Clinical Cancer Investigation Journal-ISSN: 2278-0513

Representation of A Modern Strategy for Identifying Bot-Net Attacks in Cloud Computing Using the Neural Network

Abstract

A bot is a computer infected by malware, controlled remotely by one or more human factors without any user knowledge and will. This research offers a modern strategy for identifying botnet attacks in cloud computing using the neural network (NN). The genetic algorithm has used an improved NN method to detect botnet attacks in this study. In this way, it is noted that the proper characteristics were selected for this. According to historical studies, the features such as Average Flow Size, Packet Average Size, Average Number of Packets, different numbers of flows to the safe destination IP, number of flows to distinctive destination Ports, SYN-SYN/ACK, and Land were chosen. Regression exploration was used to explore the correlation rate between different features and the possibility of botnet attacks. The possibility of botnet attacks is highest for the Average Flow Size. After exploring the correlation of input/output features, Graph Neural Network (GNN), Radial Basis Function (RBF), and Support Vector Machine (SVM) were classified. The best quality of its classification is GNN.

Keywords: *Botnet Attacks, Cloud Computing, Neural Networks, Genetic Algorithm*

Maryam Afshun*

Master of Science Software engineering Department of Industrial engineering, faculty of engineering, Baft branch, Islamic Azad University, Kerman, Iran

mhd.mtd3132@gmail.com

Introduction

Someone calls cloud computing a pattern variation, which tracks the change of the large computer model, at the beginning of the 1980s, to a user/employer one. Hi-tech information requires computing work everywhere and all the time.[1] Also, people must be able to do their heavy computing work without having expensive hardware and software. Cloud computing has been the last response to these needs. Since this technology is currently in its onset, there is no popular standard scientific definition, but most experts agree with some parts.[2] The National Institute of Standards and Technology (NIST) defines cloud computing as "a model for providing easy access based on the user demands through the network to the variable computing resources and configuration (such as networks, servers, storing space, application programs, and services) so that this access can leave the lowest need to the resource management or direct involvement of a service provider quickly. [3]This cloud computing model has five main features, three service models, and four spread models (deployment). The architecture of the software platforms involved in presenting cloud computing usually consists of the components that connect through the software programming link and mostly the web service. This design is similar to Unix, in which multiple programs, doing well, work with each other through the world interface. These platforms are more controllable than the same integrated ones and are complexly handled.[4]

Botnets are the newest type of malware on the internet scale that have threatened internet platforms so much during recent years. A bot is a computer infected by malware without any user's knowledge and will and controlled by a human factor or more. [5]This factor is called the controlling factor or bot manager, and sometimes the infected system is known as a victim. When doing disruptive software, the computers on a botnet can decide together. They can do this by tracking the users into creating a drive using a download, gaining from a web browser vulnerable, or through a Trojan horse in an attached file. This malware usually installs the modules which cause the computer to be commanded and controlled by a botnet operator.[6] A Trojan may delete itself concerning what has been written or stay to upgrade and maintain the modules. The architecture of the botnet has evolved; all botnets do not use the topology of control and command. Advanced topology has more flexibility than shutdown, counting, or detection. However, some topologies limit the marketability to third parties. The common topologies include the star botnet, multiple services, hierarchical, and occasional. [7]

Choi et al. have proposed an unsupervised online method to detect botnets called Botnet Group Activity Detector (BotGAD). First, they define a group activity as a vital feature of the botnet and propose measures for detecting the botnet and monitoring the group activities in DNS traffic. [8]Wang et al. have proposed a system based on the behavior to detect botnets. This technique is based on fuzzy pattern recognition techniques and uses single-level analysis. The main idea of their method is based on the detection of domain names and malevolent IP addresses used by the botnet. Yahyazadeh et al. have proposed an unsupervised online method called BotOnus to detect different botnets. Gu et al. have proposed an inactive network supervised system called BotHunter, which focuses on detecting communication-related to the infection and synchronization occurring during the successful infection of the malware to identify the hosts infected by a bot immediately.[9]

Botnet detection is a novel research area with a challenge in computer network security. In recent years, various methods have been proposed for botnet detection. However, they mostly have constraints such as depending on a structure and a specific protocol for the command and control channels, lack of the ability to identify at the beginning of the botnet life cycle, being offline, and the need for the labeled data for learning.[10] The evolution path of botnets has been based on their command and control channels. Therefore, the ongoing botnets can be predicted by modeling all their state spaces to some extent. This is done by considering five significant factors affecting the command/control channels: links, adjoin mechanisms, communication protocol, control mechanism, and command authentication mechanism. The methods for detecting the coming botnets can be proposed by their modeling to act efficiently. The detection methods of botnets describe their structure in detail in different studies. So, there is always this danger, and the attackers offer different escape techniques to them. On the other hand, the ongoing botnets are likely to gain from different techniques to escape detection. Thus, one of the most existing challenges will be enhancing the proposed methods as much as possible against various escaping techniques to detect th botnets,

Since the use of clouds has expanded nowadays, their security discussion is of importance. Also, botnets have a high and almost imperceptible influence. Therefore, we offer a modern strategy for detecting botnet attacks in cloud computing using NN. Hence, the current study aims to offer a modern strategy to identify and detect botnet attacks in cloud computing with the help of NN.

Research method

This study aims to identify the botnet attacks using NN improved by the genetic algorithm. In general, this method is implemented in Figure 1.

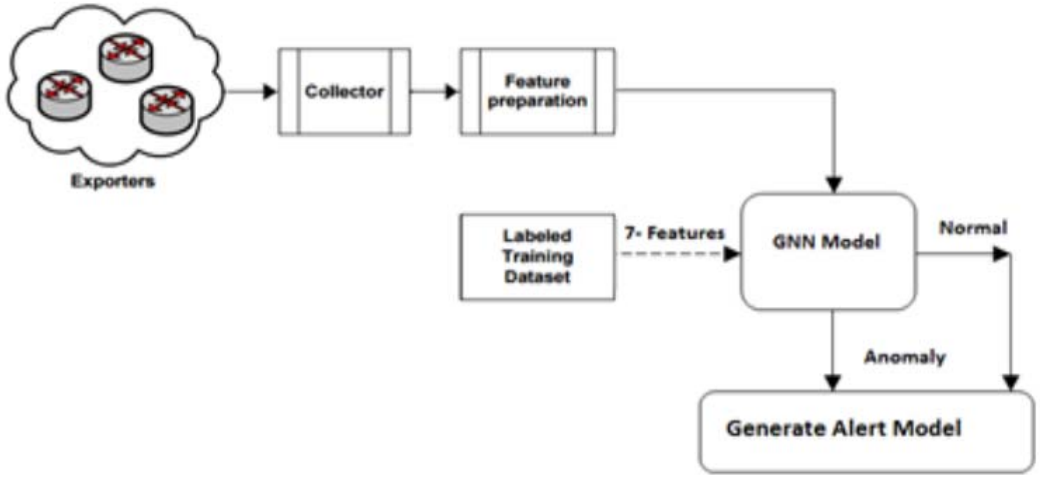

Figure 1: The method for implementing the plan.

In the first stage, the data related to the botnet attacks are recorded, and their features are extracted from them. Then, the model was created using NN improved by GNN, and the labeled data and the precision values were determined.

A local network based on the depicted topology in Figure 2 was built to gather accurate testing data. To utilize this topology, it is of utmost necessity that the normal agent traffic targeted samples were gathered with the size to cover the agent traffic during different 24 hours. In this way, the CE agent traffic has been gathered in this paper.

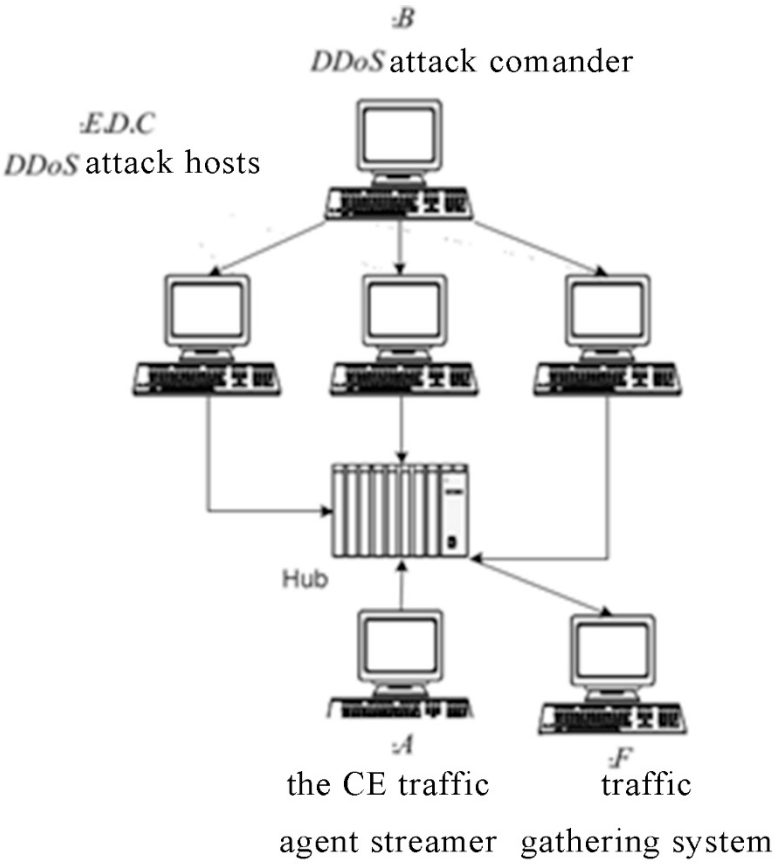

Figure 2: The network topology for accumulating the study data.

The proposed topology in Figure 3 involved six computers, and 5 joined each other by a hub. Machine A flows the accumulated traffic of machine CE to the network. Machine B exports the order for beginning the attacks to machines C, D, and E. Once these machines receive the order, they begin their attacks on the agent representative system, i.e., Cyber Essential (CE). When this network begins to act, machine F accumulates the flowing traffic with the help of host attack machines, i.e., distributed denial-of-service (DDoS), and the CE machine as mixed. If there is no attack on this network, machine F only accumulates the traffic of machine CE. In this way, eventually, the accumulated traffic of machine F includes both the regular traffic of machine CE and the attack of DDoS created on this machine.

To generate the training data, the long-term traffic of the CE machine was accumulated. Then, several different DDoS attacks were created on the accumulated traffic using the mentioned method. To test the trained network, the testing traffic was also produced using this method by various attacks created on the traffic of the CE machine accumulated in another tie with a lower volume.

Findings

1. Feature selection

The most important state for building a system that can detect botnet attacks is how to extract and choose the main features/features of these. The most important features of this data involve the seven parameters below that we used in this study.

- Average Flow Size: this feature provides valuable and essential value for abnormal events such as port scans.
- Packet Average Size: this feature determines the size of the sent/received packet; if small, it can express an abnormal event.
- Average Number of Packets: One of botnet attacks' primary features is to change the source IP forever, which can cause trouble for the primary source.
- Different flow numbers to the same destination IP: This feature determines the number of flows sent to a destination IP. If the number of flows is high, it can refer to an attack.
- Number of flows to distinctive Destination Ports: this feature can affect the attack differentiation. If the number of flows passing through the port is high, it can represent an event for an attack on a port scan.
- SYN - SYN/ACK: this feature has been utilized by many scholars so that the number of SYN and that of SYN/ACK are compared.
- Land: this feature determines whether an attack has been created or not.

So, based on the 833 data in this study, we first chose seven features above and then normalized them, as depicted in Figure 3.

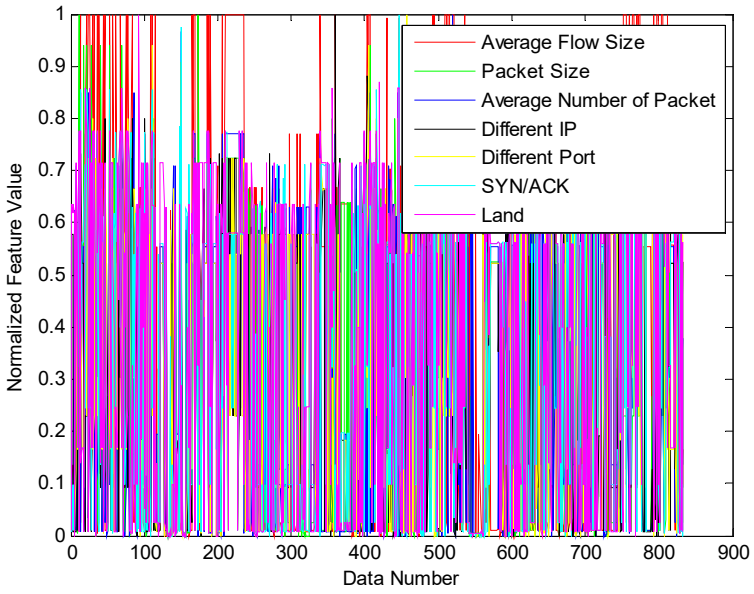

Figure 3: Chart of normalized characteristics.

2. Data Regression

The relationship between the inputs has been depicted in Figures 4-10, respectively, with the possibility of a botnet attack, which is as follows:

1. Average Flow Size;
2. Packet Average Size;
3. Average Number of Packets;
4. Different numbers of flow to the same destination IP;
5. Number of flows to distinctive Destination Ports;
6. SYN - SYN/ACK;
7. Land.

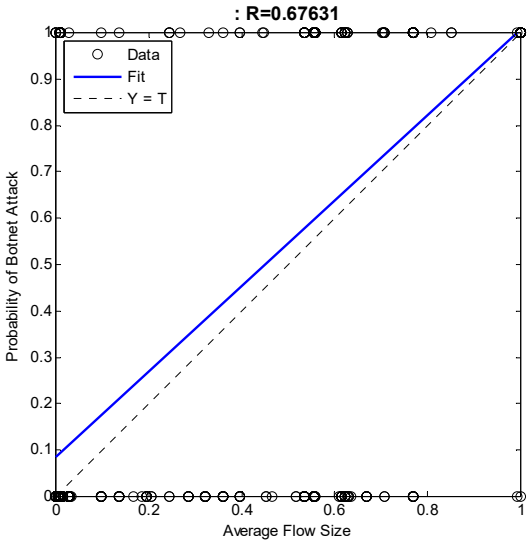

Figure 4. The regression of the botnet attack possibility and the Average Flow Size.

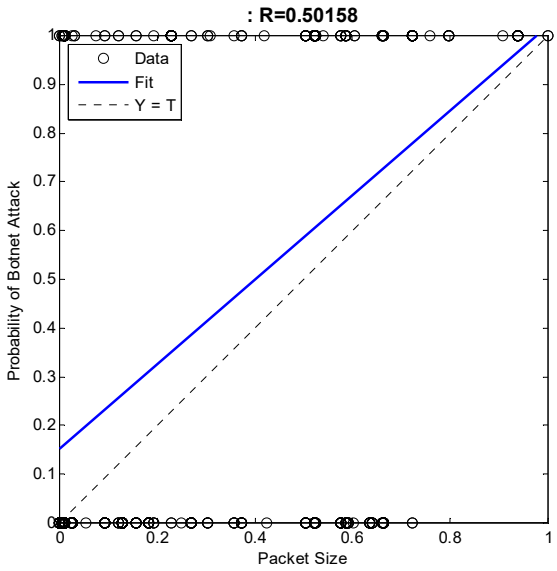

Figure 5. The regression of the botnet attack possibility and the Packet Size.

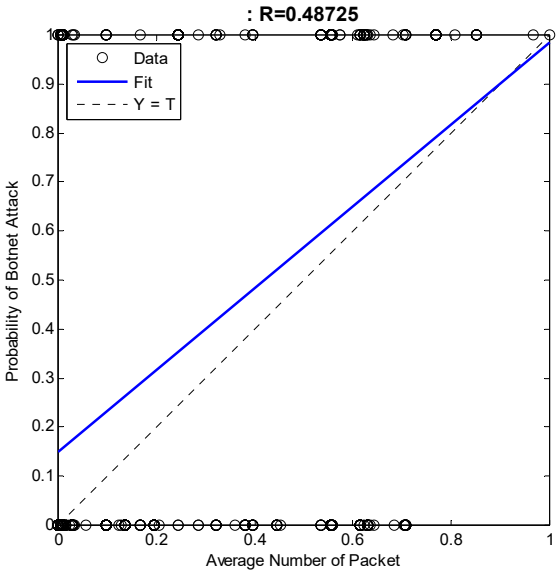

Figure 6. The regression of the botnet attack possibility and the Average Number of Packets.

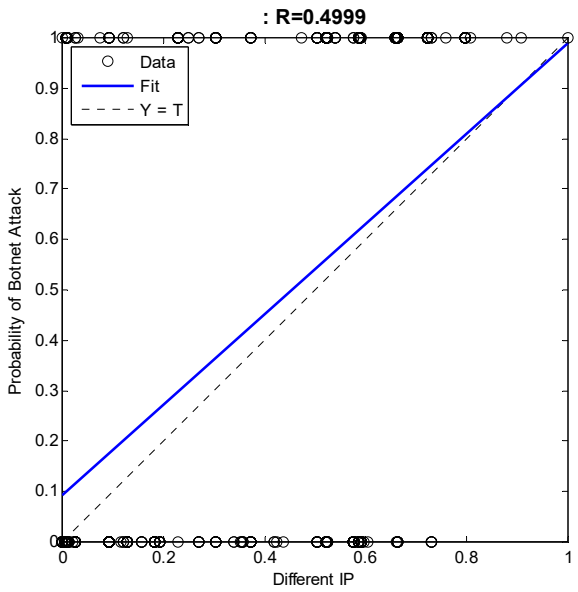

Figure 7. The regression of the botnet attack possibility and Different IPs.

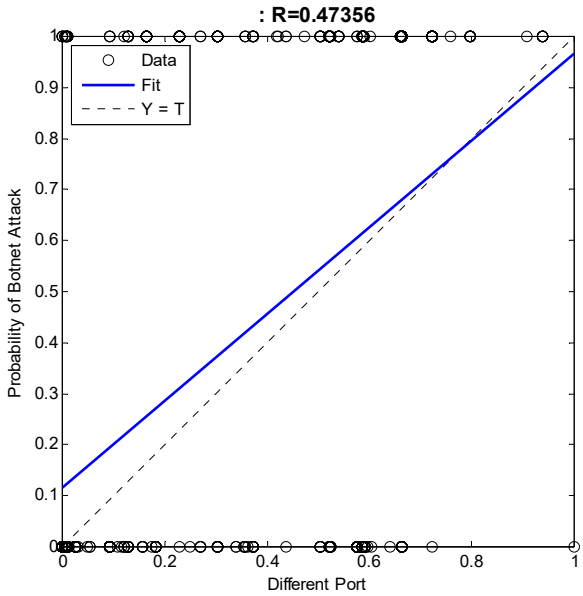

Figure 8. The regression of the botnet attack possibility and Different Port.

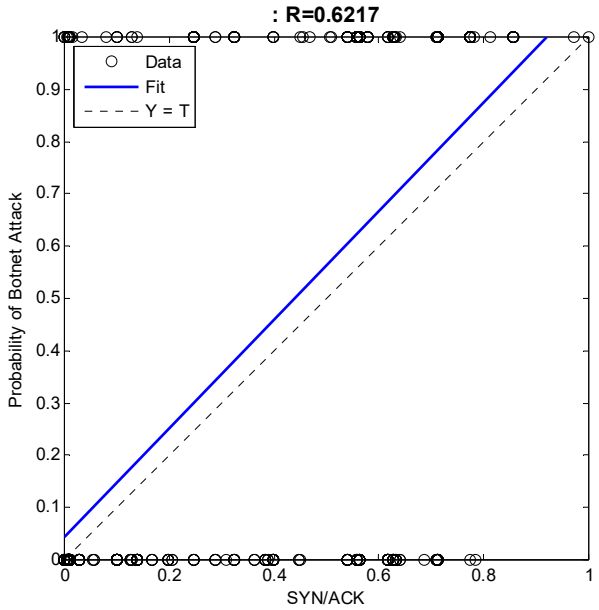

Figure 9. The regression of the botnet attack possibility and SYN/ACK.

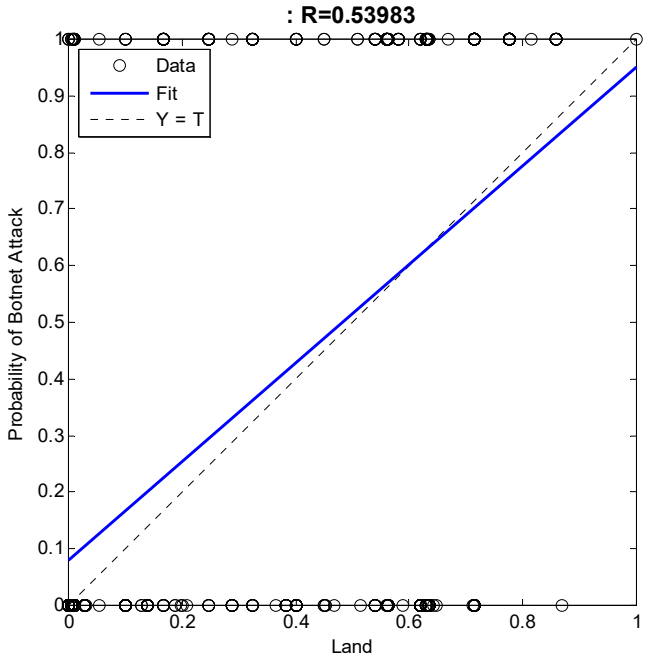

Figure 10. The regression of the botnet attack possibility and LAND.

It obtained an R-value for each character representing the correlation between input and output in the regression chart. This value has been obtained for each feature in Table 1.

Table 1: The rate of R for various features.

Land	SYN/ACK	Different Port	Different IP	Average Number of Packets	Packet Size	Average Flow Size	
0.54	0.62	0.47	0.50	0.48	0.50	0.67	R-value

According to the correlation values between different features and the possibility of botnet attacks, the Average Flow Size has the highest effect on the attack possibility, and the lowest is related to the Different numbers of flow to the same destination Port.

3. Classification results

Three methods were used for classification, including GNN, RBF, and SVM. This way, data were divided into two classes: testing and training. The training data includes 733 data, and the remaining was used for testing. Then, the Accuracy, Precision, and Recall values were measured for testing and training states. A Multilayer Perceptron Neural Network (MLP NN) with a hiding layer and five neurons has been used for the targeted GNN. In Table 2, the results of values have been shown for testing and training the NN.

Table 2: The genetic algorithm improved classification results using the NN.

Recall	Precision	Accuracy	
99.64%	100%	99.86%	Train
100%	100%	100%	Test

To compare the strength of this proposed system with other methods, the classification was compared with other modules such as RBF and SVM. The results are shown in Tables 3 and 4.

Table 3: The classification results using the RBF.

Recall	Precision	Accuracy	
84%	84%	84%	Train
85%	83%	84%	Test

Table 4: The classification results using the SVM.

Recall	Precision	Accuracy	
91%	94%	95%	Train
92%	90%	95%	Test

Comparing Tables 2 and 3, it is concluded that the NN method improved by the genetic algorithm depicts a higher strength than others.

Conclusion

In this study, the improved NN method has been used by GNN to optimize the identification of botnet attacks. It was necessary to select the proper characteristics for this. Based on the previous studies, the characteristics below were used.

Average Flow Size, Packet Average Size, Average Number of Packet, Different numbers of flow to the same destination IP, Number of flows to distinctive Destination Ports, SYN - SYN/ACK, and Land.

To explore the correlation rate between different features and the possibility of a botnet attack, a regression exploration was used so that the possibility of a botnet attack for the Average Flow Size is the highest. After reviewing the input/output correlation, the GNN, RBF, and SVM were classified. The highest quality of the classification is related to the GNN.

According to the results of the composing genetic algorithm and fuzzy logic, it is proposed that the GNN is used to locate the botnet attacks in the network, and then these attacks are modeled by the cellular automata.

Acknowledgments:Non
Conflict of Interests: Non
Ethical Considerations: Non
Financial Disclosure: Non
Funding/Support: Non

Reference

1. B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman, S. Li, “Incremental learning for v-Support Vector Regression”, Neural Networks, 2015, 67:140-150, 13, 03, (2015).
2. J. Wang, Y. Yin, J. Zhang, S. Lee, and R. Simon Sherratt, “Mobility-based energy efficient and multi sink algorithms for consumer home networks”, IEEE Transactions on Consumer Electronics, 59, 1, (2013).
3. KERAS Development Team, “Keras: Deep Learning library for Theano and TensorFlow,” 2016. [Online]. Available: https://keras.io
4. Stratosphere IPS Project, “Stratosphere Project,” 2015. [Online]. Available: https://stratosphereips.org
5. Yu Zhao, “Novel approach of P2P Botnet Node-based detection and applications”, “Journal of Chemical and Pharmaceutical Research”, 2014, vol. 6, no. 7,: (2014), pp 1055-106.
6. C. Yin, “Towards accurate node-based detection of P2P botnets”, “Scientific World Journal”l, 2014 ,24 , :June, (2014). pp .425491-425491.
7. S. Garcia, “Identifying, Modeling and Detecting Botnet Behaviors in the Network,” Ph.D. dissertation, UNICEN University, 2014.
8. S. Garcia, “Modelling the Network Behaviour of Malware To Block Malicious Patterns . The Stratosphere Project: a Behavioural Ips,” in Virus Bulletin, no. September 2015, pp. 1–8. [Online]. Available: https://www.virusbtn.com/pdf/conference slides/2015/GarciaVB2015.pdf
9. B. Gu, V. S Sheng, K. Yeow Tay, W. Romano, and S. Li, “Incremental Support Vector Learning for Ordinal Regression”, “IEEE Transactions on Neural Networks and Learning Systems”, 2015, vol. 26 ,no.7, Aug. 12, (2014), pp. 1403-1416 ,
10. Garcia, Sebastian, “Malware Capture Facility Project,” 2013. [Online]. Available: https://mcfp.felk.cvut.cz/