IJCTA, 9(40), 2016, pp. 87-97
© International Science Press

Multi Variant Statistical Model for Improved Botnet Detection in Dynamic Networks Using Particle Swarm Optimization

P. Panimalar* and K. Rameshkumar**

Abstract : The presence of compromised hosts in any network degrades the performance in many ways. The compromised nodes form the botnet and the detection of them is most primary solution to improve the network performance. The problem of botnet detection has been approached in many ways using flow factors but does not produce a remarking solution. To improve the botnet detection efficiency, a novel statistical analysis model has been discussed in this paper. The multi variant statistical analysis model considers payload, TTL, host names, hop count, transmission frequency, and common nodes to perform botnet detection. The statistical model computes stream weight, transmission delay weight, and host weight. The proposed method applies the particle swarm optimization to perform botnet detection. The selection function computes the botnet weight using the result of statistical model.

Keywords: Dynamic Networks, Particle Swarm Optimization, Botnet Detection, Multi Variant Statistical Model.

1. INTRODUCTION

The dynamic network has no fixed topology and the topology changes in fraction. There exists a combination of mobile and non-mobile nodes. The mobile nodes move across the network and involves in cooperative transmission. The source node sends packets towards destination through intermediate nodes. Each node present in the network identifies the routes available and selects an optimal route to reach the destination. The node forward the packet through the selected route where the route selection is performed based on hop count or traffic. There will be varying traffic conditions in the network according to the sending rate of packets. Each node has different traffic condition according to number of neighbors and their sending rate. However the nodes select an optimal route to forward the data packets. If there exist a malicious node called botnet controller then it feeds false information to the neighboring nodes and make them compromised. By generating the compromised node the botnet controller forms the botnet. The nodes from the botnet generate enormous number of malicious packets towards selected destination. This introduces higher traffic in all the intermediate host of any specific route. Also, the presence of higher traffic increases the latency of the network packets. On other side, the packet drop ratio gets increased and reduces the throughput also.

The botnet detection has been approached based on the payload details, but not suitable for modern attacks because even a legitimate node can send huge sized packets. Similarly, the ttl based methods are suitable for less congestive conditions. Such methods are not suitable to handle botnet attacks when it generated in modern sense. The requirement of handling botnet attacks by considering the traffic, flow and delay factors is essential. The compromised nodes from the botnet generate large number of malicious

* Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore-641046, India. E-Mail: *panimalarkp@gmail.com*

** Research Supervisor, Research and Development Centre, Bharathiar University, Coimbatore-641046, India. E-Mail: *rameshkumark.dr@gmail.com*

packets to the destination. This creates more traffic in most of the nodes of the network. Also if they choose the same route, then the intermediate nodes lose their energy in short time and become dead. In this case, the malicious nodes could make the network to fail very shortly. On other side, the botnet nodes can sent the packets in longer route which also creates energy depletion in the intermediate nodes. This also makes the network to fail in short time, so there is efficient botnet detection is required which consider various factors.

This paper discusses about a multi variant statistical analysis model to handle the issue of botnet detection. The proposed method, approach the botnet attacks in three ways: first with the stream orient technique, which monitors the stream feature of the packet. For example, the route followed by the packet and the source has certain style in sending the packets. Suppose, if the source would send maximum of 20 packets in a time, then at the current time if it is sending 40 packets then it is suspicious. By monitoring the streaming of packets and their payloads the method would find out the malicious packets. In the second approach, the method counts the network delay factors as the key to find out the botnet attacks. In this approach, the delay between the source and destination at the previous time will be computed and based on that the decision will be taken. Finally, the method monitors the route being followed by the packet and count for the hops and looks for the common nodes. Based on these factors the method identifies the botnet attacks.

The method takes the advantage of particle swarm optimization methods to take decisions. The PSO technique has been used to decide whether the particular node is suspicious enough to be decided as botnet node. The proposed method use the selection function which computes the measure based on all the features considered. Based on the result from the particle swarm optimization technique, the method decides the trustworthy of the nodes. The application of particle swarm optimization has well adapted to the problem of botnet detection.

2. LITERATURE REVIEW

There are number of methods has been discussed for the problem of botnet detection and this section briefs some of the methods around the problem. Host Based intrusion Detection system [1] offerings intrusion detection system which notifies system administrator about potential intrusion incidence in a system. The intended architecture employee's statistical method of data evaluation, that permits detection based on the knowledge of user activity deviation in the computer system from learned profile representing standard user behavior. Network Intrusion Detection System [2] is proposed which embedded a NIDS in a smart-sensor-inspired device under a service-oriented architecture (SOA) method. Using this embedded NIDS we can able to operate freely as an anomaly-based NIDS, or integrated transparently in a Distributed Intrusion Detection System (DIDS). It combines the advantages of the smart sensor method and the consequent offering of the NIDS functionality as a service with the SOA use to achieve their integration with other DIDS components.

An Activity Pattern Based Wireless Intrusion Detection System [3] is intended for wireless network. It exploits pattern recognition techniques to model the usage patterns of authenticated users and uses it to detect intrusions in wireless networks. The novelty of the recommended system lies in its light-weight design which needs less processing and memory resources and it can be used in real-time environment. EAACK [4], suggest and implement a new intrusion-detection system named Enhanced Adaptive Acknowledgment (EAACK) especially designed for MANETs. EAACK is entailed of three major parts, namely, ACK, secure ACK (S-ACK), and misbehavior report authentication (MRA). In order to separate different packet types in different schemes, they included a 2-b packet header in EAACK.

ANN Based Scheme to Predict Number of Zombies involved in a DDoS Attack [5], present a comprehensive study to show the danger of Botnet-based DDoS attacks on application layer, especially on the Web server and the increased incidents of such attacks that has evidently increased recently. DDoS Attacks Detection by Means of Greedy Algorithms [6], emphasis on DDoS attacks detection by means

of greedy algorithms. In particular they were suggest to use Matching Pursuit and Orthogonal Matching Pursuit algorithms. The major impact of the paper is the proposition of 1D KSVD algorithm as well as its tree based structure representation (clusters) that can be successfully applied to DDos attacks and network anomaly detection. Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art [7], present a comprehensive study to show the danger of Botnet-based DDoS attacks on application layer, particularly on the Web server and the increased incidents of such attacks that has obviously increased recently.

Analyzing Feasibility for Deploying Very Fast Decision Tree for DDoS Attack Detection in Cloud-Assisted WBAN [8], suggest classifying data mining techniques which uses, Very Fast Decision Tree (VFDT) and considered as the most promising solution for real-time data mining of high speed and non-stationary data streams collected from WBAN sensors and therefore is selected, studied and explored for efficiently analyzing and detecting DDoS attack in cloud-assisted WBAN environment. A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment [9], offers a method of integration between HTTP GET flooding among Distributed Denial-of-Service attacks and Map Reduce processing for fast attack detection in a cloud computing environment. The experimental results show that the offered method is better than Snort detection because the processing time of the former is shorter with increasing congestion. A Mulitiprocess Mechanism of Evading Behavior-Based Bot Detection Approaches [10], suggest a new evasion mechanism of bot, multiprocess mechanism. They first identify two specific features of multiprocess bot: separating C&C connection from malicious behaviors, and assigning malicious behaviors to several processes. Then they were implement a single process and multiprocess bot, and use signature and behavior detection methods to evaluate them. The results indicate that multiprocess bot can effectively decrease the detection probability matched with single process bot.

Unified P2P Botnet Detection Using Behavioral Analysis and Graph Analysis [11], propose a novel technique for detecting P2P botnets. Detection is based on unifying behavioral analysis with structured graph analysis. First, their inference technique exploits a fundamental property of botnet design. Modern botnets use peer-to-peer communication topologies which are fundamental to botnet resilience. Second, our technique extends conventional graph-based detection by incorporating behavioral analysis into structured graph analysis, thus unifying graph-theoretic detection with behavioral detection under a single algorithmic framework. They carried out evaluation over real-world P2P botnet traffic and show that the resulting algorithm can localize the majority of bots with low false-positive rate. On the Effectiveness of Different Botnet Detection Approaches [12], investigate four different botnet detection approaches based on the technique used and type of data employed. Two of them are public rule based systems (BotHunter and Snort) and the other two are data mining based techniques with different feature extraction methods (packet payload based and traffic flow based). The performance of these systems is range from 0% to 100% on the five publicly available botnet data sets employed in this work.

A Technique for Detection of Bots Which Are Using Polymorphic Code is discussed in [13]. A new technique of botnet detection which bots use of polymorphic code was proposed in it. Performed detection is based on the multi-agent system by means of antiviral agents that contain sensors. For detection of botnet, which bots use polymorphic code, the levels of polymorphism were investigated and its models were built. A new sensor for polymorphic code detection within antivirus agent of multi-agent system was developed. Developed sensor performs provocative actions against probably infected file, restarts of the suspicious file for probably modified code detection, behavior analysis for modified code detection, based on the principles of known levels of polymorphism. PeerShark: Flow-clustering and conversation-generation for malicious peer-to-peer traffic identification [14], present a methodology to detect P2P botnet traffic and differentiate it from benign P2P traffic in a network. Their approach neither assumes the availability of any ‘seed’ information of bots nor relies on deep packet inspection. It aims to detect the stealthy behavior of P2P botnets. PeerShark combines the benefits of flow-based and conversation-based approaches with

two-tier architecture, and addresses the limitations of these approaches. By extracting statistical features from the network traces of P2P applications and botnets, they build supervised machine learning models which can accurately differentiate between benign P2P applications and P2P botnets.

Improved Detection of P2P Botnets through Network Behavior Analysis [15], proposes a model to distinguish P2P botnet command and control network traffic from normal traffic at higher rate of both the classes using ensemble of decision trees classifier named Random Forests. Further to optimize the performance, this model also addresses the problem of imbalanced nature of dataset using techniques like down sampling and cost sensitive learning. Performance analysis has been done on the proposed model and evaluation results show that true positive rate for both botnet and legitimate classes are more than 0.99 whereas false positive rate is 0.008. P.Panimalar et al. [16] have proposed an approach time orient multimodal traffic analysis for efficient Botnet detection based on time orient behavior analysis and time orient traffic analysis and stream approach. Even the method has produced efficient results in terms of performance and accuracy, the time complexity has to be reduced. Phoenix: DGA-Based Botnet Tracking and Intelligence [17], propose Phoenix, a mechanism that, in addition to telling DGA- and non-DGA-generated domains apart using a combination of string and IP-based features, characterizes the DGAs behind them, and, most importantly, finds groups of DGA-generated domains that are representative of the respective botnets. As a result, Phoenix can associate previously unknown DGA-generated domains to these groups, and produce novel knowledge about the evolving behavior of each tracked botnet.

Hybrid Botnet Detection Mechanism [18], analyzes layouts of different detection techniques. The paper tries to find features that, when combined together, complement each other's strengths and eliminate the weaknesses and suggests a framework consisting of a combination of those features which, theoretically, should overcome most of the common problems faced by detection techniques. Detecting HTTP Botnet using Artificial Immune System [19], proposed a new general HTTP Botnet detection framework for real time network using Artificial Immune System (AIS). Generally AIS is a new bio-inspired model which applies to solving various problems in information security; we used this concept in our proposed framework to make it more efficient in detection of HTTP Botnet. Our experimental evaluations show that our approach can detects HTTP Botnet activities successfully with high efficiency and low false positive rate. All the above discussed methods have the problem of poor botnet detection accuracy. Also the methods produce less time complexity.

3. MULTI VARIANT STATISTICAL MODEL

The Multi variant statistical model handles the problem of botnet detection in three ways. The model monitors the incoming packets and extracts payload, ttl, hop count, host names from the packets received. The extracted features are stored in the log for further usage. The model first applies the stream analysis which computes the rate of packet reception from any particular source to compute the stream weight. The delay approximation model computes the delay factor from the trace available for the same source. Based on the delay factor the method computes the transmission weight. The third model generates the common hops from the trace and their hop count. Based on the features computed, the method computes the routing weight. All these three models are applied through the Particle Swarm optimization technique to compute the selection weight. Based on the weight computed, that performs botnet detection. Figure 1, shows the architecture of multi variant statistical model and the stages. This section explains each stage of the proposed algorithm in detail.

A. Feature Extraction

The feature extraction method extracts the packet features like payload, TTL value, Hop count and host names from the packets received. For each packet being received by the destination node, the method extracts the above mentioned features. The extracted features are converted into feature vector. Constructed feature vector is added to the network trace and will be used to perform botnet detection.

Pseudo Code of Feature Extraction :

Input : Packet Pt.

Output : Feature Vector Fv.

Start

Read Input Packet Pt.

Extract Source Address $sa = \int$ Source-Addr $\in$ Pt

Extract

Payload $pl = \int$Payload(Pt)

Extract TTL $ttl = \int$Pt.TTL

Extract Hop count Hc = Size($\sum$ Host $\in$ Pt.Traversalpath)

Extract host names Hnames = $\sum$ Host $\in$ Pt.Traversalpath

Generate Feature Vector Fv = {*sa*, *pl*, *ttl*, Hc, Hnames}.

Stop.

The Feature extraction algorithm receives the packet and extracts the above mentioned features. Extracted features are used to construct the feature vector and added to the access trace.

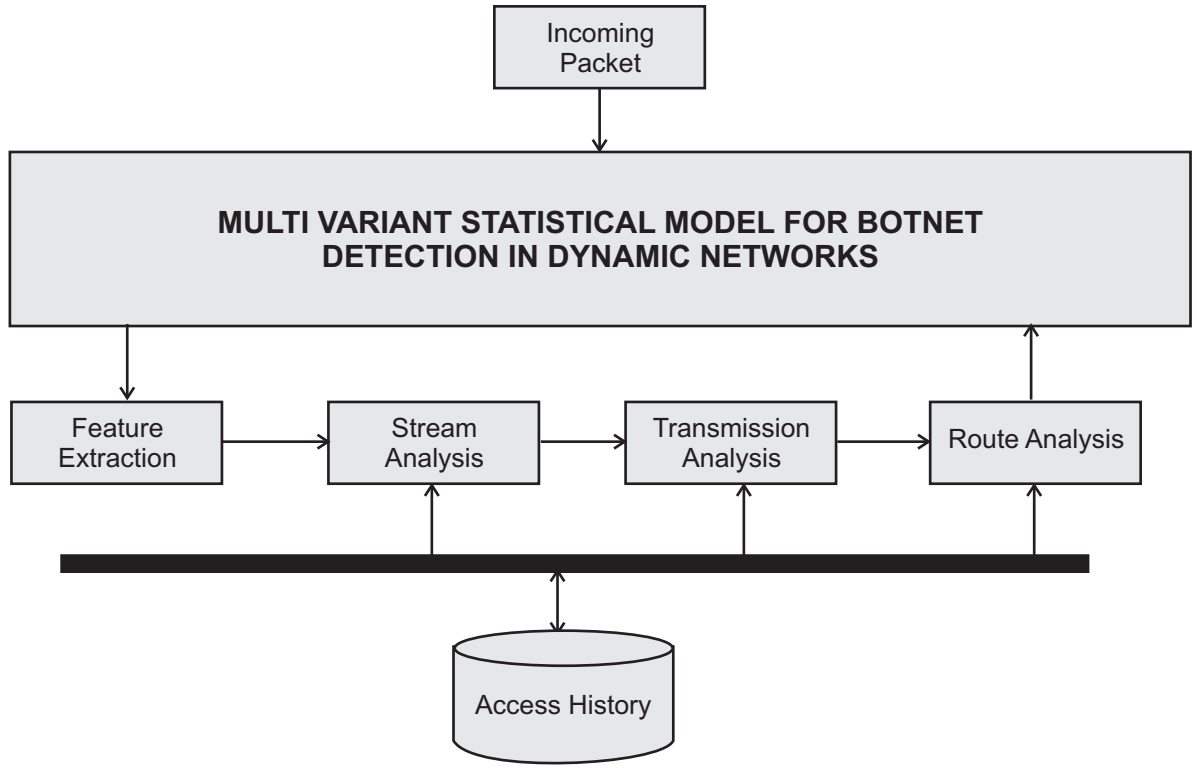

Figure 1: Architecture of Multi Variant Statistical Model

B. Stream Analysis

The nodes send packets through intermediate nods to the destination node. The packets pass through number of intermediate nodes. Each source node sends number of packets to the destination. The number of packets sent by any genuine node will be within the limit. When the number of packets sent crosses certain limit then the stream can be identified as malicious. Also the packet size or payload of packets will be within a limit in general case. If the payload feature reaches a limit then it can be concluded as malicious. Based on above two corollary, the stream analysis is performed. From the result of feature extraction and from the access trace, the method collects set of traces belongs to the source node.

Pseudo Code of Stream Analysis:

Input : Access Trace ATr, Feature Vector Fv.

Output: Stream Weight Sw

Start

Read Access Trace Atr.

Read Feature Vector Fv.

Initialize payload support psup.

Extract all traces belongs to the source node.

User Trace $\mathrm{Ut} = \int_{i=1}^{\mathrm{sir}(\mathrm{Atr})} \sum \mathrm{Atr}(i).\mathrm{SAddress} == \mathrm{Fv.SAddress}$

For each trace Ti from UT

$$\mathrm{Psup} = \int_{i=1}^{\mathrm{size}(\mathrm{UT})} \sum \mathrm{size}(\mathrm{Fv.payload})$$

End

Compute Stream Frequency $sf = \dfrac{\mathrm{Size\ (UT)}}{\mathrm{Size\ (Atr)}}$

Compute Payload factor $\mathrm{Pf} = \dfrac{\mathrm{psup}}{\mathrm{size\ (Ut)}}$

Compute Stream Weight $\mathrm{Sw} = \dfrac{\mathrm{St} \times \mathrm{Pf}}{100}$

Stop.

Using the sub set of traces; the method computes the stream frequency and stream payload factor. Based on features computed, the method computes the stream weight. The stream analysis algorithm computes the stream frequency and payload factor. Using computed measures, the algorithm computes the stream weight.

C. Transmission Analysis

The packets from the source node have to travel through number of intermediate nodes. Each intermediate node present in the transmission path has certain amount of traffic and the traffic introduce certain amount of delay. Totally in general condition, there will be a delay within certain amount. When there is a presence of malicious node or the entry of botnet node will increase the delay value. The botnet compromised nodes would involve in modification attack or it may involve in learning the packet features. This in turn increase the delay time of the packet delivery. So by maintaining the sent time as a special field with the packet structure, the destination node can compute the transmission delay. Once the delay of packet being received is higher than the delay learned from previous traces then it can be concluded that there exist a malicious node. The transmission analysis approach computes the delay factor for the source node. The delay factor for the same source at the previous stage and current stage is computed. Based on the delay factors computed the method computes the transmission weight for the source node. Computed transmission weight will be used by the botnet detection algorithm.

Pseudo Code for Transmission Analysis:

Input : Access Trace Atr, Feature Vector Fv.

Output : Transmission Weight Tw.

Start

Read Access Trace Atr.

Read Feature Vector Fv.

Initialize Delay Factor Df.

Extract all traces belongs to the source node.

User Trace $\mathrm{Ut} = \int_{i=1}^{\mathrm{sir\ (Atr)}} \sum \mathrm{Atr}(i).\mathrm{SAddress} = = \mathrm{Fv.SAddress}$

For each trace Ti from UT

Delay factor $\mathrm{Df} = \int_{i=1}^{\mathrm{sir\ (UT)}} \sum \mathrm{UT}(i).\mathrm{RTime} - \mathrm{UT}(i).\mathrm{Stime}$

End

Compute Delay Factor $\mathrm{Df} = \dfrac{\mathrm{Df}}{\mathrm{Size(Atr)}}$

Stop.

The transmission analysis algorithm computes the delay factor for the source node and computed delay factor will be used for botnet detection.

D. Route Analysis

The packets sent from the source node travels through number of intermediate nodes to reach the destination. There will be N+1 number of routes available between any sources to destination. To reach the destination there will be shortest route and there will be power efficient route available. In general case, the intermediate node or the source node will choose the shortest route. Also the number of hops traversing will be less and when the malicious node or botnet comes, then the number of hops would be increased. The malicious node would try to forward the packet through longest route in the sense to increase the packet drop ratio. In other case, if the malicious node tries to perform modification attack then the packet will be traversed through the same route where there is a compromised node. To handle this, the method identifies the common nodes present in the malicious history. The same will be identified in the packet transmission route. Based on the information about the common hosts, the method computes the route factor and the method computes the route weight. Computed route weight will be used to perform botnet detection.

Pseudo Code of Route Analysis:

Input : Malicious Trace Mt, Feature Vector Fv, Access Trace Atr

Output : Route Weight Rw.

Start

Read Access Trace Atr.

Read Malicious Trace Mt.

Read Feature Vector Fv.

Extract all traces belongs to the source node.

User Trace $\mathrm{Ut} = \int_{i=1}^{\mathrm{sir\ (Atr)}} \sum \mathrm{Atr}(i).\mathrm{SAddres} = = \mathrm{Fv.Address}$

Identify distinct routes Dr.

$\mathrm{Dr} = \int_{i=1}^{\mathrm{size\ (Ut)}} \sum \mathrm{Ut}(i).\mathrm{Route} \not\in \mathrm{Dr}$

For each route Ri from DR

Identify host names.

$\mathrm{Hn} = \int_{i=1}^{\mathrm{size\ (DR)}} \sum \mathrm{Host}(\mathrm{Dr}(i)) \not\in \mathrm{Hn}$

End

Identify common host names CHn = $\int \sum$ Host $\in \forall$(Ri(DR))

Compute route factor $rf = \int_{i=1}^{\text{size (Mt)}} \frac{\sum \text{Mt}(i) \in \forall(\text{CHN})}{\text{size(Mt)}}$

Compute number of malicious packets followed the route Rk.

$\text{Nr} = \int_{i=1}^{\text{size (Mt)}} \sum \text{Mt}(i).\text{Route} = = \text{Fv. Route}$

Compute route weight $rw = rf \times \text{Nr}$

Stop.

The route analysis algorithm computes the route factor and computes the route weight. Computed route weight will be used to perform botnet detection.

E. PSO Based Botnet Detection

The botnet detection is performed with the particle swarm optimization technique. The method receives the packet and extracts the packet features. Extracted packet features will be applied for particle swarm optimization technique. The PSO technique computes stream weight, transmission weight and route weight. If the stream weight and transmission weight are greater than particular threshold then the selection algorithm computes the route weight for different hops identified. Based on the weight of route the method concludes the presence of botnet and declares the hosts of botnet.

Pseudo code of Botnet Detection:

Input : Packet P

Output : Null

Start

Receive packet P.

Feature Vector Fv = Perform Feature Extraction.

Compute Stream Weight *sw* = Stream Analysis (*fv*).

Compute Transmission weight *tw* = Transmission Analysis (Fv).

If Sw > STh and Tw > TTh Then

For each host Hi from Fv

Compute route weight *rw* = Route Analysis (Fv).

If *rw* > RTh then

Declare malicious.

End

Add route to malicious history.

End

End

Stop.

The PSO based botnet detection algorithm computes stream weight, transmission weight and route weight to perform botnet detection.

4. RESULTS AND DISCUSSION

The multi model statistical model for botnet detection has been implemented and evaluated for its efficiency in botnet detection. The approach has been validated with different data sets and real time analysis. The details of simulation have been presented and Table 1 shows the details of simulation being used to evaluate the performance of the proposed multi model approach for botnet detection. Graph 1

shows the comparative result on botnet detection produced by different methods and it shows clearly that the proposed method has produced more efficient accuracy than other methods. Graph 2, shows the comparison of false classification ratio produced by different methods and it shows clearly that the proposed method has produced less false classification ratio than other methods.

Table 1

Details of simulation being used

Parameter	*Value*
Simulator	Advanced Java
Number of Nodes	200
Simulation Time	5 Minutes
Protocol	RTI
Simulation Area	1000 × 1000 meters
Node Range	100 meters

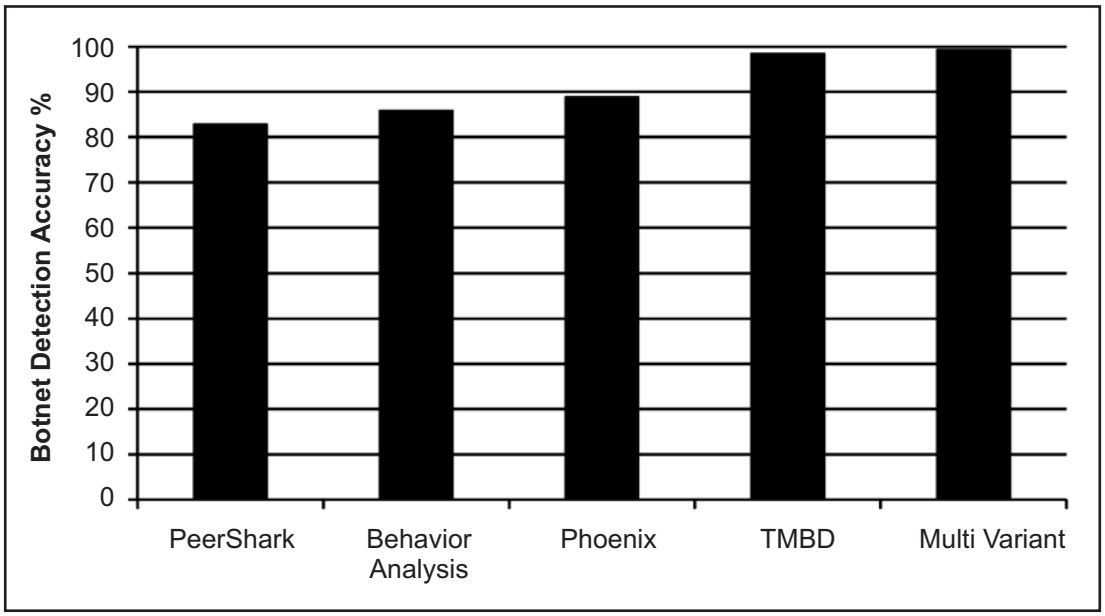

Figure 2: Comparison of botnet detection accuracy

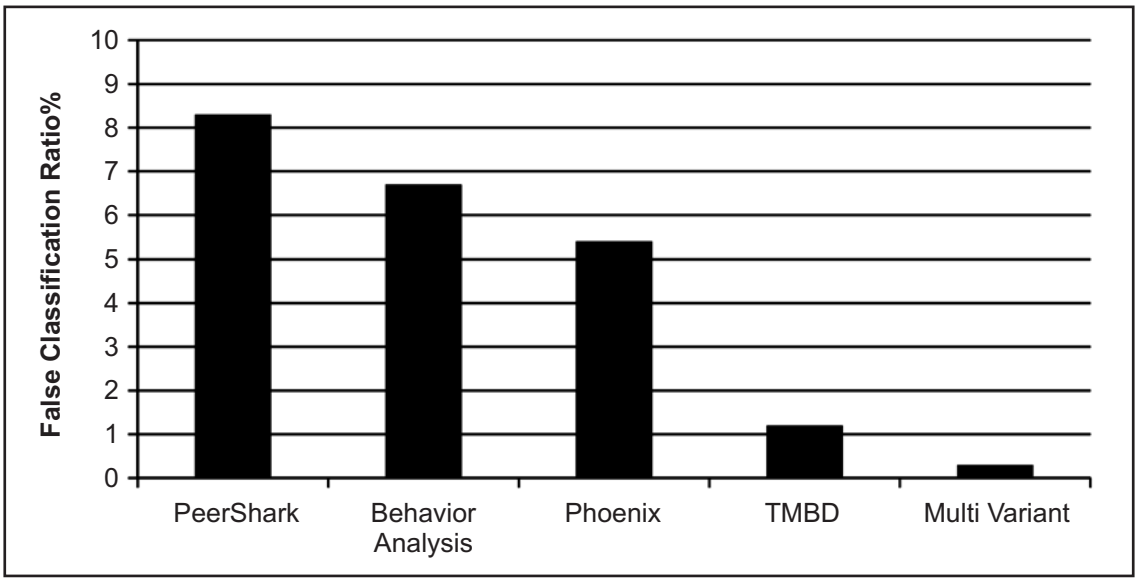

Figure 3: Comparison of false classification ratio

Graph 3, shows the comparative result of time complexity produced by different methods and it shows clearly that the proposed method has produced less time complexity than other methods.

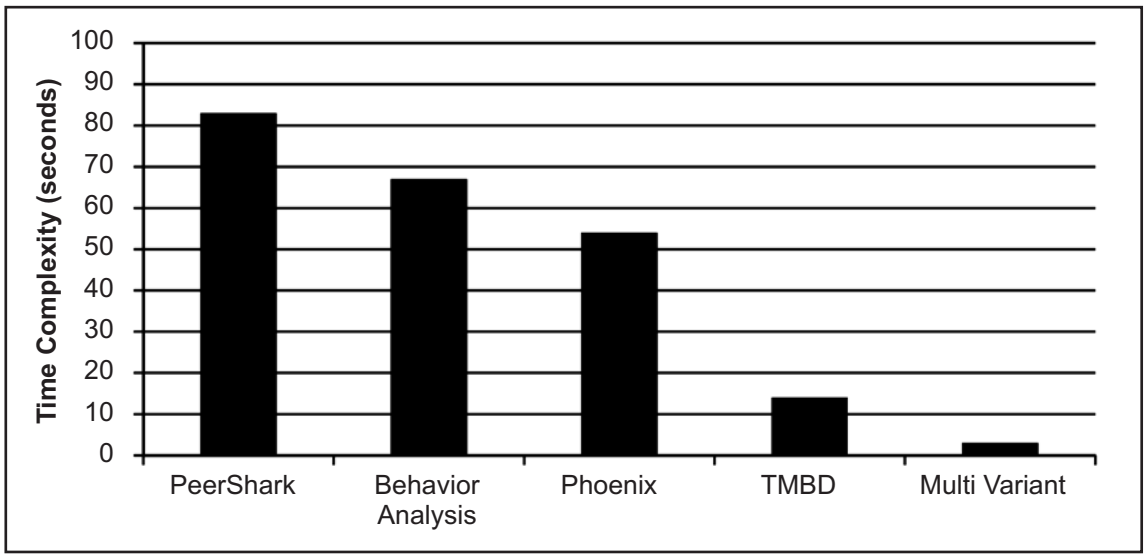

Figure 4: Comparison of time complexity

5. CONCLUSION

The problem of botnet detection has been approached in different ways and the proposed method receives the packet and extracts the features from the packet. The multi variant statistical model performs stream analysis, transmission analysis and route analysis to compute different weight factors. Based on the weight factors computed the particle swarm optimization approach has performs botnet detection. The time taken by the proposed method is very less comparing with the other methods in the same context of the experiment. The proposed method has produced efficient results in botnet detection and reduces the false classification ratio.

6. REFERENCES

1. Vokorokos, L,"Host Based Intrusion Detection System", Intelligent Engineering Systems (INES), pp 43 – 47, 2010.
2. Maciá-Pérez, F, "Network Intrusion Detection System Embedded on a Smart Sensor", Industrial Electronics, IEEE Transactions, Vol.58, Issue 3 pp. 722 – 732, 2012.
3. Haldar, N.A.H,"An Activity Pattern Based Wireless Intrusion Detection System", Information Technology: pp. 846 – 847, 2012.
4. Elhadi M. Shakshuki, "EAACKA- Secure Intrusion-Detection System for MANETs", IEEE Transactions on Industrial Electronics, Vol. 60, No. 3, March 2013.
5. B. B. Gupta, R. C. Joshi, ManojMisra, "ANN Based Scheme to Predict Number of Zombies involved in a DDoS Attack," International Journal of Network Security (IJNS), Vol. 14, No. 1, pp. 36-45, 2012.
6. Tomasz Andrysiak, ŁukaszSaganowski, MichałChoraś, "DDoS Attacks Detection by Means of Greedy Algorithms", Image Processing and Communications Challenges 4, Advances in Intelligent Systems and Computing, Vol 184, 2013, pp 303-310.
7. EsraaAlomari, SelvakumarManickam, B B Gupta, Shankar Karuppayah and RafeefAlfaris. "Article: Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art", International Journal of Computer Applications, 49(7):24-32, July 2012.
8. RabiaLatif, Haider Abbas, Saïd Assar, SeemabLatif, "Analyzing Feasibility for Deploying Very Fast Decision Tree for DDoS Attack Detection in Cloud-Assisted WBAN", Springer, Intelligent Computing theory, Vol 8588, pp 507-519, 2014.
9. Junho Choi, Chang Choi, ByeongkyuKo, Pankoo Kim, "A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment", Springer, Soft computing, Vol 18, Issue 9, pp 1697-1703, 2014.

10. Yuede Ji, Yukun He, Dewei Zhu, Qiang Li, Dong Guo, "A Mulitiprocess Mechanism of Evading Behavior-Based Bot Detection Approaches", Springer, Information Security Practice and Experience, Vol 8434, pp 75-89, 2014.

11. Shishir Nagaraja, Botyacc, "Unified P2P Botnet Detection Using Behavioural Analysis and Graph Analysis", Springer, Computer Security - ESORICS 2014 Lecture Notes in Computer Science Vol 8713, pp 439-456 , 2014.

12. Fariba Haddadi, Duc Le Cong, Laura Porter, "A. Nur Zincir-Heywood, On the Effectiveness of Different Botnet Detection Approaches", Springer, Information Security Practice and Experience Lecture Notes in Computer Science, Vol 9065, 2015, pp 121-135.

13. Oksana Pomorova, Oleg Savenko, Sergii Lysenko, Andrii Kryshchuk, Andrii Nicheporuk, "A Technique for Detection of Bots Which Are Using Polymorphic Code", Springer, Computer Networks Communications in Computer and Information Science, Vol 431, pp 265-276, 2014.

14. Pratik Narang, Chittaranjan Hota, VN Venkatakrishnan, PeerShark: "Flow-clustering and conversation-generation for malicious peer-to-peer traffic identification", Springer, EURASIP Journal on Information Security, October 2014.

15. Shree Garg, Anil K. Sarje, Sateesh Kumar Peddoju,"Improved Detection of P2P Botnets through Network Behavior Analysis", Springer, Recent Trends in Computer Networks and Distributed Systems Security Communications in Computer and Information Science, Vol 420, pp 334-345, 2014.

16. P. Panimalar and K.Rameshkumar, "Time Orient Multi Model Traffic Analysis for Efficient Botnet Detection in Internet Communication", International Journal of Applied Engineering Research, Vol 10, no. 21, pp. 42183-42188, 2015.

17. Stefano Schiavoni, Federico Maggi, Lorenzo Cavallaro, Stefano Zanero, "Phoenix: DGA-Based Botnet Tracking and Intelligence", Springer, Detection of Intrusions and Malware, and Vulnerability Assessment Lecture Notes in Computer Science, Vol 8550, pp 192-211, 2014.

18. Katha Chanda. "Article: Hybrid Botnet Detection Mechanism", International Journal of Computer Applications, 91(5):12-16, April 2014.

19. Amit Kumar Tyagi and Sadique Nayeem. "Article: Detecting HTTP Botnet using Artificial Immune System (AIS)", International Journal of Applied Information Systems, 2(6):34-37, May 2012.