Journal of Theoretical and Applied Information Technology
30th April 2017. Vol.95. No 8
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

# HYBRID BOTNET DETECTION USING ENSEMBLE APPROACH

[1]SAMSON F, [2]VAIDEHI V
[1]MCA Student, Department of Computer Science, Christ University, Bengaluru
[2]Associate Professor, Department of Computer Science, Christ University, Bengaluru
E-mail: [1]samsonhacks@gmail.com, [2]vaidhehi.v@christuniversity.in

## ABSTRACT

Botnets are one of the most threatening cyber-attacks available today. This paper proposes a hybrid system which can effectively detect the presence of C&C, P2P and hybrid botnets in the network. The powerful machine learning algorithms like BayesNet, IBk, KStar, J48 and Random Tree have been deployed for detecting these malwares. The performance and accuracy of the individual classifiers are compared with the ensemble approach. Labelled dataset of botnet logs were collected from the Malware Facility. Secured data was collected from Christ university network and the combined dataset is tested using virtual test bed. The performance of the algorithms is studied in this paper. Ensemble approach out performed individual classifiers.

**Keywords:** *Botnet, C&C, P2P, Hybrid Botnets, Ensemble*

## 1. INTRODUCTION

The total number of devices connected over the internet is constantly on the increase each day. Parallel to the increase of users the security attacks are also on the increasing scale. One of the prominent attacks that contribute to the overall security attacks is botnet, yet the internet users have least knowledge about this attack. Generally botnet can be defined as a network of compromised hosts controlled by an individual or group of personals known as botmaster. These compromised hosts are also called as zombies or bots. Botnets grow in size by recruiting vulnerable machines through spam emails, malicious websites, spam applications, port level exploitation, and drive-by-download. The bots are controlled by the botmaster by interacting with them through communication channels such as Internet Relay Chat (IRC). Botmasters use techniques like HTTP tunneling, IPV6 tunneling to hide their identity [15] while communicating with the bots. Once the channel is setup for communication, the bots are used for launching attacks like Distributed Denial of Service (DDoS), sending spam emails, accessing malicious phishing sites for the purpose of click-fraud, stealing confidential information like bank pin numbers, Credit card number, social security passwords, etc., manipulating the vulnerabilities of the hosts to install backdoors for other types of attacks and many unethical activities [10].

Initially the architecture that was used for the botnet implementation was based on centralized command and control (C&C) server as shown in Figure 1(a). This design was more prone to detection and take down as there is one central point for all the bots. Therefore this design has gradually evolved and produced the design of peer-to-peer (P2P) shown in Figure 1(b).

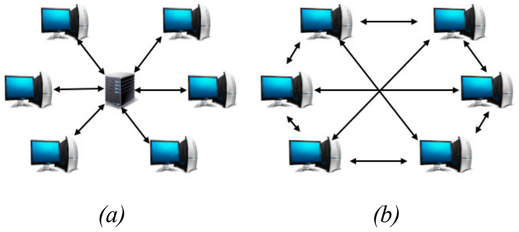


*Figure 1: (a) Centralized Architecture and (b) P2P Architecture*

P2P botnet eliminates the need for the centralized servers to make the system less prone to detection and take down. The next generation of botnets use a combination of both [14] the centralized and decentralized architectures where for instance at the top level the C&C topology is used receive commands from the master and a P2P topology is used to communicate between the remaining peers.

For an effective detection it is very important to have a good understanding about the botnet life cycle. All the botnet activities can be classified under three aspects: botnet behavior, detection mechanisms, and defense strategies. The behavior focuses on the initial life cycle events of botnets like host-infection, rallying and C&C communication. This behavior varies according to the architectures that are used for implementation. The detection is very vital among all the three because only by identifying malicious behavior the botnets can be studied closely and consequently effective measures can be taken to prevent future attacks. All these botnet related activities are very stealthy in nature and follow a kind of algorithm which can surpass the security scanning done by the firewall and many of the antivirus softwares. Therefore the detection of these malicious binaries becomes a challenging task.

The machine learning approach has been proposed by many researchers as best feasible solution for the detection of the botnets in recent times. Among the many available machine learning algorithms BayesNet, J48, Naive Bayes, Decision Tree and Support Vector Machine are the most prominently used algorithms [1,2,3,5]. Among these algorithms the botnet malwares detection accuracy ranges from 95% – 99.9%. But given a single algorithm among these does not promise a 100% for all the available architectures and diverse situations. That is if one single algorithm is considered for all situations the results may not always be reliable as the predictions heavily depend upon the dataset that is being used for training the model. Hence the best performing algorithms have been identified and combined to get better results under varied circumstances. Using this approach the false-positive and false-negative rates are brought down significantly.

In real time for instance in case of web server even uptime 99.9 would mean a lot of business loss. Therefore a near perfection as suggested by 6-sigma levels would be a feasible condition. This paper approaches this problem first by suggesting an ensemble algorithms that achieve the expected accuracy of detection and second by generalizing the detection to all the available botnet architectures by considering the diverse behavior of the malwares and hybrid nature of the network. The rest of the paper is organized as follows. Section 2 reviews the previous work done on botent detection. Section 3 elaborates the exclusive methodology used for botnet detection. Section 4 explains about the test bed creation. Section 5 briefs the performance of the algorithms. Section 6 concludes by summarizing the results and suggesting future enhancements.

## 2. LITERATURE REVIEW

Jignesh Vania et.al [4] reviews on all the areas of botnets starting from the types of architectures used for construction. They also provide overview of the attacks performed using botnets and the three detection approaches ie, centralized, decentralized and hybrid. All the approaches that are used for detection of botnets can be broadly classified as i)Signature-based approach which maintains a list of botnet signatures for detection, ii)Anomaly-based approach which monitors the abnormal behavior and anomalies in the network for detection, iii)DNS-based approach which identifies the suspicious DNS network traffic for detection, iv)Mining-based approach takes the available log files as input and correlates the data and observes the trends to identify the malicious botnet and v)Honeypot botnet detection that has a dedicated detection environment that exposes itself for the botnet attack and analyzes the behavior, the size and structure of the botnet. Among the above mentioned detection methods the Honeypot falls in the active detection category as it partakes in the botnet attack by projecting itself vulnerable to the attack and the rest falls in the passive detection category as they perform analysis on the observed data. The Honeypot (also called Honeynet) is time-consuming and static in nature [3]. It is very expensive to set up and needs a protected environment to implement. Due to these limitations the current trend pre-dominantly focuses on the passive detection approaches.

The Data-mining approach comprises the most of the features of other passive detection approaches [3]. This approach has the advantage of easy implementation on desired host machines irrespective of the operating system types and incurs less expenses and resources. Carl Livada et.al [2] analyses the IRC botnet traffic to detect the C&C server. Their work consists of two stages. In the first stage the IRC traffic is separated from the other network traffic and in the second stage the botnet IRC traffic is separated from the benign IRC traffic. For stage one Naïve Bayes classification produced best results and for the subsequent stage J48 and Bayes network classifier produced best results. Shree Garg et.al [3] enhances the detection methodology proposed by Livadas et.al [2] by analyzing the P2P network traffic. Their work compares the performances of J48, Naïve Bayes and IBk. Their finding is that J48 and IBk perform better than Naïve Bayes but J48 has the limitation of high training time and IBk has the limitation of high testing time. *Their future work focuses on improving the testing and training times.*

For each algorithm the accuracy and efficiency of detection of botnet varies according to the dataset used for building the model and the number and types attributes used. Raman Singh et.al [13] analyses the general network traffic to find the set of features that give high accuracy with less detection time and less space complexity. This comparative study suggests filtered subset evaluation as the best technique for feature selection. Biglar et.al [11] studied the feature sets that were used in classification algorithms. After evaluating all the flow level features of 16 major botnet traces proposed a final feature set with 99% of detection rate. Matija et.al [12] monitored the network flows for a limited time duration and logged small amount of network traffic for each flow. This data is then analyzed under 8 machine learning algorithms and finally proposes Random Tree classifier as the best detection algorithm. Ritu et.al [1] use only a fraction of the complete dataset to developing the model and hence they do not always assure a 100% accuracy of detection. The attributes considered in [1] are source IP, destination IP, source port, destination port, protocol, total packets, total bytes and duration.

Mohammad et.al [5] correlates tcpdump and exedump. The tcpdump has packet number, arrival time, depature time, source IP, destination IP and payloadInfo as its attributes. The exedump has process start time and process name as its attributes. Their model gives better performance compared to Livadas et.al [2] using the classification algorithms: support vector machine, Bayes Net, Boosted decision trees, Naïve Bayes and decision tree. *However the limitations of their approach is that they make assumptions about the response time between the C&C server and the hosts which is a drawback of this system.* David et.al [6] use the anomaly based detection for finding the P2P malicious traffic. A total of 12 attributes are filtered from the captured data and classified using Bayes network classifier and decision tree. *They look forward to produce a system which can detect a zero day attack.* Hossein et.al [8] overcome this drawback by identifying hosts which perform atleast one malicious activity. Using the data captured from these malicious hosts they group the hosts which show similar behaviors. This approach does not requires any prior knowledge of the malwares and can be used for effective detection of zero day attacks.

Sherif Saad et.al [7] after classifying the captured dateset under machine algorithms conclude that online detection methodology would give a realistic approach to detect the botnets. And if this online system is implemented it would remove the need for further analysis of the performance of ML algorithms. Farhood et.al [10] detect botnet based on protocols and patterns used for communication between hosts. *The detection methods suggested in this paper are limited to only the centralized botnets and does not exclusively talk about the P2P related botnet detection.* Shehar Bano et.al [14] used 200 GB of data collected at B-RAS ISP corresponding to 511 homes and found that 58(11.3%) of them as compromised botnets. They monitored the bot life cycle events till the attack stage. Moheeb et.al focused on the lack of complete knowledge about the botnets. Hence they combine independent sources to construct a complete knowledge of the botnet. They elaborated on the botnet structure, size, lifetime and growth patterns by observing 192 unique IRC botnets.

From the literature it is evident that lots of research has been done on finding out the best suited classifier using machine learning approach. However, the research related to combine these algorithms using ensemble approach is minimum. Though the ensemble algorithm approach consumes relatively more classification time but due its high accuracy of classification it is used for implementation of Botnet Detection System. The ensemble set containing BayesNet, IBk and J48 and the second set containing BayesNet, IBk and Random Tree give a 100% accuracy.

## 3. METHODOLOGY

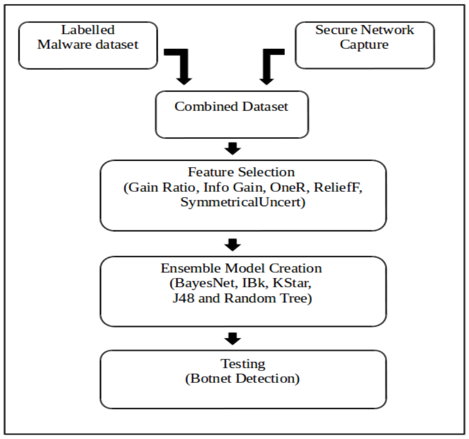

Labelled Malware dataset
Secure Network Capture
Combined Dataset
Feature Selection (Gain Ratio, Info Gain, OneR, ReliefF, SymmetricalUncert)
Ensemble Model Creation (BayesNet, IBk, KStar, J48 and Random Tree)
Testing (Botnet Detection)

*Figure 2: Flow Of Botnet Detection System*

### 3.1. Data Collection

The strength of any detection approach lies in the set of features of the dataset used for building the model. Most of the previous works have used a custom built dataset that lacks some important features. These custom built datasets have different properties and cannot be considered for the generalized detection system [9]. This research is an effort to overcome above mentioned drawbacks and aims to detect both the centralized C&C botnets and the distributed P2P botnets given any valid network dump. All these requirements are fulfilled by the dataset that was captured by Malware Capture Facility Project, CTU-13 in the CTU University, Czech Republic in 2011. This dataset has labeled malware capture for six botnets namely donbot, fastflux, neris, qvod, rbot and sogou. Garcia et.al [9] used this dataset for the first time. To constitute a complete dataset the network logs are dynamically captured from the secured network Christ University [16]. These logs are then appended to the botnet logs to produce a complete dataset that can be used for building the training model. A random collection of 1000 packets from each botnet logs is selected and is appended to a collection of 6000 secure packets. Therefore out of a total of 12000 instances 50% are botnet instances and the remaining are genuine network instances.

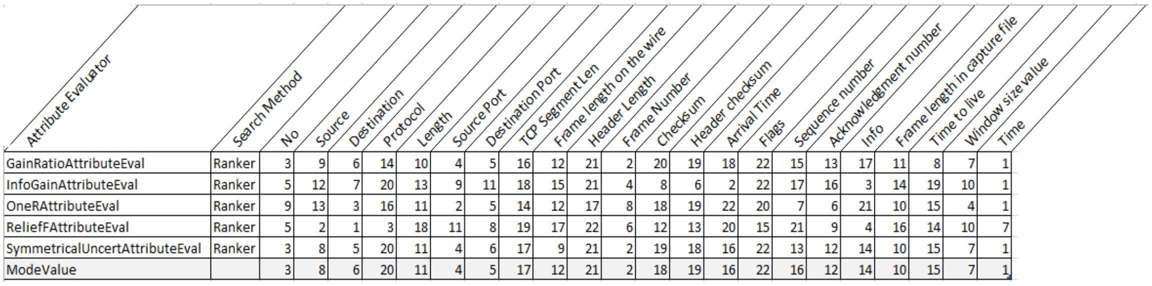
### 3.2. Feature Identification And Selection

As per the observations of [15] about 80% of the internet traffic is based on TCP. So the TCP packet related attributes are given prominence over the other attributes. The main advantage of considering the TCP packets is that the number of null valued attributes are closed to none. This improves the accuracy of the detection system. Table 1 lists all the features initially considered for detection.

*Table 1: All The Features Initially Considered For Botnet Detection*

| S. No. | Feature | Description |
|---|---|---|
| 1 | No | The serial number of packet capture |
| 1 2 | Source IP | Sender's IP address |
| 2 3 | Destination IP | Receiver's IP address |
| 3 4 | Protocol | Protocol used at transport layer |
| 4 5 | Length | Length of the packet |
| 5 6 | Source Port | Port number used on the source host |
| 6 7 | Destination Port | Port number used on the destination host |
| 7 8 | TCP Segment Len | The number of bytes received in the current TCP segment |
| 8 9 | Frame Length | The number of bytes of data in the frame's payload |
| 9 10 | Header Length | Four bit field that specifies the length of the header |
| 10 11 | Frame Number | The sequence number of the current frame |
| 11 12 | Checksum | A redundancy check that is used to detect errors in data |
| 12 13 | Header Checksum | A checksum used in IP4 to detect errors and data corruption |
| 13 14 | Arrival Time | The time of arrival at the destination |
| 14 15 | Flags | Predefined bit sequence that holds a binary value |
| 15 16 | Sequence Number | Sequence number of the packet |
| 16 17 | Acknowledgement Number | The acknowledgement number sent by the receiver |
| 17 18 | Info | Information about the data being transmitted |
| 18 19 | Frame Length Stored on the captured File | The length stored in the .pcap file |
| 19 20 | Time to Live | The number of hops that a packet is permitted to travel |
| 20 21 | Window Size Value | The amount(bytes) of data that can be buffered at the receiving end |
| 21 22 | Time | Time stamp of packet transfer |

*Table 2: The List Of Ranks For Each Attributes*



| Attribute Evaluator | Search Method | No | Source | Destination | Protocol | Length | Source Port | Destination Port | TCP Segment Len | Frame length on the wire | Header Length | Frame Number | Checksum | Header checksum | Arrival Time | Flags | Sequence number | Acknowledgement number | Info | Frame length in capture file | Time to live | Window size value | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GainRatioAttributeEval | Ranker | 3 | 9 | 6 | 14 | 10 | 4 | 5 | 16 | 12 | 21 | 2 | 20 | 19 | 18 | 22 | 15 | 13 | 17 | 11 | 8 | 7 | 1 |
| InfoGainAttributeEval | Ranker | 5 | 12 | 7 | 20 | 13 | 9 | 11 | 18 | 15 | 21 | 4 | 8 | 6 | 2 | 22 | 17 | 16 | 3 | 14 | 19 | 10 | 1 |
| OneRAttributeEval | Ranker | 9 | 13 | 3 | 16 | 11 | 2 | 5 | 14 | 12 | 17 | 8 | 18 | 19 | 22 | 20 | 7 | 6 | 21 | 10 | 15 | 4 | 1 |
| ReliefFAttributeEval | Ranker | 5 | 2 | 1 | 3 | 18 | 11 | 8 | 19 | 17 | 22 | 6 | 12 | 13 | 20 | 15 | 21 | 9 | 4 | 16 | 14 | 10 | 7 |
| SymmetricalUncertAttributeEval | Ranker | 3 | 8 | 5 | 20 | 11 | 4 | 6 | 17 | 9 | 21 | 2 | 19 | 18 | 16 | 22 | 13 | 12 | 14 | 10 | 15 | 7 | 1 |
| ModeValue | | 3 | 8 | 6 | 20 | 11 | 4 | 5 | 17 | 12 | 21 | 2 | 18 | 19 | 16 | 22 | 16 | 12 | 14 | 10 | 15 | 7 | 1 |

Maximum accuracy with minimum number of attributes is required to minimize the computation time of the detection system [16]. Various ranking algorithms have been employed for the selection of the desired attributes. Table 2 lists the ranks and their model value with respect to the ranking algorithms used.

After ranking the attributes using all feasible algorithms the best ones are selected by using majority voting. During the execution of various feature selection algorithms it is found that top 13 features had better importance value than others. Therefore the top 13 attributes based on their importance value are considered in this work. The selected attributes are listed in Table 3.

*Table 3: The Selected Attributes*

| S. No. | Feature | Description |
|---|---|---|
| 1 | Source IP | Sender's IP address |
| 2 | Destination IP | Receiver's IP address |
| 3 | Length | Length of the packet |
| 4 | Source Port | Port number used on the source host |
| 5 | Destination Port | Port number used on the destination host |
| 6 | Frame Length | The number of bytes of data in the frame's payload |
| 7 | Frame Number | The sequence number of the current frame |
| 8 | Acknowledgement Number | The acknowledgement number sent by the receiver |
| 9 | Info | Information about the data being transmitted |
| 10 | Frame Length Stored on the captured file | The length stored in the .pcap file |
| 11 | Time to Live | The number of hops that a packet is permitted to travel |
| 12 | Window Size Value | The amount(bytes) of data that can be buffered at the receiving end |
| 13 | Time | Timestamp of packet transfer |

### 3.3. Model Creation

For effective botnet detection models are created using BayesNet, IBk, KStar, J48 and Random Tree. The classifiers are trained using 10 fold cross validation method. The performance of various classifier models is discussed in detail in section 4.

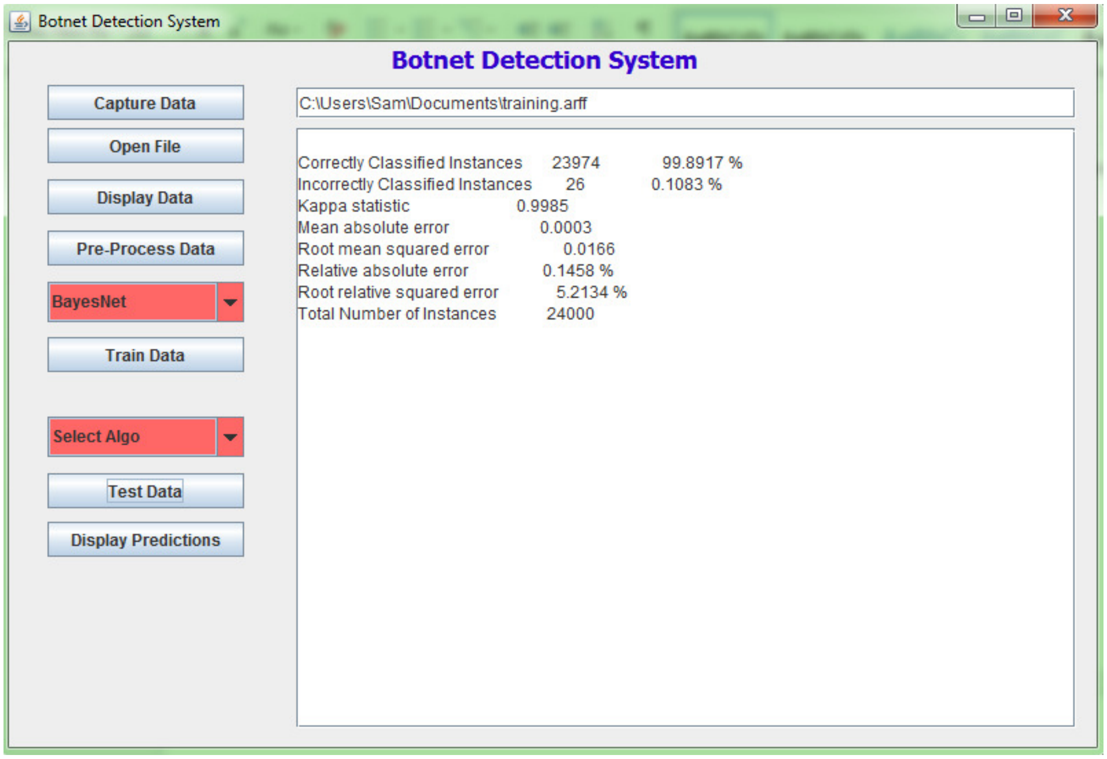
### 3.4. Implementation

All the steps involved in detection of botnets starting from capturing the network logs through detection of the botnet traffic is easily available under one integrated tool 'Botnet Detection System'(BDS). Figure 3 shows an instance of BDS. This integrated tool is built completely using Java and therefore it can support multiple platforms and very easy to implement on any operating system. The live data is captured from the currently connected network using tshark, an API of

Wireshark. Through this the dependency over an abstract dataset is reduced by 50%. The captured normal data is preprocessed by associating each instance with a normal class label and merged with other botnet data to get the complete dataset. This dataset is imported into BSD tool to perform testing and training. All the algorithms related implementations are done using the weka API. The object oriented paradigm used in weka boost the performance and execution time of the tool, paving way for efficient detection of malwares even in high band-width network.

## 4. TEST BED CREATION

The trained model is tested in the virtual system by using test bed. To capture the real world network traffic a virtual network was modeled using the Oracle Virtual box. This test network consisted of 10 host machines following the p2p and centralized network architectures. A secure network environment was created using the internal network facility provided by the virtualization software. A dedicated host was used as victim host and it received genuine users' data and botnet data from all the other hosts in the network. Another six hosts were projected as genuine users having genuine commutations with the target system. And the remaining three hosts are considered as bots or zombies and they transmit the botnet related packets in the network along with the benign network traffic. A random collection of 50 packets each were picked from complete dataset pertaining to each bots using weka random sample creation utility. The chosen packets were processed in Wireshark to get a valid hex code dump for each of the packets. The hex code dump was then injected into the network using packet sender from zombie hosts. The TCP packets constituted over 80% of the test data as they are the majority used protocols in the wide area network. The remaining protocols such as UDP constitute the remaining 20% of the network traffic. The network logs are captured at the victim host using Wireshark network capture tool. Figure 4 is a pictorial representation of set up used for test bed creation.



C:\Users\Sam\Documents\training.arff

```
Correctly Classified Instances     23974          99.8917 %
Incorrectly Classified Instances      26           0.1083 %
Kappa statistic                    0.9985
Mean absolute error                   0.0003
Root mean squared error                  0.0166
Relative absolute error             0.1458 %
Root relative squared error            5.2134 %
Total Number of Instances           24000
```

*Figure 3: An Instance Of Botnet Detection System (BDS)*

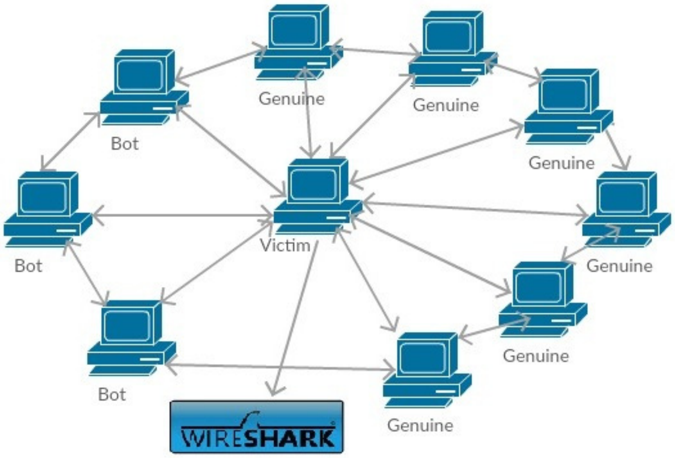


*Figure 4: Network Structure Used For Test Bed Creation*

## 5. RESULTS AND DISCUSSIONS

After analyzing all of the Bayesian classification algorithms the Network classifier is very apt as the dataset constitutes network logs. Bayes Net is capable of finding the inter-dependencies between different attributes [5]. Though the lazy algorithms consume more time to build the model they provide a perfect accuracy and hence choose two of best performing lazy algorithms ie., IBk and KStar. IBk algorithm calculates the centroid for the class and minimum distance is calculated to achieve the classification. From the third category Tree classification algorithms J48 and Random Tree are selected as they considered best among the trees. J48 builds a tree using divide-and-conquer technique [3]. The chosen dataset was initially classified under five chosen algorithms.

The various classifier models were trained by using 10 fold cross validation method. The accuracy and time taken to build the model are considered as performance measures as listed in Table 4.

*Table 4: The Performance Of Various Classifiers*

| S.No. | Algorithm | Efficiency (%) | Time Taken (sec) |
|---|---|---|---|
| 1 | Bayes Network | 99.89 | 0.06 |
| 2 | Ibk | 100 | 8.41 |
| 3 | Kstar | 100 | 612.44 |
| 4 | J48 | 99.95 | 0.01 |
| 5 | Random Tree | 100 | 0.01 |

*Table 5: The Results Of Ensemble Algorithms*

| S.No. | Algorithm1 | Algorithm2 | Algorithm3 | Efficiency (%) | Time Taken (sec) |
|---|---|---|---|---|---|
| 1 | BayesNet | IBk | J48 | 100 | 8.24 |
| 2 | BayesNet | KStar | J48 | 100 | 609.63 |
| 3 | BayesNet | IBk | Random Tree | 100 | 8.34 |
| 4 | BayesNet | KStar | Random Tree | 100 | 610.26 |

But the above mentioned algorithms does not perform the same on all network logs. It differs according to the dataset used for classifying. In order to overcome this drawback ensemble approach is used to get better results and performance at diverse situations as shown in Table 5. Clearly in the above table all of them give a 100% efficiency but the sets 1 and 3 take relatively a less time than the other two sets. Therefore these two sets are chosen for building the model and detection of botnet.

The ensemble algorithms approach presented in this paper improves the accuracy of botnet malware detection to reach a near perfection state that achieves the 6-sigma quality levels. This approach also posted reliable given the different architectures available for botnet implementation. This approach can be implemented off-line as well to inspect the network logs for detecting any possible botnet related activities.

Due to the presence of more than one algorithm for building a single detection model the time consumption for building the model is relatively more than the individual classifiers. A better system can be identified which either uses a high computation power to match up the classification time or a separate algorithm can be proposed that removes the redundant steps among the ensemble algorithms. As this approach also uses a per-defined dataset for build the ensemble model it will fail to detect the zero day botnet attacks.

## 6. CONCLUSION

This paper proposes a tool which works on the basis of ensemble algorithms for detecting the botnet activity. The proposed tool takes into account both the Centralized and distributed architectures of botnet and builds a hybrid model which can effectively classify any network flows as botnet or non-botnet. Initially single algorithm classification is considered and the accuracy of detection ranges from 99.89% to 100%. But due to the inconsistency at adverse situations a boosted ensemble algorithm is used to achieve reliability, high performance and accuracy. To achieve the proposed objectives initially the best performing algorithm from Bayesian, Lazy and Tree algorithms are selected. Among Bayesian algorithms Bayes Network is chosen as it can effectively classify the network data, among Lazy algorithms though both IBk and KStar produce a perfect accuracy but only Ibk is chosen due to its less classification time (8.41 sec) when compared to KStar's execution time (612.44 sec) and among the Tree algorithms both J48 and Random Tree are chosen as they both have high accuracy and consume less time (0.01 sec) to build the model. So out of four chosen algorithms four sets of algorithms are identified each containing a combination of three algorithms ie, 4!/3! = 4. Among these four sets two are chosen for the implementation of Botnet Detection System considering their classification accuracy and time taken to build the model. The primary set containing BayesNet, Ibk and J48 with 100% accuracy and 8.24 seconds of execution time and the secondary set containing BayesNet, Ibk and Random Tree with 100% accuracy and 8.34 seconds of execution time.

## REFERENCES:

[1] Kaushal, Rishabh. "Machine Learning Approach for Botnet Detection."

[2] Livadas, Carl, et al. "Usilng machine learning technliques to identify botnet traffic." *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*. IEEE, 2006.

[3] Garg, Shree, et al. "Behaviour analysis of machine learning algorithms for detecting P2P botnets." *Advanced Computing Technologies (ICACT), 2013 15th International Conference on*. IEEE, 2013.

[4] Vania, Jignesh, Arvind Meniya, and H. B. Jethva. "A review on botnet and detection technique." *Int J Comput Trends Technol* 4.1 (2013): 23-29.

[5] Masud, Mohammad M., et al. "Flow-based identification of botnet traffic by mining multiple log files." *Distributed Framework and Applications, 2008. DFmA 2008. First International Conference on*. IEEE, 2008.

[6] Zhao, David, et al. "Peer to peer botnet detection based on flow intervals."*IFIP International Information Security Conference*. Springer Berlin Heidelberg, 2012.

[7] Saad, Sherif, et al. "Detecting P2P botnets through network behavior analysis and machine learning." *Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on*. IEEE, 2011.

[8] Zeidanloo, Hossein Rouhani, et al. "Botnet detection based on traffic monitoring." *2010 International Conference on Networking and Information Technology*. IEEE, 2010.

[9] Garcia, Sebastian, et al. "An empirical comparison of botnet detection methods." *computers & security* 45 (2014): 100-123.

[10] Etemad, Farhood Farid, and Payam Vahdani. "Real-time Botnet command and control characterization at the host level." *Telecommunications (IST), 2012 Sixth International Symposium on*. IEEE, 2012.

[11] Beigi, Elaheh Biglar, et al. "Towards effective feature selection in machine learning-based botnet detection approaches." *Communications and Network Security (CNS), 2014 IEEE Conference on*. IEEE, 2014.

[12] Stevanovic, Matija, and Jens Myrup Pedersen. "An efficient flow-based botnet detection using supervised machine learning." *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE, 2014.

[13] Singh, Raman, Harish Kumar, and R. K. Singla. "Analysis of Feature Selection Techniques for Network Traffic Dataset." *Machine Intelligence and Research Advancement (ICMIRA), 2013 International Conference on*. IEEE, 2013.

[14] Bano, Shehar. *A Study of Botnets: Systemization of Knowledge and Correlation-based Detection*. Diss. Department of computing A thesis submitted in partial fulfillment of the requirements for the degree of Masters in Computer and Communication Security (MS CCS) In School of Electrical Engineering and Computer Science, National University of Sciences and Technology, 2012.

[15] US-Cert, "Malware tunneling in ipv6." https://www.us-cert.gov/sites/default/files/publications/IPv6Malware-Tunneling.pdf, 2005, [Online; accessed 15-December-2011].

[16] https://christuniversity.in