*IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555*
*Vol.6, No1, Jan-Feb 2016*

# Mining Based Detection of botnet traffic in Network Flow

P.Kalaivani
Research scholar
PSGR Krishnammal College for women
Coimbatore, India
kalainisa@gmail.com

M.S.Vijaya
Associate Professor
Department of computer science
PSGR Krishnammal College for women
Coimbatore, India
msvijaya@psgrkc.com

*Abstract*— A significant threat to any network is the presence of botnet traffic. Traffic originated from botnet is called as botnet traffic which performs malicious activities like distributed denial of service (DDOS), mass spam, click fraud, as well as password cracking via distributed computing and other forms of cybercrime. Identification of botnet traffic from normal traffic in the massive internet traffic is one of the most trivial tasks as they are reusable and renewable resources. The aim of this research is to analyze and identify such destructive botnet traffic. The categorization of network traffic is carried out by implementing powerful machine learning algorithms like support vector machine, naïve bayes, decision tree, and neural networks were applied to train the model. The performance of the model was evaluated using 10 fold cross validation.

***Keywords—Botnet traffic; Network Security; Machine Learning.***

## I. INTRODUCTION

A major threat to the network is the presence of botnet and it is the predominant threat on the Internet today. Botnets are designed to operate in the background, often without any visible evidence of their existence. Over the past decade botnets are heavily used for all kinds of computer crimes such as phishing, distributing pirated media and software, identity theft, adware, stealing information and computer resource and so on. Majority of these attacks are focused on making money through illegal means. Hence finding ways to counter botnets is a challenge of great importance. Botnet detection typically demonstrate uniformity of traffic behavior, present unique communications behavior, and that these behaviors may be characterized and classified using a set of attributes which distinguishes them from normal traffic. Therefore botnet traffic detection is a significant task in any communication network environment.

As the fast development of information technology, the population of using the Internet increases rapidly. Due to the prevalence of social networks, a huge amount of data are transmitted via networks in every second, causing the potential risks of disclosing personal information. A hacker steals confidential data for illegal usage, and they may use a variety of methods such as Distributed Denial of Service (DDoS), Spam and Trojan. These methods require the cooperation of many computers, so hackers often spread out malicious software to achieve the goal of attacks. Educational resources (.edu) are high level of target as they are poorly secured, and government-military systems are also popular target as access to information and high tech resources. Therefore, it is important to enhance network security while developing and applying new botnet detection techniques to prevent from illegal access to confidential information.

Many research works have been developed towards detection of botnets. Moreover, the flow based botnet detection approaches have only emerged in the recent years.

Wernhuar Tarng, Cheng-Kang Chou and Kuo-Liang Ou performed detection of P2P botnet viruses in the infection stage and report to network managers to avoid further infection. The system adopted real-time flow identification techniques to detect traffic flows produced by P2P application programs and botnet viruses. The experimental results showed that the accuracy of Bayes Classifier was 95.78% and that of NN Classifier was 98.71% in detecting P2P botnet viruses and suspected flows to achieve the goal of infection control [2].

In [3] the authors Pratik Narang, Chittaranjan Hota1 and VN Venkatakrishnan combined the benefits of flow-based and conversation-based approaches with two-tier architecture, and addressed the limitations of the two approaches. The statistical features were extracted from the network traces of P2P applications and botnets. The supervised machine learning model was built to differentiate between benign P2P applications and P2P botnets that could also detect unknown P2P botnet traffic with high accuracy. The three classifiers such as Decision trees, Random forests, and Bayesian network were used to consistently detect P2P botnets with a recall ranging between 88% to 95% and achieved a low false positive rate of 0.2% to 0.3%.

Liu Bin, Lin Chuang, Ruan Donghua, and Peng Xuehai put forward a flow analysis and monitor system based on net flow. The model was built on Brower-Server framework aimed at enterprise scene. Data collection module received and analyzed net flow-exported packets and inserted per flow traffic information into Oracle database. Display module acted

as a J2EE web server and fetched history traffic information from database. A real-time abnormal flow monitor module is embedded into the model to detect worm and other malicious attacks. [4]

The authors, Pijush Barthakur, Manoj Dahal and Mrinal Kanti Ghose presented a comparative analysis of machine-learning based classification of botnet command & control(C&C) traffic for proactive detection of Peer-to-Peer (P2P) botnets. In this paper three models like Decision Tree (C4.5), Bayesian Network and Linear Support Vector Machines were used to detect botnet and their performance results were compared. The proposed algorithm produces better accuracy than the original decision tree classifier. [5]

R.Kannana and V.Ramani developed a system for botnet detection to identify a botnet activity in a network, based on traffic behavior analysis and flow intervals. The approach was to classify packets based on source IP, destination IP, number of packet, etc., using decision tree classification technique in machine learning. The attribute selection was mainly based on packet attribute and does not consider the data part. The feasibility of the work was to detect botnet activity without having seen a complete network flow by classifying behavior based on time intervals [6].

The authors, Rajesh Kumar and TajinderKaur generated an automated system that contained packet capturing, processing of multiple attack logs, labeling of network traffic based on low level features an and delivered a traffic classifier which have classified the normal and malicious traffic. The attack data was collected through honey pot system and normal user browser. The classification algorithm was applied and the model has been built. [7].

In most of the existing work, internet traffic categorization was carried out with low level features to classify the botnet traffic. Also conversation approach were employed which has unavailable features. Hence in the proposed work the flows affected by botnet are collected from different network and extractions of statistical features have been considered for botnet categorization.

## II. PROPOSED WORK

The main aim of this paper is to predict the botnet traffic in the network using machine learning techniques. The problem of botnet traffic detection is modeled as binary classification task and solved using powerful supervised pattern learning algorithms. Machine learning techniques are effective than statistical approach since machine learning techniques automatically learns training data by taking intelligent hints from the data and predicts accurately. The machine learning techniques are used to improve the efficiency of the model and to predict botnet traffic accurately. In this work the CTU 13 dataset is used and the discriminative features are identified to build the classifiers. The classification models are built using Support Vector Machine (SVM), neural network, naïve bayes and decision tree. This botnet identification model can be employed in network administrators and communication service provider helps to protect their networks and subscribers from advanced malware and botnet threats.

The proposed framework of this model consists of four phases: data collection, feature extraction, training and testing and classification. Each phase is described in following sections and the architecture of the proposed model is shown in Fig.1

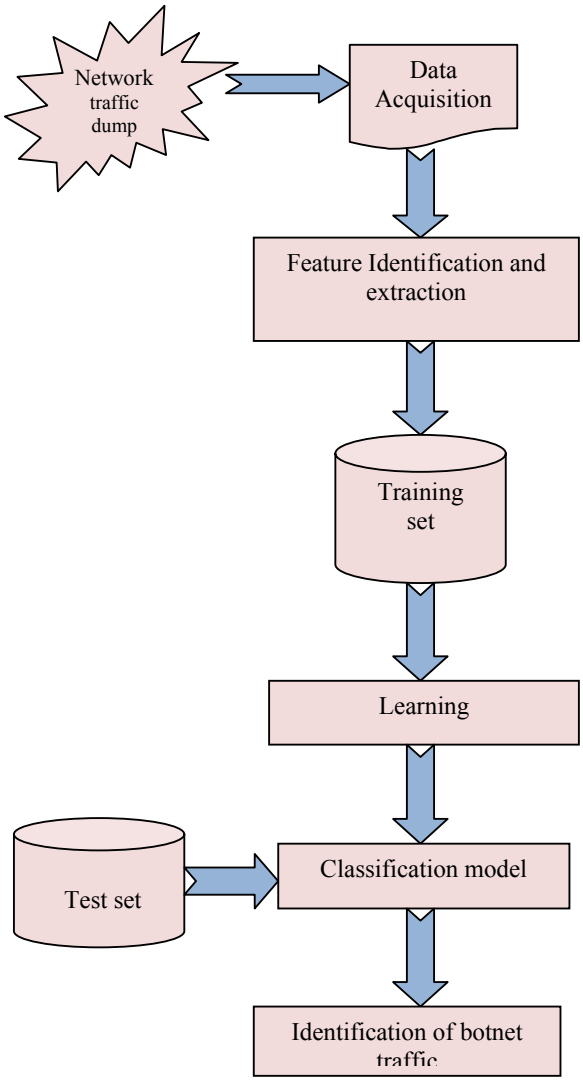


Fig.1 Block diagram for botnet traffic classification

### A. DATASET

Botnet identification problem is modeled as classification task and to build the proposed model, CTU 13 dataset[1] has been used and obtained from malware capture facility project. The dataset consists of 10,48,576 number of network flows including both botnet and normal flows. 500 botnet flows and

[1] http://mcfp.weebly.com/ctu-malware-capture-botnet-42.html

500 normal flows have been collected and the statistical features are extracted to prepare the training dataset. The original dataset consists of attributes like start time, duration, protocol, source and destination address, state, source and destination port, source and destination type of service, total packets, total bytes. These attributes are deficient to identify the botnet traffic. So the features describing the properties of botnet and normal traffic are extracted to attain the high classification accuracy.

## B. FEATURE EXTRACTION

Feature extraction plays vital role in classifier building. Distinctive features that assist to predict the botnet traffic flow are extracted from CTU database. The existing dataset consists of basic features such as start time and duration of particular flow, source and destination port used in specifying the services offered on local or remote hosts, source and destination ip address used for packet to traverse in the network, protocol that specifies interaction between communicating entities, ToS field used in assigning priority for IP packet and total bytes and total packets transmitted during particular flow. These features are not sufficient to discriminate botnet traffic from normal traffic. So features like average byte rate, average packet rate, ping bytes, time comparison, malicious ports which will help in determining the botnet traffic are identified and extracted.

*Start time*

This feature defines the starting time of the flow. It is one of the basic features of net flow fields. Start time is converted as numerical value for classification. The formula for changing time format to numerical value is

Hour*3600+minute*60+second

It is compared with the end time to indicate the flow is malicious.

*End time*

The end time of the flow depicts the flow completion time. The end time is compared with the start time for malicious flow indication. It is calculated using the formula

$$T_e = T_s\text{-Dur}$$

where $T_s$ depicts the flow start time, $T_e$ is the flow end time and Dur represents duration of the flow.

*Duration*

The duration of the flow indicates the total time taken to complete the particular flow. The duration of the flow is used to calculate average packet rate and average byte rate.

*Protocol*

A protocol is the special set of rules that end points in a telecommunication connection use when they communicate. Protocols specify interactions between the communicating entities. There are different types of protocols used they are TCP, UDP, ICMP, SMTP and etc.

*Source IP Address*

The IP Address is used to uniquely identify the desired host to contact. It is also one of the basic features of net flow fields. The source IP address is the IP address of the computer and or website that are currently visiting, or using. The source IP address is converted to decimal format for further processing. It is computed as follows

10.0.2.112 is converted to 167772784

*Destination IP Address*

A destination IP address is the IP address to which a message is sent. IP addresses are used to deliver packets of data across a network and have what is termed end-to-end significance. This means that the source and destination IP address remains constant as the packet traverses a network. Destination IP address is the receiver of information .It is computed as follows

8.8.8.8 is converted to 134744072.

*Source and destination port*

The port number, allows us to identify the service or application our data or request must be sent to, and have previously stated. They can be used to gain information on remote systems that have been targeted for attacks. Port number 80, 53, 25 are marked as malicious flows with different botnet attacks they are http established botnet, spam botnet and DNS server based botnet.

*Direction*

Direction specifies whether data travels in both direction or in just one direction. Direction also specifies the path that flow takes as it travels from source to destination through an internetwork. Most flows are bidirectional and can be represented by double sided arrow and unidirectional flow is represented using single sided arrow. Most of spam botnets use unidirectional flow.

*States*

There are different types of states that represent the network flow they are SYN, RST, CON, ACK, FIN. In the SYN state client sends a SYN message which contains the server's port and the client's Initial Sequence Number to the server. The server sends back its own SYN and ACK. The Client sends an ACK. Final state is the state is a now a half-closed connection. The client no longer sends data, but is still able to receive data from the server. Upon receiving this FIN, the server enters a close state. CON is the connection state in when once the connection is established it is in CON state. The RST state is the connection reset state in which the host refuses a connection. Too many SYN state is received means sender is infected. Too many RST state is received means receiver is infected.

*ToS*

ToS is defined as type of service. It is a mechanism for assigning a priority to each IP packet as well as a mechanism

to request specific treatment such as high throughput, high reliability or low latency. Usually ToS field will be 0.

*Total packets*

Total packet feature is defined as number of packets transferred during the particular flow. It stores the number of packets transmitted during the particular period of time or flow.

*Total Bytes*

The Total Bytes property specifies the total number of bytes that the client sent in the basis of the request. It is total byte size is an important metric for network measurement.

*Time comparison*

A time comparison field is compared using flow start time and flow end time. Ts is the flow start time and Te is the flow end time such that $Ts \leq t \leq Te$ that flow is marked as malicious and the value is 1 otherwise 0.

*Number of RST connection*

The number of reset connection of a flow is the host refuses to make a connection. Number of reset connection is calculated by analyzing the repeated RST connection from same IP address. If too many RST connection is received means receiver is infected.

*Average Byte Rate*

Average byte rate is calculated with the help of existing feature like total bytes and duration. This feature is to identify the average bytes transferred within the flow for particular time.

$$\text{Average byte rate} = \frac{\text{Total bytes}}{\text{Duration}}$$

$$\text{for example average byte rate} = \frac{2811}{2.234} = 1258.162$$

*Average Packet rate*

Average packet rate is derived using the existing feature like total packets and duration. This is used to identify the average number of packets transmitted in a flow during particular interval of time.

$$\text{Average packet rate} = \frac{\text{Total packets}}{\text{Duration}}$$

$$\text{for example average packet rate} = \frac{8}{2.234} = 3.581$$

*Ping Bytes*

The correctly formed IP packet including IP header is 65535 bytes, including the payload size 84 bytes. Sending of packets larger than 65535 bytes violates IP; attacker sends malformed packets in fragments. Hence if total bytes are greater than 65535 the flow is marked as malicious and the value is 1 otherwise 0.

A set of 22 features including the above mentioned basic features and discriminative features are extracted from each network flow to form a feature vector. The above features are computed using java code.

## III. MACHINE LEARNING ALGORITHMS

Four machine learning algorithms, Decision tree classifier, Multilayer perceptron, Naïve bayes classifier and Support Vector Machine were used for learning the classification model.

### A. *Multilayer Perceptron*

Multilayer Perceptron (MLP) network is the most widely used neural network classifier. MLP networks are general-purpose, flexible, nonlinear models consisting of a number of units organized into multiple layers. The complexity of the MLP network can be changed by varying the number of layers and the number of units in each layer. Given enough hidden units and enough data, it has been shown that MLPs can approximate virtually any function to any desired accuracy. In other words, MLPs are universal approximators. MLPs are valuable tools in problems when one has little or no knowledge about the form of the relationship between input vectors and their corresponding outputs.

### B. *Decision tree induction*

Decision Tree Classification generates the output as a binary tree like structure called a decision tree, in which each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or decision. A Decision Tree model contains rules to predict the target variable. This algorithm scales well, even where there are varying numbers of training examples and considerable numbers of attributes in large databases.

J48 algorithm is an implementation of the C4.5 decision tree learner. This implementation produces decision tree models. The algorithm uses the greedy technique to induce decision trees for classification. A decision-tree model is built by analyzing training data and the model is used to classify unseen data. J48 generates decision trees, the nodes of which evaluate the existence or significance of individual features.

### C. *Naïve Bayes Classification*

The Naive Bayes Classifier (NB) is a simple but effective classifier which has been used in numerous applications of information processing including, natural language processing, information retrieval, etc. The Naive Bayes Classifier technique is based on Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Naïve Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of other variable. The Naive-Bayes inducer computes conditional probabilities

of the classes given the instance and picks the class with the highest posterior. Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting.

D. *Support vector machine*

Support Vector Machine a new approach to supervised pattern classification which has been successfully applied to a wide range of pattern recognition problems. Support vector machine is a training algorithm for learning classification and regression rules from data. SVM is most suitable for working and efficiently with high dimensionality feature spaces. SVM is based on strong mathematical foundations and results in simple yet very powerful algorithms. [22-24]

The standard SVM algorithm builds a binary classifier. A simple way to build a binary classifier is to construct a hyper plane separating class members from non-members in the input space. SVM also finds a nonlinear decision function in the input space by mapping the data into a higher dimensional feature space and separating it there by means of a maximum margin hype rplane. The system automatically identifies a subset of informative points called support vectors and uses them to represent the separating hyper plane which is sparsely a linear combination of these points. Finally SVM solves a simple convex optimization problem.

The machine is presented with a set of training examples, (xi,yi) where the xi are the real world data instances and the yi are the labels indicating which class the instance belongs to. For the two class pattern recognition problem, yi = +1 or yi = -1. A training example (xi,yi) is called positive if yi = +1 and negative otherwise. SVMs construct a hyper plane that separates two classes and tries to achieve maximum separation between the classes. Separating the classes with a large margin minimizes a bound on the expected generalization error.

The simplest model of SVM called Maximal Margin Classifier, constructs a linear separator given by w T x - $\gamma = 0$ between two classes of examples. The free parameters are a vector of weights **w** which is orthogonal to the hyper plane and a threshold value $\gamma$. These parameters are obtained by solving the following optimization problem using Lagrangian duality

$$L(\mathbf{w},b,u)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{l}u_i\left[D_{ii}(\mathbf{w}^T\mathbf{x}_i-\gamma)-1\right] \quad (1)$$

.where Dii corresponds to class labels, which assumes value +1 and –1. The instances with non null weights are called support vectors.

When the number of classes is more than two, then the problem is called multiclass SVM. There are two types of approaches for multiclass SVM. In the first method called indirect method, several binary SVM's are constructed and the classifier's output are combined for finding the final class. In the second method called direct method, a single optimization formulation is considered.

## IV.EXPERIMENTS AND RESULTS

The botnet classification model is generated by implementing supervised machine learning algorithms. The experiments have been performed using R and classification algorithms like Naïve Bayes, Decision Tree, Support Vector Machine and Neural Networks. Features such as average byte rate, average packet rate, average packet length, number of reset connection, time comparison, end time, ping bytes, malicious ports are extracted based on the characteristics of flow data and the training dataset with 1000 instances is developed as described in section 2.1. For each feature vector, the class labels M or N are assigned. Label M indicates botnet traffic and label N indicates normal traffic.

The performance of the classifiers is evaluated and comparative analysis has been carried out. Classification accuracy is used as a primary performance measure for evaluating the classifiers and is measured as the ratio of the number of correctly classified instances in the test dataset and the total number of test cases. The performances of the trained models are evaluated based on the criteria of precision, recall, f-measure and accuracy using 10 fold cross validation.

The formula for calculating accuracy is,

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

The formula for calculating recall is,

$$\text{Recall} = \frac{TP}{TP + FN}$$

The formula for calculating precision is,

$$\text{Precision} = \frac{TP}{TP + FP}$$

the formula for f-measure is,

$$F = 2*(\text{Precision}*\text{Recall}) / (\text{Precision} + \text{Recall})$$

Where TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

The results of four models in terms of prediction accuracy, Precision, recall and f-measure are shown in Table 1 and the comparative performance of classifiers is shown in Fig. 2.

TABLE I COMPARITIVE RESULTS OF THE CLASSIFIER

| Evaluation criteria | Classifiers | | | |
|---|---|---|---|---|
| | DT | NB | NN | SVM |
| Precision | 0.973 | 0.984 | 0.336 | 1.000 |
| Recall | 0.929 | 0.991 | 0.533 | 0.998 |
| F-measure | 0.95 | 0.987 | 0.412 | 0.998 |
| Prediction Accuracy | 95.7% | 98.4% | 54.5% | 99.8% |

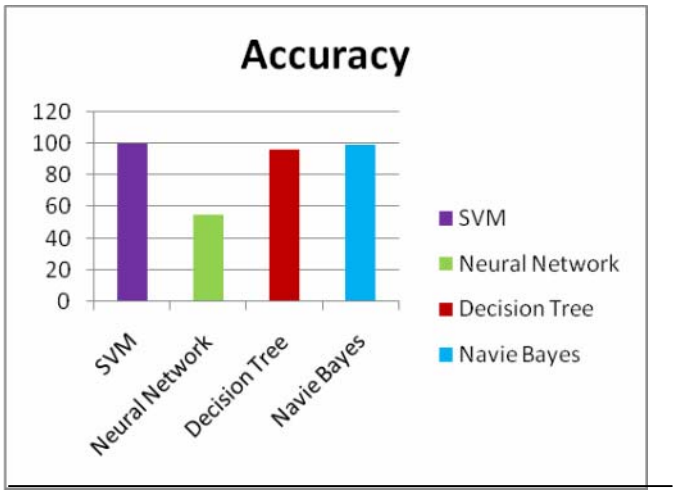


Fig.2 Classification accuracy of the models

## CONCLUSION

This research work describes the modeling of the botnet traffic prediction task as classification and its implementation using supervised learning techniques and also indicates the execution of suitable machine learning algorithm in the field of network security. The CTU 13 dataset has been obtained from malware capture facility project and the feature extraction was done. The trained models have been generated using Bayes classifier, NN classifier, Decision tree classifier and SVM classifier. The performance of the classifiers are evaluated using 10 fold cross validation and observed that SVM based botnet categorization model outperforms.

From the results, it is suggested that this approach of employing machine learning technique to classify botnet traffic will be appropriate to provide a highly secure environment**.**

## REFERENCES

[1] P.S.Lokhande, B.B.Meshram, " Botnet: Understanding Behavior, Life Cycle Events & Actions", International Journal of Advanced Research in Computer Science and Software Engineering ,vol.4, March 2014.

[2] Wernhuar Tarng, Cheng-Kang Chou and Kuo-Liang Ou, " A P2P Botnet Virus Detection System Based on Data-Mining Algorithms", International Journal of Computer Science & Information Technology Vol 4, No 5, October 2012

[3] Pratik Narang, Chittaranjan Hota1 and VN Venkatakrishnan, "PeerShark: flow-clustering and conversation-generation for malicious peer-to-peer traffic identification", EURASIP Journal on Information Security 2014, **2014**:15

[4] Liu Bin, Lin Chuang, Ruan Donghua, Peng Xuehai (2003), "*NetFlow Based Flow Analysis and Monitor",* National Natural Science Foundation of China and Microsoft Research.

[5] Pijush Barthakur, Manoj Dahal and Mrinal Kanti Ghose, " An Efficient Machine Learning Based Classification Scheme for Detecting Distributed Command & Control Traffic of P2P Botnets", International Journal of Modern Education and Computer Science,November 2013.

[6] R.Kannan, A.V.Ramani, "Flow based analysis to identify botnet infected systems", Journal of Theoretical and Applied Information Technology , vol.67, No.2, September 2014

[7] Rajesh Kumar, TajinderKaur, "Machine learning based Traffic Classification using Low Level Features and statistical Analysis", International Journal of Computer Applications (0975 – 8887), vol 108, No 12, December 2014

[8] Ahmed Abdalla, Haitham A. Jamil, Hamza Awad, Hamza Ibrahim, Sulaiman, Mohd Nor, "Malware Detection using IP Flow Level Attributes", Journal of Theoretical and Applied Information Technology, Vol. 57 No.3, November 2013.

[9] Pratik Narang, Chittaranjan Hota1 and VN Venkatakrishnan, "Peershark: Detecting Peer-to-Peer Botnets by Tracking Conversations, IEEE Security and Privacy Workshops,2014

[10] Santhana lakshmi V, Vijaya MS, "Efficient prediction of phising websites using supervised learning algorithms", International conference on Communication Technology and system design, 2011.

[11] Gracia, S. (2013). Malware Capture Facility Project. CVUT University. Retrieved Februrary 03, 2013, from https://agents.fel.cvut.cz/malware-capture-facility .

[12] Faily, Maryam, Shahrestani, Alireza and Ramadass, Sureswaran, "A Survey of Botnet and Botnet Detection", In Third International Conference on Emerging Security Information, Systems and Technologies, 2009

[13] Carl Livadas, Robert Walsh, David Lapsley, W. Timothy Strayer (2006), "Using Machine learning techniques to Identify Botnet Traffic", second IEEE LCN workshop on network security.

[14] Pijush Barthakur, Manoj Dahal, Mrinal Kanti Ghose, "An Efficient machine Learning based Classification Scheme for Detecting Distributed Command & Control Traffic of P2P botnets", International journal of Modern Education and Computer Science, 2013

[15] W.Timothy Strayer, David Lapsely, Robert Walsh, Carl Lividas, "Botnet Detection Based on Network Behaviour", In Advances in Information Security, Springer, 2008

[16] David Zhao, Issa Traore, Bassam Sayed, Wei Lu, SherifSaad, Ali Ghorbani, Dan Garant, "Botnet detection based on traffic behaviour analysis and flow intervals", Computer and Security, 2013.

[17] Bhavani Thuraisingham, Latifur Khan, Mohammad M. Masud, Kevin W. Hamlen, "Datamining for Security Application", IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2008.

[18] P Narang, JM Reddy, C Hota, "Feature selection for detection of peer-to-peer botnet traffic", Proceedings of 6th ACM India Computing Convention, 2013

[19] Haritha S Nair, Vinoth Ewards S E, "A Study on Botnet Detection Techniques", International Journal of Scientific and Research Publications, Vol. 2, issue 4, April 2012.

[20] David Anselmi, Jimmy Kuo, Richard Boscovich, "Battling Botnets for ontrol of computres" in Microsoft/Security Intelligence Report.

[21] Pratik Narang, Chittaranjan Hota1 and VN Venkatakrishnan, "PeerShark: Deteting Peer – to – Peer Botnets by tracking conversations,IEEE security and Privacy Workshops,2014.

[22] Khan , L. and Masud, M. et al. "Flow-based identification of botnet traffic by mining multiple log files", Distributed Framework and Applications, First International Conference on. IEEE, 200-206,2008

[23] John Shawe-Taylor, Nello Cristianini, "Support Vector Machines and other kernel-based learning methods", 2000, Cambridge University Press, UK.

[24] Vapnik V.N,"Statistical Learning Theory", J.Wiley & Sons, Inc.,1998, New York.

[25] Soman K.P, Loganathan R, Ajay V, " Machine Learning with SVM and other Kernel Methods", 2009, PHI, India.