyze fundamental features for detecting and classifying the attack category.

A distributed solution is required in Distributed P2P networks. DDoS problem is an active area of research that needs to be resolved .When users or hosts are prevented from utilizing a legitimate service provided by a system there emerges a Distributed Denial of Service (DDoS). Widely used method of generating a DDoS in a Distributed P2P network is through artificial exhaustion of resources, such as bandwidth, processor cycles, or memory.

In a Distributed Denial of Service (DDoS) attack, a collective power of many hosts to exhaust the resources of a server system. A robust detection technique is required to detect a DDoS attack with high consistency at an early stage of the attack [23]. DDoS attacks grow in wide variety along with size and frequency. The frequency of today’s DDoS problem demands specialized and systematic attention in order to effectively lessen the attacks. Actually, the DDoS attacks are of type distributed because of denial of services from majorities, and flooding of large quantities of packets.

Referring to discussed process, an attacker can be anyone with a certain knowledge and privileged access to information, with the master host. All he/she has to do is to enter a few commands, and the whole zombie army would awaken up and mount a massive attack against the target of his/her choice. When the DoS attack is originated from various source addresses, it is known as a distributed denial-of-service (DDoS) attack. Typically, spoofed source addresses in a DDoS attack, or the actual addresses of compromised hosts, are used as “zombie agents” to launch the attack.

An attack named blue screen freeze and crash results Bonk/teardrop in TCP/IP. The execution takes advantage of a known bug called “Ping of Death”. Ping of Death attack normally attacks on TCP/IP; the attacker uses the ping system utility to make up an IP packet and when the system receive a huge number of data such as 65,536 bytes allowed by the specified IP, may crash or reboot.

IV. Feed Forward Neural Networks

In a feed forward neural network information enters from the input layer then it is passed to some hidden layers and then finally it reaches to the output layer. The flow of information is straight from layer to layer and there is no feedback loop. Figure 1 shows the architecture of a feed forward neural network.

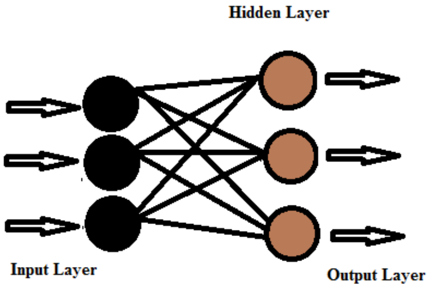


Figure 1. Feed Forward Neural Network Architecture

V. Deep Learning Architectures

Machine learning has a subset named Deep learning that has networks which are capable of learning from unstructured data. There are several types of deep learning architectures. Some popular are enlisted here;

- Unsupervised Pretrained Networks (UPNs)
- Convolutional Neural Networks (CNNs)
- Recursive Neural Networks
- Deep stacking networks
- Long short-term memory / Gated recurrent unit / Recurrent Neural Networks

Unsupervised Pretrained Networks are further categorized in three types of networks;

- Autoencoders Neural Networks
- Deep Belief Networks (DBNs)
- Generative Adversarial Networks (GANs)

Autoencoders are a type of feed forward neural networks used as dimensionality reduction algorithms.

An autoencoder neural network has input layer, a hidden layer also known as encoding layer and an output layer called decoding layer. Preserving information as much as possible when an encoder/decoder pair runs input, but autoencoders also help in representing various properties of a dataset. Autoencoders give rise to their own labels from the training data.

Autoencoders automatically learn from data examples thus, to train specialized instances of the algorithm that will perform well on a specific type of input becomes easy without any tiring effort, just the appropriate training data. Autoencoders are data-specific by nature which means they learn specific features of training data. An autoencoder trained for handwritten digits cannot be used for landscape images. Autoencoders are unsupervised machine learning algorithms.

If an autoencoder has multiple hidden layers it is known a deep autoencoder. It is essential for deep autoencoders that the layer sizes have to be symmetric. By symmetric it means, the last and first layer has to have the same size, similarly the second last layer and the second layer has to have the same size likewise, corresponding layers must continue this symmetry which is essential for pre-trained networks. Deep autoencoders can be trained as a stack of single layer autoencoders.

A. *Autoencoder Architecture*

Autoencoder architecture is shown in figure 2. It has encoder, code and decoder. Here both encoder and decoder are fully-connected feed forward neural networks. Code is a single layer artificial neural network.

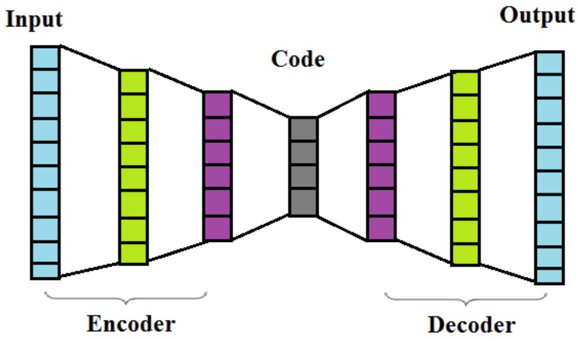


Figure 2. Auto Encoder Architecture

This is a detailed visual representation of an autoencoder. At first, the input is supplied to encoder, which is a fully-connected Artificial Neural Network (ANN), to generate the code. The decoder, containing architecture similar to ANN structure, produces the output only using the code. The aim is to get an output identical with the input. The decoder architecture is the mirror image of the encoder. It is just a typical case not a requirement. There lies the restriction of the dimensionality of input and output to be the same. Anything in the middle can be played with. The loss function maps a 28x28 grid to real numbers like 784.

Before training an autoencoder, following four hyperparameters are needed to be set:

Code size: Number of nodes in the middle layer. More compressions are due to smaller size.

Number of layers: Depth of autoencoder can be of your choice. In the figure above we have two layers; the encoder and decoder, rather taking input and output into account.

Number of nodes per layer: The autoencoder architecture considered here is called a stacked autoencoder since; the layers are stacked one over other. The stacked architecture of autoencoders looks like a "sandwitch". Nodes per layer decreases with each subsequent encoder layer, and increases back in the decoder. Symmetry follows in terms of layer structure for both encoder and corresponding decoder. As noted above this is not necessary and there is flexibility in parameters handling.

Loss function: we consider mean squared error (mse) or binary cross entropy. If the input values are in the limits [0, 1] then we use cross entropy, else the mean squared error.

Autoencoders are trained via back-propagation similar the way as ANNs.

VI. Experimental setup

In this research work KDD-CUP99data set is used. It contains data which includes both normal requests and DDoS attacks. KDDCUP99 is accessible via MIT Lincoln lab which is located in Lexington, Massachusetts. It is a research center of United States department of Defense. The datasets consists of several records, each record has 42 fields. Among these fields first 41 fields indicates the features such as duration, protocol_type, service, flag, src_bytes, dst_bytes, etc. The last field indicates the attack. There are several attacks types such as neptune, smurf, teardrop etc. The KDD data set contains both symbolic and continuous data. To make it Matlab compatible, symbolic data fields are replaced with the continuous data. The symbolic data includes protocol types, service types, flags and attacks type. Protocol types and service types are replaced with values like tcp=1, udp=2, icmp=3, http=4, smtp=5. In the data set there are more than 65 protocol types. There are total ten error flags and are replaced with values like S0=1, SF=2, S1=3, REJ=4, S2=5, RSTO=6, S3=7, RSTR=8, SH=9, and OTH=10. There are more than twenty attacks types and are replaced with values like normal=1, smurf=2, neptune=3, teardrop=5, pod=6, land=7, etc. Here we design two types of neural networks (1) Feedforward Neural Netowkrs with 20 neurons in hidden layer and five neurons in output layer. In hidden layer a sigmoid transfer function is used while in output layer a linear transfer function is used. (2) Autoencoder is rained with hidden layer of size 20 neurons. For decoder a linear transfer function is used. L2 weight regularizer is set to 0.001, sparsity proportion is set to 0.05 and sparsity regularizer is set to 4.

A. Feedforward Neural Network

The first neural network is designed using feedforward neural networks using one hidden layer with 20 neurons and sigmoid transfer function [20] and an output layer with 5 neurons and a linear transfer function as shown in figure 3. Since majority of the requests include either normal or, neptune and smurf attacks. Here normal indicates that there is no attack. Therefore, for simplicity the attacks are classified in five classes. Therefore the output layers consist of only five neurons.

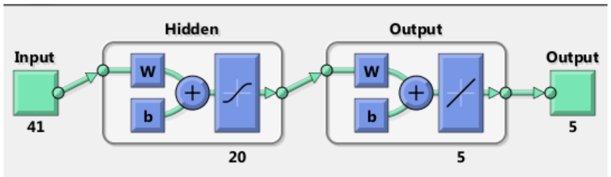


Figure 3. Feedforward Neural Network

B. Deep network using Autoencoders

As shown in figure 4 Autoencoder consists of both encoder and decoder. Encoder has 20 neurons with sigmoid transfer function while decoder has 41 neurons with linear transfer function. Since in autoencoders the input is the same as output. Therefore we have 41 inputs and 41 outputs. This autoencoder performs the features learning task. However it is not sufficient to provide reliable accuracy using only this layer.

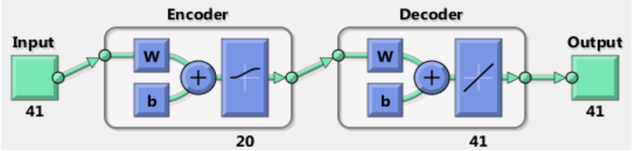


Figure 4. Autoencoder with 41 inputs and 41 outputs

In order to get more accurate predictions accuracy, Another layer of auto encoders is designed which has 20 neurans in encoder and 20 neurans in decoder. As shown in figure 5, there are 20 inputs as well as 20 outputs. Here is the architecture of the second autoencoder which learns uses the features already learned by the first layer of autoencoder and also learns more features.

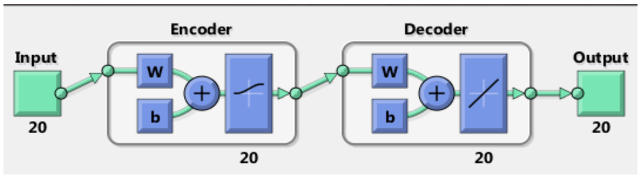


Figure 5. Autoencoder with 20 inputs and 20 outputs

After these two autoencoders a softmax layer is designed. Fig 6 shows the architecture of the softmax layer. Softmax layer has twenty inputs and five outputs. It has total 5 neurons. The softmax layer normalizes the outputs obtained from autoencoders.

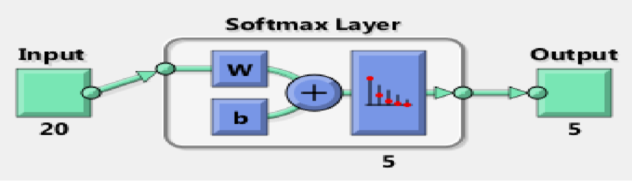


Figure 6. Softmax layer with 20 inputs and 5 outputs

Finally, stack the encoders and the softmax layer to form a deep network as shown in figure 7. It has 41 inputs which are the features in the KDD CUP 99 data set and 5 outputs which are classes of attacks.

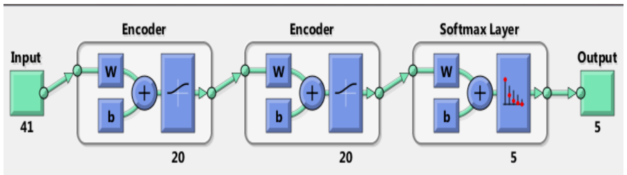


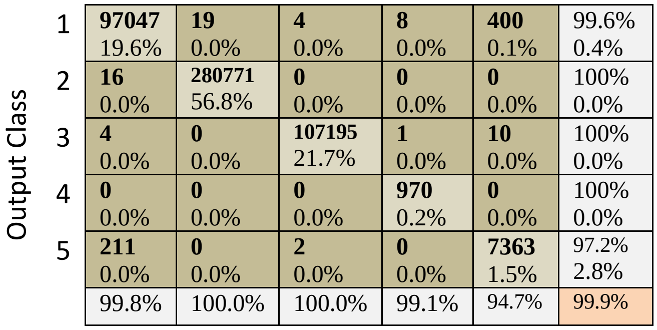
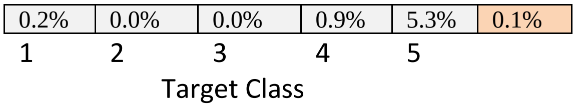
Figure 7. Deep network with 41 inputs and 5 outputs

In the KDD Cup 99 data set there are total 22 classes of attacks. However in this research these classes are merged and only five classes are considered.

VII. RESULTS

Here a confusion matrix is shown in figure 8 to describe the performance of FeedForward Neural Network classifier on KDD-CUP99 data set for which the actual values are known. In KDD-CUP99 there are more than twenty classes of attacks. For simplicity, in this paper these attacks are classified in five categories. The first category shows the normal requests which means that there is no attack. The second category shows Neptune attacks, third category shows smurf attacks fourth category shows teardrop and fifth category indicates all other attacks.

Figure 8 shows that for the first class which is the class of normal requests there are total 231 misclassifications. For Class 2 there are total 19 misclassifications. For third class there are total 6 misclassifications. For fourth class there are total 9 misclassifications and for fifth class there are total 410 misclassifications.

Output Class

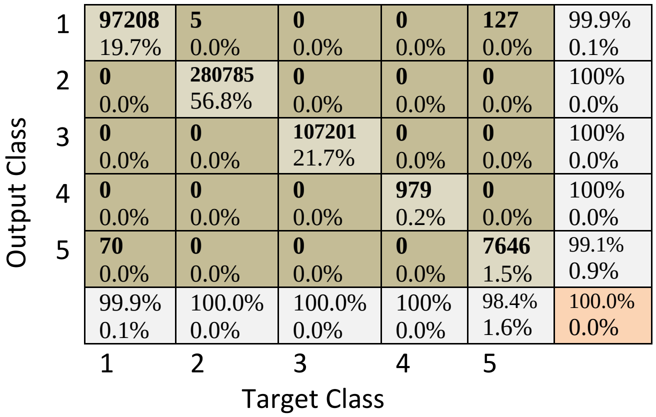
	1	2	3	4	5	
1	**97047** 19.6%	**19** 0.0%	**4** 0.0%	**8** 0.0%	**400** 0.1%	99.6% 0.4%
2	**16** 0.0%	**280771** 56.8%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
3	**4** 0.0%	**0** 0.0%	**107195** 21.7%	**1** 0.0%	**10** 0.0%	100% 0.0%
4	**0** 0.0%	**0** 0.0%	**0** 0.0%	**970** 0.2%	**0** 0.0%	100% 0.0%
5	**211** 0.0%	**0** 0.0%	**2** 0.0%	**0** 0.0%	**7363** 1.5%	97.2% 2.8%
	99.8% 0.2%	100.0% 0.0%	100.0% 0.0%	99.1% 0.9%	94.7% 5.3%	99.9% 0.1%

Target Class

Figure 8. FeedForward Neural Networks Confusion Matrix

In the confusion matrix 19.6% shows that among all requests in the data set, 19.6% belongs to first class i.e. normal requests and are correctly classified. In the seconds row and second column 56.8% shows that among all requests available in the data set 56.8% requests belongs to class two i.e. Neptune attacks and are correctly classified. In the third column and third row the number 21.7% shows that these requests belongs to the third class and are correctly classified. Also it indicates that 21.7% of the whole data set belongs to the third class.

Figure 9 shows the confusion matrix for Autoencoders. Accuracy obtained by using autoencoders is 100%. Although there are some misclassifications in first class, second class and fifth class. However the number of misclassifications is very small. The results in confusion matrix indicates that first, second and fifth class have 70, 05and 127 misclassifications respectively.



Output Class

	1	2	3	4	5	
1	**97208** 19.7%	**5** 0.0%	**0** 0.0%	**0** 0.0%	**127** 0.0%	99.9% 0.1%
2	**0** 0.0%	**280785** 56.8%	**0** 0.0%	**0** 0.0%	**0** 0.0%	100% 0.0%
3	**0** 0.0%	**0** 0.0%	**107201** 21.7%	**0** 0.0%	**0** 0.0%	100% 0.0%
4	**0** 0.0%	**0** 0.0%	**0** 0.0%	**979** 0.2%	**0** 0.0%	100% 0.0%
5	**70** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**7646** 1.5%	99.1% 0.9%
	99.9% 0.1%	100.0% 0.0%	100.0% 0.0%	100% 0.0%	98.4% 1.6%	100.0% 0.0%

Target Class

Figure 9. Autoencoders Confusion Matrix

These results indicate that there are more misclassifications when FeedForward Neural networks are used as compared with the autoencoders. This is due to the fact that autoencoders have more hidden layers to extract features at multiple levels. For feed forward neural network the total number of misclassifications for all classes is 675 which is approximately 0.1% of the requests available in the KDD CUP dataset. Autoencoders classified all attacks types successfully, there are only 202 misclassifications.

VIII. CONCLUSION AND FUTURE WORKS

In this paper we have compared performance of FeedForward Neural networks and autoencoders to detect and classify DDoS attacks. Since autoencoders use multiple layers, therefore they perform the feature learning in more efficient ways as compared with the feature learning with

FeedForward Neural networks. Performance accuracy of autoencoders is 100% while the performance accuracy of feedforward neural ntwork is 99.9%.

Future researchers can use other type's data sets and can develop various architectures of autoencoders to test accuracy of these deep networks.

ACKNOWLEDGMENT

The authors wish to acknowledge Quaid e Awam University of Engineering Science and Technology, Nawabshah, Pakistan, for providing the technical facilities to carry out this research work.

REFERENCES

[1] Kilari, N., &Sridaran, R. (2015). An Overview of DDoS Attacks in Cloud Environment. International Journal of Advanced Networking & Applications.
Akamai (State of the Internet), https://www.akamai.com/us/en/multimedia/ Documents/state-of-the-internet/akamai-q2-2016-state-of-the-internet-security-report.pdf (Accessed on 4.10.2016).

[2] Lanjuan Yang, Tao Zhang. Jinyu Song, Jinshuang Wang and Ping Chen, "Defence of DDoS attack for cloud computing", In Computer Science and Automation engineering, 2012 IEEE International Conference, vol 2, pages 626-629, 2012.

[3] J. Mirkovic and P. Reiher, A taxonomy of DDoS attack and DDoS defense mechanisms, ACM SIGCOMM Computer Communications Review, vol. 34, no. 2, pp. 39-53, April 2004.

[4] T. Peng, C. Leckie, and K. Ramamohanarao, Survey of network-based defense mechanisms countering the DoS and DDoS problems, ACM Comput. Surv. 39, 1, Article 3, April 2007.

[5] C. Douligeris, and A. Mitrokotsa, DDoS attacks and defense mechanisms: classification and state-of-the-art, Computer Networks, Vol. 44, No. 5, pp. 643-666, April 2004.

[6] Darwish, M., Ouda, A., &Capretz, L. F. (2013, June). Cloud-based DDoS attacks and defenses. In Information Society (i-Society), 2013 International Conference on (pp. 67-71). IEEE.

[7] D. Fu and F. Shi, "Buffer Overflow Exploit and Defensive Techniques," 2012 Fourth International Conference on Multimedia Information Networking and Security, pp. 87-90, Nov. 2012.

[8] Wang, B., Zheng, Y., Lou, W., &Hou, Y. T. (2015). DDoS attack protection in the era of cloud computing and software-defined networking. Computer Networks, 81, 308-319.

[9] N. McKeowen et al., "OpenFlow: Enabling Innovation in Campus Networks," OpenFlow white paper, 14 Mar. 2008, http://www.openflow.org//documents/openflowwp-latest.pdf, accessed 10 Oct. 2016.

[10] Siamak Azodolmolky, Philipp Wieder, Ramin Yahyapour. (2013). Cloud computing networking: challenges and opportunities for innovations. IEEE Communications Magazine, 51(7), 54-62.

[11] J. Mirkovi´c, "D-WARD: DDoS Network Attack Recognition and Defense," 2002.

[12] M. Sachdeva, G. Singh, and K. Kumar, "Deployment of Distributed Defense against DDoS Attacks in ISP Domain," International Journal of Computer Applications, vol. 15, no. 2, pp. 25–31, February 2011, published by Foundation of Computer Science.

[13] F. Kargl and J. Maier, "Protecting web servers from distributed denial of service attacks," 2001.

[14] T. H. Nguyen, C. T. Doan, V. Q. Nguyen, T. H. T. Nguyen, and M. P. Doan, "Distributed defense of distributed DoS using pushback and communicate mechanism," in Proc. Int Advanced Technologies for Communications (ATC) Conf, 2011, pp. 178–182.

[15] S. M. Khattab, C. Sangpachatanaruk, R. Melhem, D. l Mosse, and T. Znati, "Proactive server roaming for mitigating denialof-service attacks," in Proc. ITRE2003 Information Technology: Research and Education Int. Conf, 2003, pp. 286–290.

[16] L. M., W. C.-H. J., J. Y. Hung, and I. J. D., "Mitigating performance degradation of network-based control systems under denial of service attacks," in Proc. 30th Annual Conf. of IEEE Industrial Electronics Society IECON 2004, vol. 3, 2004, pp. 2339–2342.

[17] A. K. Pandey and C. Pandu Rangan, "Mitigating denial of service attack using proof of work and Token Bucket Algorithm," in Proc. IEEE Students' Technology Symp. (TechSym), 2011, pp. 43–47.

[18] Vashisht, Shikha, and Mandeep kaur. "A Novel Active Data Filtration for the Cloud based Architecture against Packet Flooding Attacks", International Journal of Modern Education and Computer Science, 2015.

[19] Shtern, Mark, Roni Sandel, Marin Litoiu, Chris Bachalo, and Vasileios Theodorou. "Towards Mitigation of Low and Slow Application DDoS Attacks", 2014 IEEE International Conference on Cloud Engineering, 2014.

[20] Paul Rubel. "Web and Grid Services for Improving Ambient Intelligence Embedded in Pervasive, Personal ECG devices", 2008 International Conference on Complex Intelligent and Software Intensive Systems,
03/2008

[21] Xiapu Luo, R.K.C. Chang, E.W.W. Chan. "Performance Analysis of TCP/AQM Under
Denial-of-Service Attacks" , 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2005

[23] Paulo M. Mafra. "Octopus-IIDS: An anomaly based intelligent intrusion detection system", The IEEE symposium on Computers and Communications, 06/2010
[24] T. Subbulakshmi. "Feature Selection and Classification of Intrusions Using Genetic Algorithm and Neural Networks" , Communications in Computer and Information Science, 2010

[25] Y. Xie and S.-Z. Yu, "Monitoring the application-layer ddos attacks for popular websites," Networking, IEEE/AcM Transactions on, vol. 17, no. 1, pp. 15–25, 2009.

[26] Ali, O., & Cotae, P. Towards DoS/DDoS Attack Detection Using Artificial Neural Networks. In 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (pp. 229-234) 2018