International Journal of Security and Its Applications
Vol. 10, No. 9 (2016), pp.39-44
http://dx.doi.org/10.14257/ijsia.2016.10.9.05

Mobile Botnet Detection Model based on Retrospective Pattern Recognition

Meisam Eslahi[1*], Moslem Yousefi[2], Maryam Var Naseri[3], Y.M.Yussof [1], N.M. Tahir[1] and H. Hashim[1]

[1]*Faculty of Electrical Engineering, Universiti Teknologi MARA, Malaysia*
[2]*Center for Advanced Mechatronics and Robotics, Universiti Tenaga Nadional Malaysia*
[3]*Advanced Informatics School, University Technology Malaysia, Malaysia.*
[1]*eslahi@ieee.org, yusna233@salam.uitm.edu.my, nooritawati@ieee.org, habib350@salam.uitm.edu.my,*
[2]*Yousefi.moslem@gmail.com*
[3]*varnaseri.m.v@ieee.org*

Abstract

The dynamic nature of Botnets along with their sophisticated characteristics makes them one of the biggest threats to cyber security. Recently, the HTTP protocol is widely used by Botmaster as they can easily hide their command and control traffic amongst the benign web traffic. This paper proposes a Neural Network based model to detect mobile HTTP Botnets with random intervals independent of the packet payload, commands content, and encryption complexity of Bot communications. The experimental test results that were conducted on existing datasets and real world Bot samples show that the proposed method is able to detect mobile HTTP Botnets with high accuracy.

Keywords*: HTTP Botnets, BYOD, Mobile Security, Botnet Detection, Traffic Analysis*

1. Introduction

Botnet is a network of sophisticated connected malwares called Bots which are designed to stealthy infect different devices (computers and mobiles) and remain active and undetectable as long as possible [1]. There are a number of characteristics that differentiate Bots from traditional malware (*e.g.*, Virus, Worms, and Trojans) such as use of standard protocols and services, flexibility in operations and attacks, and minimum resource usages on infected machines to reduce the chance of detection and elimination [2]. Moreover, the Bots and Botnets are developed and maintained by high skills attackers known as Botmasters who control the Bots via command and control mechanism or C&C. Since the main aim of the Botmaster is to make illegal profits they constantly evolve their Botnets by designing foolproof communication mechanism to deliver new commands to Bots, receive attack reports and bots' status, and regularly update their Botnet to harden the code and binaries [3, 4].

The latest generation of Botnets employed HTTP Protocol as it allows Botmasters to impersonate their activities as normal HTTP flows which make them more stealth and hard to detect. In addition, since the HTTP service is widely used by Internet applications it is not easy to block [2]. Therefore, the HTTP Botnets are considered as one of the most sophisticated type of Botnets which are widely used by Botmasters [5]. Although the Botmasters keep evolving their Botnets, for almost two decades their activities were

* Corresponding Author: Meisam Eslahi (eslahi@ieee.org)

ISSN: 1738-9976 IJSIA

mainly limited to the computer and computer networks before migrating to mobile networks [6].

In recent years, mobile devices have quickly become an integral part of every daily activity. The advanced capabilities of smart phones such as sufficient process power along with high speed Internet has motivate more organizations to implement BYOD or Bring Your Own Device at workplaces [7]. The BYOD has brought numbers of benefits for organizations such as cost efficiency, mobility and flexibility; however, it has become an attractive environment for Botmaster as since mobile devices are not properly protected compared to computer, and their users pay less attention to the security updates [8]. The Zitmo, DroidDream, Smartroot, AnserverBot, Ikee.B and TigerBot are the real world examples of mobile Botnets that can be named as proof of the concept that Botmasters have selected mobile network as a new platform for their malicious activities [6]. On the other hands, the current BYOD security models (*e.g.*, MDM, MAM, and MIM) are not adequate enough to protect the organizations from sophisticated cyber threats such as Botnets [9]. Therefore, more research is required to address the aforementioned issues.

As pointed by Eslahi *et al.* [10] there is a considerable gap between current studies and the real state of mobile Botnets. Despite the fact that the main focus of researchers is to propose or detect SMS-based Mobile Botnets, the majority of existing mobile Botnets communicate via HTTP protocol and web services [11]. Therefore, this paper aims to propose a detection approach for HTTP Botnets on BYOD networks. The main objectives of this research is achieved by realization of the following steps: (i) Extraction of HTTP traffic features and proposing new metrics, (ii) Design and implementation of Neural Network based detection model, (iii) Data collection and formation of Botnet dataset, (iv) Evaluation of the proposed model performance.

2. Related Works

In general, the current mobile malwares and Botnets detection approaches fall into two categories such as operational behavior of malwares in mobile devices, and Botnet command and control traffic analysis. The first category focuses on the behavior of malwares on mobile devices such as the amount of resource usage, system calls, android permissions, and etc. [12]. Regardless of advantages and disadvantages of this method, it works with traditional mobile malwares and it is less effective for Botnets as the Bots are silent and sophisticated threats that do not make any unusual or suspicious use of the battery, CPU, memory, or other mobile resources, which will otherwise, cause their presence to be exposed [2].

Unlike the aforementioned techniques, the second category aims to detect Botnets by analyzing the command and control mechanism where the Bots on infected devices communicate with their Botmaster via SMS or Internet to receive new commands and updates. Although, most of the current studies focus on proposing new commands and control mechanisms for mobile Botnets instead of detecting, mitigating, and responding to them [6], there are number of researches on mobile Botnet detection. Table 1 summarizes existing mobile Botnet detection with respect to the type of command and control (*i.e.*, SMS or HTTP), ability of random pattern detection, and use of effective Botnet detections approaches such as retrospective and cooperative analysis approaches.

Table 1. Current Studies on Mobile Botnet Detection

Related Works	*C&C Model*	*Periodicity Detection*	*Random Pattern Detection*	*Retrospective Detection*	*Cooperative Analysis*
[13]	SMS	✓	×	✓	×
[14]	HTTP	×	×	×	×
[15]	HTTP	✓	×	×	✓
[16]	HTTP	×	×	×	×
[17]	SMS	×	×	×	×
[18]	SMS	×	×	×	×

One of the early studies on mobile Botnet detection conducted by Vural *et al.* [13] suggest a number of SMS-oriented metrics such as volume of incoming and outgoing SMSs, sending delays along with their median to propose a forensics model for Botnet detection. Moreover, a retrospective approach is used to increase the accuracy of the proposed method in which consistency of SMS volume is analyzed in the duration of one week. Finally, a correlation model is employed to determine the repeating patterns in the history of single device activities. Although they employed retrospective-based detection to look for repeating patterns, they only investigate a history of single devices instead of group of devices within a network (*i.e.*, Cooperative Analysis). The cooperative analysis approach is one of the successful methods that rely on the fact that the Bots belonging to a Botnet pose similar activities. Thus, monitoring the group activities and similarity analysis has been successfully employed by several studies on computer-based Botnets detection [19]. Finally, their method is less effective to deal with Botnets that have random behavior as they look for similar and fix volume and delays of SMSs that are regularly generated form an infected device. The Bots can easily generate random delay or volume to bypass the proposed detection method [2].

Accordingly in [17], the authors proposed a set of selected SMS-based features such as phone number (both sender and receiver) along with specific words, URLs, and phone numbers in SMS text. Besides the fact that their SMS signature machining method is less effective on detection of zero-day Bots, a simple SMS encryption and obfuscation can be used to bypass the proposed method as it relies on scanning of SMS content. Finally, the Johnson *et al.* [18] employed features include Intent of sent and received SMSs, average, median and standard deviation of response time to detect SMS-based Botnet. This method is also suffering from aforementioned shortcomings.

Regardless of the advantages and disadvantages of SMS-based mobile Botnet detection methods, they cannot be considered as an effective method as the SMS is not a preferred medium for Botmasters to establish their command and control mechanism. The SMS-based C&C design is based on peer-to-peer structure, thus, it comes with a high level of command latency and complexity in implantation [6]. Moreover, SMS messages are not free; thus, costs of messages may notify mobile device owners and disclose suspicious activities. Therefore, the majority of real world mobile Botnets such as Geinimi, Anserverbot, and Beanbot employ HTTP and WEB technology as the mobile device are well integrated with the Internet [8].

Based on the literature, there are only a few numbers of studies on mobile HTTP Botnet detection as listed in Table 1. Su *et al.* [14] propose a dual defense approach which comprises a combination of static system calls analysis and network traffic monitoring. The system calls module considers the fifteen system calls to propose a classifier that distinguishes normal applications from malwares. As the sophisticated Botnets pose less abnormal activities on the mobile client, they have also proposed network traffic monitoring (NTM) to look for any evidence of Botnet command and control mechanisms. The NTM was designed based on selected traffic features such as average and standard deviation of the packets number (sent/received), the bytes' number and the average TCP/IP session duration. There are several benign applications such as

location-based services, Gmail sessions, and auto refresh pages which generate same values that increase the rate of false positive [2, 19].

Byungha *et al.* [15] proposed a VPN-based detection approach that consists of three components such as the abnormal model, whitelist, and signature matching. The abnormal model employed the packet features such as sum and average of packets, bytes, and flows. These factors are not generalized and can vary from Bot to Bot. Moreover, they measured the periodical behavior of HTTP Botnets as they are connected to their command and control server in regular interval [20]. However, the proposed model only considers the request that are generated with the same time interval (*e.g.*, every 5 minutes), thus, the Bots with random intervals can easily evade this module.

Finally, Feizollah *et al.* [16] conducted a study on efficiency of machine learning classifiers such as Naïve Bayes, k-nearest neighbor, decision tree, multi-layer perceptron, and support vector machine in mobile Botnet detection. However their selected features such as connection duration, TCP size, and Number of parameters in GET/POST methods are not sufficient enough to detect sophisticated mobile Botnets. The proposed connection duration relies on the assumption that most of the HTTP Bots communications consist of a simple TCP handshake only. This feature is not that accurate as the Bot communication time might vary since the command and control mechanism involves in different activities such as updating, getting new commands, submitting reports to the Botmasters and *etc.* Moreover, as discussed earlier, generating random values is one of the common evasion techniques used by Botmasters. The proposed TCP size feature claimed to be effective as the Bots are designed to steal users information and thus their traffic TCP size is distinguishable (*i.e.*, having large TCP size) from normal activities. A normal mobile download manager and updater can similarly generate traffic with distinguishable TCP size. In fact the simple numeric parameters such as size, numbers, and duration are not that accurate to use in malware detection in comparison with variability measurements such as Standard deviation, Shannon entropy, Range, and *etc.* as they look for pattern of data changes instead of simply focusing on actual data volume as a factor to make decision [20]. The last feature counts the number of parameter transferred by HTTP methods (*i.e.*, Get and Post). This assumes that the mobile Bots' requests are generated with higher parameters as they are designed to steal sensitive information on mobile devices and send them to Botmaster. The GET and POST methods are also used by normal applications to send and receive data from server. Moreover, the Bots are not designed to steal sensitive information only, but they are used as platform to conduct several type of cyber-attacks such as DDOS, Adware, Illegal Transactions, Spamming and *etc.* in which less data (*i.e.*, less GET and POST parameters) is sent to the Botmaster [1, 2]. The aforementioned discussion shows that the current studies on mobile Botnet detection and HTTP Bots, in particular, come with significant shortcomings. Therefore, this paper aims to propose a framework for mobile Botnet detection using new features, correlation and retrospective analysis approaches as explained in the next section.

3. Proposed Architecture and Detection Methodology

This paper proposes a passive mobile Botnet detection in which particular mobile network traffic is collected for a period of time and analyzed in order to identify any signs of Bots and Botnet activities. Figure 1 illustrates the proposed architecture.

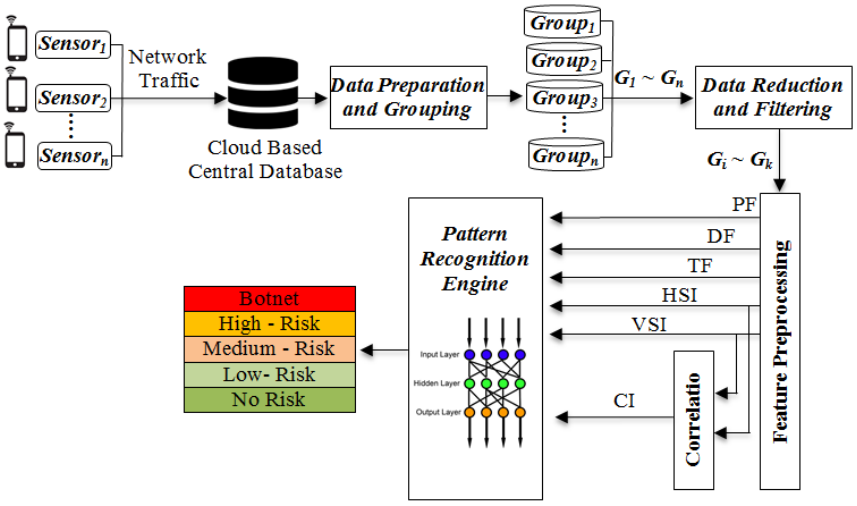

Figure 1. Proposed Method Architecture

3.1. Data Collection, Preparation and Grouping

The data collecting is conducted by a set of light applications such as Tpacketcapture [21] that are installed on each mobile device to collect network traffics and store them into a local repository (*i.e.*, Device Memory). The collected data is automatically transferred to a central database located in the cloud infrastructure as it comes with significant advantages such as ease implementation, ease of maintenance, reliability, scalability, guaranteed levels of services, and efficient total cost of ownership [22]. During the data preparation stage a simple data filtering is applied on collected network traffic to select HTTP traffic only specially GET and POST methods as the HTTP based Bots use these methods to communicate with their Command and Control server [19, 20]. In addition, the selected HTTP traffics are divided into different groups based on the similarity of their source IP address, destination IP address, domain name and URL.

3.2. Data Reduction and Filtering

The different groups of collected traffic are examined using two data filtering approaches to reduce the amount of unwanted data and consequently minimize the data analysis processing time and power. As highlighted by Strayer *et al.* [23], the Bots are designed to generate given tasks more faster than human, therefore, they do not generate the brief data. Thus, the first data filtering removes all groups with a low number of members which mean only a few HTTP packets are observed in a group for the entire data collecting period. On the other hand, The Botmasters aim to keep their Bots alive as long as possible; hence, the Bots do not generate the bulk data which would lead them into being detected [23]. Thus the second data filtering removes the HTTP connections which are generated extremely fast by updaters and downloaders.

3.3. Feature Preprocessing

The feature preprocessing extracts 7 different parameters from the collected network packet which are divided in different groups. The first three parameters are proposed to measure the level of periodicity as summarized in Table 2.

Table 2. Extracted Features to Measure the Level of Periodicity

NO	*Feature*	*Description*
1	Periodic Factor (PF)	A numeric factor to measure the level of periodicity of HTTP traffic and generated requests
2	Diversity Factor (RD)	A Shannon Entropy based factor to measure the density of generated requests in time windows
3	Time Difference Factor (TF)	A Binary value to define whether the HTTP request occurred with random or fixed interval

3.3.1. Third-order Headings: The periodic factor or PF is a numeric measure between 0 and 1 which respectively represents the least (0) and the most periodic (1) pattern. To calculate the PF the entire data collection is divided into several time windows with fixed size as shown in the Figure 2.

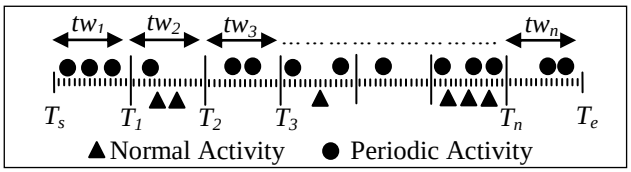

Figure 2. Total Data Collection Duration and Time Windows

For each time window we define a binary value called *O*. Thus, if the data collection time is divided to *n* time windows, we will have *n* number of *O* values such as O_1 to O_n. For each group of collected data (*e.g.*, Gi), the O_k will be set to 1 if at least one of the group members (*i.e.*, HTTP requests) is observed in time window *K*, otherwise it will be set as 0. Finally, the periodic factor of each group can be calculated by sum of *O* values divided by total number of time window as shown in formula 1.

$$PF = \frac{\sum_{tw=1}^{n} o}{n} \tag{1}$$

For instance the triangle activity in figure 2 is observed in 3 out of 7 time windows, thus the PF for that group will be 3/7=0.42. Intuitively, activities are considered more periodic if the PF is closer to 1.

3.3.2. Uniformity Factor (UF): The UF factor is designed to determine the diversity of similar requests in time windows. It is important to determine whether the same number of requests generated by similar application in each time window or not as the density of generated requests plays significant role in Botnet detection. For each group of data the number of associated activities (*i.e.*, absolute frequencies) is counted per each time window. The distribution entropy of collected absolute frequencies can be used as a measure to identify the density of similar activities. The Shannon entropy shown in formula 2 is used in this research as it is significantly employed by a number of studies as a diversity indices [24].

$$H(X) = \sum_{i=1}^{n} P(\mathrm{x_i})\,\mathrm{I}(\mathrm{x_i}) = -\sum_{i=1}^{n} \mathrm{P}(\mathrm{x_i})\log_2 \mathrm{P}(\mathrm{x_i}) \tag{2}$$

The smaller Shannon entropy represents the higher uniformity in requests density in time windows. Instinctively, the Shannon entropy value of 0 indicates that an equal

number of requests were generated in each time window and can be considered as an indicator of Bot activities.

3.3.3. Time Difference Factor (TDF): As discussed earlier the mobile HTTP Botnets periodically visit a certain number of websites that belong to their Botmaster to receive new commands or updates based on preconfigured intervals [12]. The TF is a binary value to define whether the activities in a group occurred with random (TF=0) or fixed (TF=1) intervals. To compute the TF, an ordered list of request timestamps is formed. If the terms of the timestamps list have a common difference the value of TF will be set as 1, otherwise it will be set as 0.

The rest of features are proposed to conduct a cooperative analysis approach as they look for any similarities in mobile devices activities. Table 3 summarizes the similarity based features.

Table 3. Extracted Features to Determine the Similarities

NO	*Feature*	*Description*
1	Packet Size Diversity (PD)	An standard deviation based factor to measure the diversity of packet sizes of HTTP request
2	Horizontal Similarity Index (HSI)	A numeric factor used Jaccard index to measure the level of similarities in a mobile device history
3	Vertical Similarity Index (VSI)	A numeric factor used Jaccard index to measure the level of similarities between mobile device history and other devices in the network
4	Correlation Index (CI)	A mathematical parameter to measure link between mobile device activities (HSI) and network activities (VSI)

3.3.4. Packet Size Diversity (PD): The packet size is one of the common factors used in many studies to distinguish Botnets from normal network activities. As observed by Kirubavathi *et al.* [25] Bots come with significant uniformity in their communication pattern and packet size. On the other hands the normal users generate packets with high diversity in size. Figure 3 depicts the packet sizes collected from the Neris Bot in comparison with the packet sized generated by a normal user.

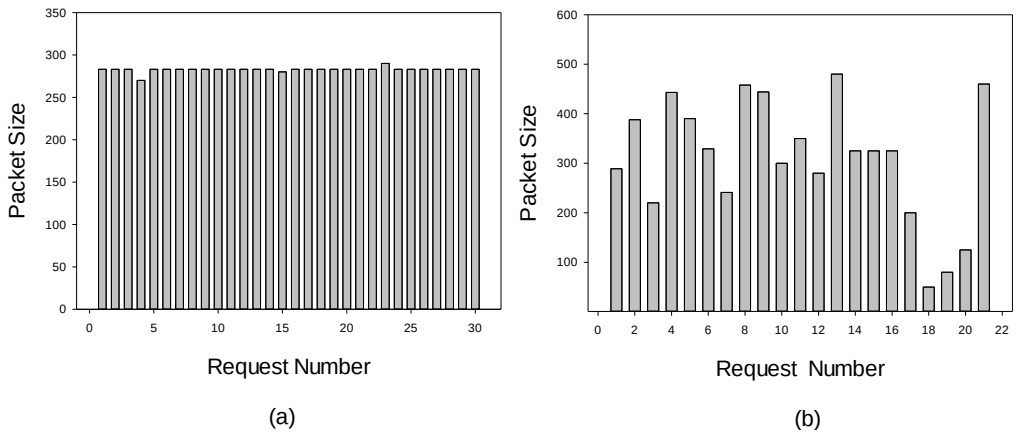

Figure 3. (a) Geinimi Botnet (b) Normal Application Packets Size

As discussed earlier, considering the actual packet size as a factor is less efficient in comparison to identifying the pattern of packet size changes, uniformity and density. The standard deviation shown in formula 3 is one of the effective measures used in this research to quantify the amount of variation of the generated packet sizes.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathrm{x}_i - \mu)^2} \tag{3}$$

For instance the standard deviation of the Geinimi Bot packet size depicted in Figure 3 is close to zero which indicates that the size of packets tend to be very similar (close to the mean of the entire set of packet sizes). On the other hands, the higher standard deviation indicates that packets were generated in a wider range of sizes that represent the normal user pattern.

3.3.5. Horizontal and Vertical Similarity Index (HSI, VSI): The cooperative analysis relies on the fact that the Bots belonging to the same Botnet come with similar characteristics and pose similar activities in compromised victims [19], therefore, analyzing cooperative patterns of Bots has become one of the effective detection approaches employed by many studies [1, 2]. Based on the above-mentioned concept, the vertical similarity index or VSI looks for any similarity of suspicious activities in the history of single mobile device. In contrast, the horizontal similarity index or HSI aims to find similarities amongst different mobile devices activities in a network. The Jaccard Index metric shown in formula 4 is employed by both vertical and horizontal indexes to measure the level of similarities in the history of single device along with similarities of group of mobile devices [26].

$$J(\mathrm{A},\mathrm{B}) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|\mathrm{A}| + |\mathrm{A}| - |A \cap B|} \tag{4}$$

In fact, each of the Jaccard indexes define a similarity metric in which the size of the intersection divided by the size of the union of the groups associated with two mobile devices or two different data archive from single mobile activities history.

3.3.6. Correlation Index: The correlation index or CI aims to determine how strongly the vertical and horizontal similarity indexes are linked together or related to each other (for instance the VSI and HSI increased together). This is a new metric which is designed to identify the connection between similarity of suspicious activities of a mobile client and other mobile devices in a network. The Pearson product correlation shown in formula 5 is selected to implement the proposed correlation index as it is commonly used in linear regression [27].

$$r(\mathrm{X},\mathrm{Y}) = \frac{\sum_{i=1}^{n}(x_i - \bar{X})(\mathrm{y}_i - \bar{Y})}{\sqrt{\sum_{j=1}^{n}(x_i - \bar{X})^2 \sum_{j=1}^{n}(y_i - \bar{Y})^2}} \tag{5}$$

Similar to the HIS and VSI explained in the previous section, the correlation index is a retrospective metric. Therefore the more network traffic is collected the more accurate index is generated. These features can effectively identify the similarity of Bot activities especially if they are applied on service provider's levels. The correlation index result is a number between 1 and -1 for Perfect Positive Correlation to perfect negative correlation [28].

3.4. Pattern Recognition Engine

Among various classification and pattern recognition techniques the neural networks and especially the feedforward neural networks which employ a backpropagation approach in their training are the most popular method [29]. In general, an artificial neural network (ANN) is an information–processing system consists of a set of nodes and links. The nodes are representing the neurons and the links between their attributes to the connection and the flow of data between the nodes [30]. These links are quantified by weights which should be tuned during the training phase. For the training purpose, a set of available instances are used which typically consist of a set of input features (called input vector) that are to be associated with a desired output vector [31]. Figure 4 depicts the feedforward neural network architecture.

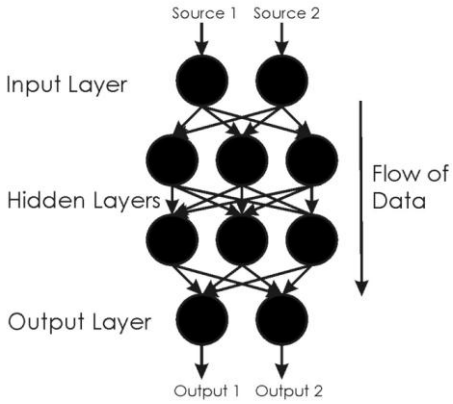

Figure 4. Feedforward Neural Network Architecture [30]

Considering that the input layer neurons has an input signal x_i, accordingly the hidden layer neurons j is defined as follows [32]:

$$input(\mathrm{j}) = \theta_{\mathrm{j}} + \sum_{i=1}^{n} x_i w_{ij} \tag{6}$$

Where w_{ij} is the weight between the input layer and the hidden layer, and θ_j is the bias in the hidden layer which is a threshold that has to be reached or exceeded for the neuron to produce an output signal. In addition an activation function is applied on each input to produced weighted signal [33]. Therefore, the outputs are formulated as follows:

$$O_k = \theta_k + \sum_{j=1}^{n} w_{jk} f(\mathrm{input(j)}) \tag{7}$$

Finally the backpropagation learning algorithm acts to update the weights of inputs following the rule below:

$$w(\mathrm{t+1}) = \mathrm{w(t)} - \eta(\mathrm{t}) \left[\sum \frac{\partial E(\mathrm{t})}{\partial w(\mathrm{t}} \right] + \alpha \Delta w(\mathrm{t}) \tag{8}$$

Where η is the learning rate, α is the momentum term ($0 < \alpha < 1$).

4. Testbed Architecture and Experimental Data Collection

Several experiments will be conducted to evaluate the proposed periodicity level classifier and mobile Botnet detection model. Figure 5 illustrates the experimental schema which is designed based on the topology proposed by many studies in the field of mobile malware detection.

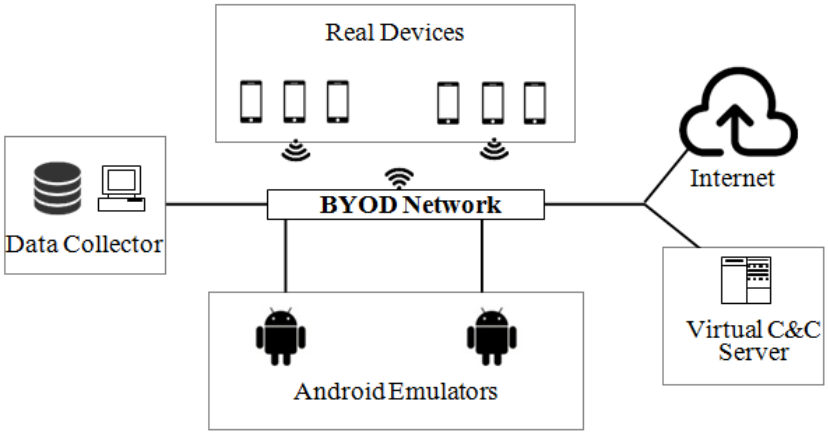

Figure 5. The Testbed Overview [10]

There are several mobile Bots from Genome [34] project and Drebin dataset [35] that are used to generate Bot data such as Geinimi, DroidKungFu, DroidDream, AnserverBot, CoinPirate, and Endofday. The mobile Bot data was captured in a few runs as more than one data set is needed to calculate the similarity metrics and their correlation index. In general six different datasets were created in a pacp file which contain Bot related packets, normal application and background packets as shown in Table 4.

Table 4. Collected Datasets

No	*Duration in Hours*	*No. Total Packets*	*No. Bot Packets*	*No. Clean Traffic*
1	6.20	668,251	10,176 (1.5%)	658075 (98.50%)
2	8.00	771,293	22,753 (2.95%)	748540 (97.05%)
3	8.00	746,190	12,163 (1.63%)	734027 (98.37%)
4	2.30	244,437	2,175 (0.89%)	242262 (99.11%)
5	4.00	477,440	12,127 (2.54%)	465313 (97.46%)
6	6.35	735,581	10,077 (1.37%)	725504 (98.63%)
Total:	**32.85**	**3,643,192**	**69,471 (1.90%)**	**3,573,721(98.10%)**

5. Experimental Result Analysis and Discussion

The core component of this model is the pattern recognition engines that employ Feedforward Backpropagation Neural networks algorithm as it is the most popular technique in data mining and pattern recognition [29]. Moreover the cooperative and retrospective approaches are designed based on the fact that Botnets are collaborated and organized cyber-crimes which consist of many Bots that pose the same Behaviour especially during the attacks [36]. The performance of the neural network ability in classifying the data is evaluated on the basis of two metrics such as percent error (E%) and mean squared error (MSE). The Mean squared error (MSE) is a common evaluation method that represents the average squared difference between outputs and targets. The lower value of MSE indicates better performance of the neural network. On the other hand, percent error (E%) is employed to represent the percent of misclassified instances. A 100 value means maximum misclassified instances while 0 represents a perfect classification [37].

The number of hidden nodes is arbitrary and it should be selected based on the problem at hand [30] . An extensive experiments conducted in this study indicates that the best performance is achieved using one hidden layer with 20 nodes. The overview of the neural network structure used in this research is shown in Figure 6.

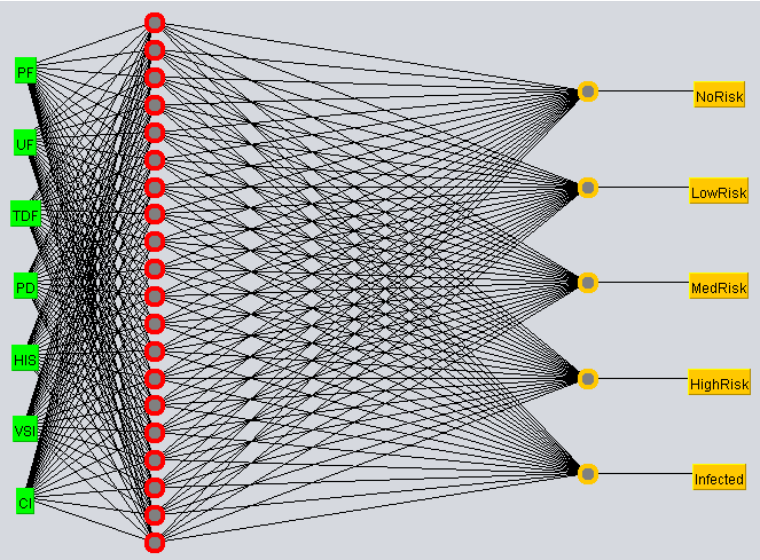

Figure 6. The Proposed Neural Network Structure

A set of 1000 instances are used for evaluating the proposed method. 70% of the instances are randomly selected for Training while 30% is used for Validation and Testing. Table 5 summarizes the performance of classifier for training and testing processes.

Table 5. Proposed Model Performance in Training and Testing

Process	*MSE*	*E%*
Training	0.007	1.66
Testing	0.012	3.27

In addition to the MSE and E%, a cross validation test is employed to evaluate the quality of dataset and accuracy of learning during the training stage [38]. The training test was conducted in 10 iterations where the certain amount of dataset instances was used in each of iterations. Figure 7 depicts the training learning curve for 10 iterations.

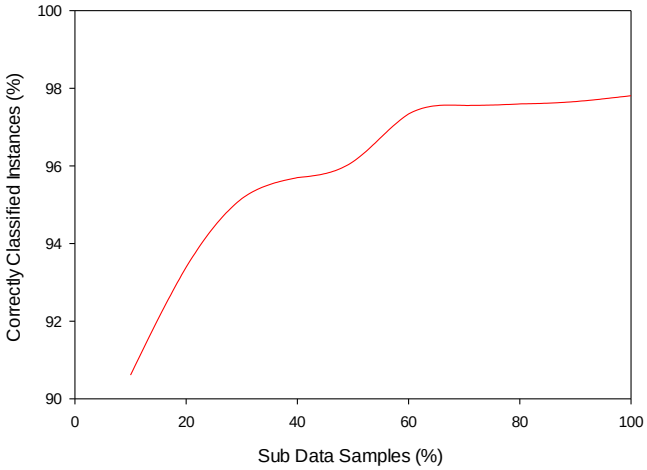

Figure 7. Training Learning Curve for 10 Iterations

As shown in the figure above, the proposed model was able to successfully classify the data with an accuracy of 97% from iteration number 6 with only 60% of instances. Table 6 shows the training test results in detail.

Table 6. Training Learning Accuracy in 10 Iterations

Iteration Number	*Percentage of Used Instances*	*Number of Instances*	*Correct Classified Data (%)*
1	10%	100	90.61%
2	20%	200	93.39%
3	30%	300	95.16%
4	40%	400	95.70%
5	50%	500	96.11%
6	60%	600	97.34%
7	70%	700	97.56%
8	80%	800	97.60%
9	90%	900	97.66%
10	100%	100	97.81%

As shown in the table above the accuracy of 97.81% was obtained in iteration 10 where all of the instances were involved in training. However, the training process obtained the similar range of accuracy from iteration 6 to iteration 10 (97.34% to 97.81%). This indicates that the model can be effectively trained by even having only 60 percent of total instances. Finally the datasets listed in table 4 are employed to evaluate the performance of proposed trained model in HTTP mobile Botnet detection. Figure 8 depicts the feedforward neural network results summary on HTTP Mobile Botnet detection.

```
=== Summary ===

Correctly Classified Instances         978               97.8    %
Incorrectly Classified Instances        22                2.2    %
Kappa statistic                          0.9725
Mean absolute error                      0.0176
Root mean squared error                  0.0934
Relative absolute error                  5.4984 %
Root relative squared error             23.3567 %
Total Number of Instances             1000
```

Figure 8. Proposed Model Results Summary

As shown in the figure above, more than 97% of data instances were correctly classified (978 out of 1000) with only 2% misclassified. The kappa statistic value of 97% (0.9725) significantly proves the efficiency of aforementioned classifier in HTTP mobile Botnet detection. Furthermore, the performance of the selected algorithm in Botnet detection is evaluated based on the number of metrics as depicted in Figure 9.

```
=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure   MCC      ROC Area  PRC Area  Class
                 1.000     0.000     1.000      1.000    1.000      1.000     1.000     1.000     NoRisk
                 0.957     0.000     1.000      0.957    0.978      0.973     0.985     0.980     LowRisk
                 0.970     0.009     0.965      0.970    0.967      0.959     0.988     0.949     MedRisk
                 0.965     0.008     0.970      0.965    0.968      0.959     0.990     0.923     HighRisk
                 1.000     0.011     0.955      1.000    0.977      0.972     0.995     0.964     Infected
Weighted Avg.    0.978     0.005     0.978      0.978    0.978      0.973     0.991     0.963
```

Figure 9. HTTP Mobile Botnet Detection Results

In average, the proposed model is able to detect the infected instances by a true positive rate of 97.8% with a very low rate of only 0.05% false positive. The main contribution of the proposed model is the ability to detect mobile Botnets with random and long term intervals. Moreover, the cooperative and retrospective approached significantly improve the performance and accuracy of the proposed model. Figure 10 depicts the ROC curve of the proposed model.

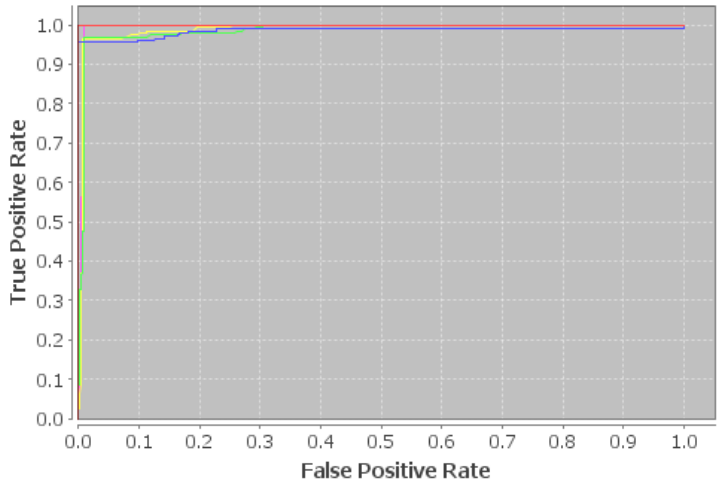

Figure 10. The ROC Curve for HTTP Mobile Botnet Detection

6. Conclusion

This paper proposed three metrics to classify the periodic behaviors posed by HTTP Botnets as they constantly communicate with their Botmaster to receive new commands. However the proposed periodic classifier shows less efficiency in the real world as there are several normal applications posing similar periodic behavior as well. Therefore, a cooperative behavior analysis approach was also proposed in which the similarities of the group activities were analyzed since the Bots belonging to Botnets have similar characteristics and conduct the same activities. Finally, a mathematical correlation method was employed to determine how strong the activities of a single infected device are linked with other potential infected devices in a network. The experimental results show that the combination of proposed periodicity and Neural Network pattern recognition is able to detect HTTP mobile Botnets with a high rate of detection and accuracy along with a low rate of false positive.

Acknowledgments

The authors would like to acknowledge the Ministry of Higher Education (MOHE) for providing the grant 600-RMI/FRGS 5/3 (141/2015) in carrying out this research work and to the Institute of Research Management and Innovation Universiti Teknologi MARA for their support.

References

[1] S. Silva, R. Silva, R. Pinto and R. Salles, "Botnets: A survey", *Computer Networks*, vol. 57, **(2013)**, pp. 378-403.
[2] M. Eslahi, R. Salleh and N. B. Anuar, "Bots and botnets: An overview of characteristics, detection and challenges", in *IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, **(2012)**, pp. 349-354.
[3] H. R. Zeidanloo and A. A. Manaf, "Botnet Command and Control Mechanisms", in *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, **(2009)**, pp. 564-568.

[4] G. Bottazzi and G. Me, "A Survey on Financial Botnets Threat", in *Global Security, Safety and Sustainability: Tomorrow's Challenges of Cyber Security*, ed: Springer, **(2015)**, pp. 172-181.
[5] U. Wijesinghe, U. Tupakula and V. Varadharajan, "An Enhanced Model for Network Flow Based Botnet Detection", in *Proceedings of the 38th Australasian Computer Science Conference (ACSC 2015)*, **(2015)**, p. 30.
[6] M. Eslahi, R. Salleh and N. B. Anuar, "MoBots: A new generation of botnets on mobile devices and networks", in *IEEE Symposium on Computer Applications and Industrial Electronics (ISCAIE)*, **(2012)**, pp. 262-266.
[7] M. Olalere, M. T. Abdullah, R. Mahmod and A. Abdullah, "A Review of Bring Your Own Device on Security Issues", *SAGE Open*, vol. 5, 2015-04-01 00:00:00, **(2015)**.
[8] M. Eslahi, M. V. Naseri, H. Hashim, N. M. Tahir and E. H. M. Saad, "BYOD: Current State and Security Challenges", presented at the IEEE Symposium on Computer Applications & Industrial Electronics, Peneng, Malaysia, **(2014)**.
[9] N. Zahadat, P. Blessner, T. Blackburn and B. A. Olson, "BYOD security engineering: A framework and its analysis", *Computers & Security*, vol. 55, **(2015)** November, pp. 81-99.
[10] M. Eslahi, M. R. Rostami, H. Hashim, N. M. Tahir and M. V. Naseri, "A data collection approach for Mobile Botnet analysis and detection", in *IEEE Symposium on Wireless Technology and Applications (ISWTA)*, **(2014)**, pp. 199-204.
[11] P. Farina, E. Cambiaso, G. Papaleo and M. Aiello, "Are mobile botnets a possible threat? The case of SlowBot Net", *Computers & Security*, vol. 58, **(2016)**, pp. 268-283.
[12] M. La Polla, F. Martinelli and D. Sgandurra, "A Survey on Security for Mobile Devices", *Communications Surveys & Tutorials, IEEE*, vol. PP, **(2012)**, pp. 1-26.
[13] I. Vural and H. Venter, "Mobile botnet detection using network forensics", presented at the Proceedings of the Third future internet conference on Future internet, Berlin, Germany, **(2010)**.
[14] X. Su, M. C. Chuah and G. Tan, "Smartphone dual defense protection framework: Detecting malicious applications in android markets", in *Eighth International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, **(2012)**, pp. 153-160.
[15] C. Byungha, C. Sung-Kyo and C. Kyungsan, "Detection of Mobile Botnet Using VPN", in *Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, **(2013)**, pp. 142-148.
[16] A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina, R. u. R. Ma'arof and S. Shamshirband, "A study of machine learning classifiers for anomaly-based mobile botnet detection", *Malaysian Journal of Computer Science*, vol. 26, **(2014)**.
[17] A. J. Alzahrani and A. A. Ghorbani, "SMS mobile botnet detection using a multi-agent system: research in progress", presented at the Proceedings of the 1st International Workshop on Agents and CyberSecurity, Paris, France, **(2014)**.
[18] E. Johnson and I. Traore, "SMS Botnet Detection for Android Devices through Intent Capture and Modeling", in *Reliable Distributed Systems Workshop (SRDSW), 2015 IEEE 34th Symposium on*, **(2015)**, pp. 36-41.
[19] G. Gu, R. Perdisci, J. Zhang and W. Lee, "BotMiner: clustering analysis of network traffic for protocol- and structure-independent botnet detection", in *17th conference on Security symposium*, San Jose, CA, **(2008)**, pp. 139-154.
[20] M. Eslahi, M. Rohmad, H. Nilsaz, M. Var Naseri, N. Tahir and H. Hashim, "Periodicity classification of HTTP traffic to detect HTTP Botnets", in *Computer Applications & Industrial Electronics (ISCAIE), 2015 IEEE Symposium on*, **(2015)**, pp. 119-123.
[21] S. Bojjagani and V. Sastry, "STAMBA: Security Testing for Android Mobile Banking Apps", in *Advances in Signal Processing and Intelligent Recognition Systems*, ed: Springer, **(2016)**, pp. 671-683.
[22] M. Chandramohan and H. Tan, "Detection of Mobile Malware in the Wild", *Computer*, vol. PP, **(2012)**, pp. 1-1.
[23] W. T. Strayer, R. Walsh, C. Livadas and D. Lapsley, "Detecting Botnets with Tight Command and Control", in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, **(2006)**, pp. 195-202.
[24] E. K. Morris, T. Caruso, F. Buscot, M. Fischer, C. Hancock and T. S. Maier, "Choosing and using diversity indices: insights for ecological applications from the German Biodiversity Exploratories", *Ecology and evolution*, vol. 4, **(2014)**, pp. 3514-3524.
[25] G. Kirubavathi and R. Anitha, "Botnet detection via mining of traffic flow characteristics", *Computers & Electrical Engineering*, vol. 50, **(2016)** Frebuary, pp. 91-101.
[26] S.-W. Hsiao, Y.-N. Chen, Y. S. Sun and M. C. Chen, "A cooperative botnet profiling and detection in virtualized environment," in *IEEE Conference on Communications and Network Security (CNS)*, **(2013)**, pp. 154-162.
[27] D. J. Weller-Fahy, B. J. Borghetti and A. A. Sodemann, "A Survey of Distance and Similarity Measures Used Within Network Intrusion Anomaly Detection", *IEEE Communications Surveys & Tutorials*, vol. 17, **(2015)**, pp. 70-91.
[28] F. Gravetter and L. Wallnau, *Essentials of Statistics for the Behavioral Science*: Cengage Learning, **(2007)**.

[29] M. Ahmed, A. N. Mahmood and J. Hu, "A survey of network anomaly detection techniques", *Journal of Network and Computer Applications*, vol. 60, **(2016)**, pp. 19-31.
[30] J. Schmidhuber, "Deep learning in neural networks: An overview", *Neural Networks*, vol. 61, **(2015)**, pp. 85-117.
[31] A. A. Freitas, *Data mining and knowledge discovery with evolutionary algorithms*: Springer Science & Business Media, **(2013)**.
[32] G. K. Venkatesh and R. A. Nadarajan, "HTTP botnet detection using adaptive learning rate multilayer feed-forward neural network", in *Information Security Theory and Practice. Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems*, ed: Springer, **(2012)**, pp. 38-48.
[33] N. Yadav, A. Yadav and M. Kumar, *An Introduction to Neural Network Methods for Differential Equations*: Springer Netherlands, **(2015)**.
[34] Y. Zhou and X. Jiang. **(2012)**. *Android Malware Genome Project*. Available: http://www.malgenomeproject.org/
[35] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon and K. Rieck, "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket", in *NDSS*, **(2014)**.
[36] M. Yahyazadeh and M. Abadi, "BotOnus: An online unsupervised method for botnet detection", *The ISC International Journal of Information Security*, vol. 4, **(2015)**.
[37] K. Diamantaras, W. Duch and L. S. Iliadis, *Artificial Neural Networks - ICANN 2010: 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings*: Springer, **(2010)**.
[38] I. H. Witten, E. Frank and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*: Elsevier Science, **(2011)**.

Authors

Meisam Eslahi is an information security researcher and digital forensic investigator, received his Masters' of Computer Science in Network Security filed. He is working toward the Ph.D. degree in Computer Engineering at UiTM, Malaysia and his domain of interests include Cyber security Threats Detection, Mitigation and Response (Mobile Botnets in Particular), Behavioral Analysis, Cyber safety and Digital Awareness. He has over 14 years of experience in the field of Information Technology mostly focused on Cyber Security, digital forensics, penetration testing and incident response and actively contributes in many projects, consultancies, developments, practical workshops and professional training programs.

Moslem Yousefi is currently working as a senior lecturer in the department of mechanical engineering, Universiti Tenga Nasional (UNITEN), Malaysia. He is presently associated with the center for advanced mechatronics and robotics (CAMARO) and Centre for Systems and Machines Intelligence at UNITEN. He is also a member of the science and engineering institute (SCIEI) and etc. Dr.Moslem did his PhD at Universiti Teknology Malaysia (UTM) (2010-2013) where he worked on developing a new evolutionary-based design approach for constrained optimization of thermal systems. Dr. Moslem was awarded the prestigious "best researcher award" from Islamic Azad Universtiy, Iran (2013) for his outstanding contribution to the development of the society.

Maryam Var Naseri received the Bachelor degree in Information System Security from Asia Pacific University of Technology and Innovation, Malaysia, in 2015. She is working toward the Master degree in Information Assurance at Universiti Teknologi Malaysia (UTM), and her domain of interests include Cyber security Threats Detection, Mitigation and Response Behavioral Analysis, Cyber safety and Digital Awareness.

Yusnani Mohd Yussoff PhD (UiTM) is a lecturer at the Faculty of Electrical Engineering, UiTM. She has more than 15 years' experience in the academia and industry. She has taught several courses including Digital System Fundamental, Data Network and Advance Data Network for Master students. Her research focuses on Lightweight Cryptography, Embedded Trusted Platform, Wireless Sensor Network and Secure e-health environment. She's currently supervising four PhD students and one master student. Her passion is on research and teaching. She is currently working on the development of new curriculum structure for 21st century engineering students. Besides research papers, she has published a Computer Organization and Architecture Fundamentals book, a custom publication by Pearson.

Nooritawati Md Tahir (PhD, CEng) is currently the Director of Research Innovation Business Unit, (RIBU) Universiti Teknologi MARA (UiTM) since Jan 2014 with her key responsibilities include planning, implementing and governing RIBU as a hub in intensifying and empowering activities of innovation and creativity as well as managing university's intellectual property (IP). She is also the founder of UiTM Innovation Sdn Bhd (UISB), a business arm for commercialization of UiTM's technology/product. She has received more than RM1.85M research grant for the last three years as principal investigator as well as collaborators. Five PhD and fifteen Masters Students have graduated under her supervision. Her research interest is in the field of Image Processing, Pattern Recognition, and Computational Intelligence. She has authored and co-authors more than 100 indexed publication too.

Habibah Hashim received her BSc. in Electrical and Electronics Engineering from Nottingham University in 1983. In 1986, she obtained her MSc. in Computer Aided Engineering from Teesside University and joined Universiti Teknologi MARA (formerly known as Institut Teknologi MARA). She received her PhD in Information and Communication Technology from Universiti Tenaga Nasional in 2007. She is an Associate Professor in the Faculty of Electrical Engineering and has served as the Deputy Dean of Research and Industrial Linkages between 2011 to 2014. Currently she is heading the Information Security and Trusted Information Laboratory and is pursuing her research interests in Wireless and Mobile Networks, Data Communications, Secure and Trusted Systems, Internet Of Things, Cloud Computing and E-Health Systems.

