A real-time botnet detection model based on an efficient wrapper feature selection method

Akram Farahmand-Nejad* and Samira Noferesti

Information Technology Department,
Faculty of Electrical and Computer Engineering,
University of Sistan and Baluchestan,
Zahedan, Iran
Email: akram.farahmand25@gmail.com
Email: snoferesti@ece.usb.ac.ir
*Corresponding author

Abstract: Botnets are one of the most widespread and serious threats of cybersecurity that have infected millions of computers around the world over the past few years. Previous research has shown that machine learning methods can accurately detect botnet attacks. However, these methods often do not address the problem of real-time botnet detection, which is one of the main challenges in this area and is essential to prevent the damage caused by botnet attacks. This paper aims to present an efficient real-time model for botnet detection. In the proposed method, a subset of the effective features in detecting the bot traffic is initially selected using the world competitive contests algorithm. Then, based on the selected features, a support vector machine model is created offline to detect real-time bot traffic from the normal one. The test results show that the proposed method can detect botnets with 95% accuracy and outperforms other methods.

Keywords: network security; botnets; real-time; machine learning; support vector machine; SVM; feature selection; world competitive contests algorithm; WCC; wrapper methods; botnet attacks.

Reference to this paper should be made as follows: Farahmand-Nejad, A. and Noferesti, S. (2020) 'A real-time botnet detection model based on an efficient wrapper feature selection method', *Int. J. Security and Networks*, Vol. 15, No. 1, pp.36–45.

Biographical notes: Akram Farahmand-Nejad is a MS student at the University of Sistan and Baluchestan. She received her BS on Computer Engineering. She is studying computer networks since 2017. Her interests include network security and artificial intelligence.

Samira Noferesti received her BS and MS both on Computer Software Engineering from the Sharif University of Technology and AmirKabir University of Technology, Tehran, Iran, respectively. She received her PhD in Computer Engineering-Software Engineering from the Shahid Beheshti University in 2015. He has been an Assistant Professor at the University of Sistan and Baluchestan from 2015. She is the Head of Information Technology Department in Electrical and Computer Engineering Faculty. Her main fields of interest are natural language processing, opinion mining and artificial intelligence.

1 Introduction

One of the most widespread and serious security issues in the current networks is the botnet attacks. In this series of attacks, the attacker can install malware remotely on victim systems, through which attacks such as spamming, phishing and denial of service (DOS) are done (Schiller and Binkley, 2011). Botnet attacks have infected millions of computers around the world in some ways over the past few years and have interfered with the normal service process of the virtual space.

The botnet is a network of compromised computers (bots) which are controlled by attackers. The controller of a botnet is known as the botmaster or botherder, and its task is configuring and controlling infected computers remotely through a command and control (C&C) channel. The C&C structure includes the way a botnet receives commands, and updates its functionality, how it transmits data, and how communication is handled between the botmaster and the bots (Schiller and Binkley, 2011).

There are three different types of botnet architectures: centralised, distributed and hybrid (Silva et al., 2013). In a centralised architecture, bots connect to one or more C&C servers, and each bot can receive command messages from the C&C server directly. This architecture is easy to design and manage but suffers from a significant drawback. It is highly vulnerable, and when the C&C server was detected and suppressed, the whole botnet would collapse. Thus, the C&C server is its vulnerable point (Silva et al., 2013).

In a distributed architecture, bots are interacting in a peer-to-peer network, and each node simultaneously acts as

C&C server and bot. In this type of architecture, the whole network cannot be broken by detecting a bot. Distributed architecture is more flexible, robust and harder to detect since bots do not establish communication with one point. However, this architecture is not easy to control and manage (Limarunothai and Munlin, 2015).

Hybrid architectures combine the advantages of centralised and distributed structures to make them more resistant to discover. This architecture has a good ability to recover from a failure, but its implementation is difficult (Limarunothai and Munlin, 2015).

Several studies have been carried out in this regard due to the importance of the problem of botnet detection and its effective role in providing cybersecurity. Due to the involvement of large amount of data, detection of a botnet using machine learning algorithms is in huge trend. Previous research has shown that machine learning methods can accurately detect botnet attacks. However, in most existing methods, the problem of the real-time detection of the botnet has not been paid much attention. Real-time detection of botnets is one of the main challenges in this area and is necessary to prevent damage caused by botnet attacks.

The purpose of this paper is to present a real-time model for detection of bot traffic from the normal one. For this purpose, a new method in the selection of effective features is proposed to detect bot traffic based on the world competitive contests (WCC) algorithm. WCC is an intelligent optimisation algorithm introduced in 2016 (Masoudi-Sobhanzadeh and Motieghader, 2016). Despite the successful results of the WCC in many biological and non-biological applications (Masoudi-Sobhanzadeh and Motieghader, 2016), this algorithm has not yet been used for the feature selection. After determining the effective features in detecting bot traffic from normal traffic using the WCC, a support vector machine (SVM) classifier is trained for this subset of features. In the end, the SVM model is used to detect real-time botnet attacks. With the above descriptions, in brief, the innovations of the article are as follows:

1 Proposing a new method for selection of the effective features in detecting botnets based on WCC.

2 Presenting a real-time model with high accuracy and precision for the detection of botnets.

The accuracy and runtime of the proposed method have been compared with the algorithms of particle swarm optimisation (PSO), ant colony optimisation (ACO) and genetic algorithm (GA) to evaluate it for the feature selection. Also, the efficiency of the model made to detect botnets has been compared with commonly used machine learning algorithms for botnet detection. The results of the tests performed on the University of California dataset (Meidan et al., 2018) show that the proposed method has achieved a higher accuracy and less error than the algorithms mentioned above. The proposed method can also detect bot traffic in less time.

The rest of this paper is organised into five sections: the second section reviews and categorises previous related research and explains the advantages and disadvantages of each category. In the third section, details of the proposed method for detection of botnet attacks are presented. Then, in the fourth section, the proposed method and the results are evaluated. The final section is the conclusion.

2 Related work

In general, techniques for botnet detection can be divided into two categories based on honeynet and intrusion detection systems (IDS). IDS include five groups of signature-based detection, anomaly-based detection, the domain name system (DNS), machine learning, and hybrid methods, which are briefly described in the following.

The honey net-based method plays the role of an infected machine to stimulate the attacker and records all communications and actions between them. This technique is used to collect data related to bots (Alauthman, 2016).

Signature based-intrusion detection method includes signatures and patterns of known botnets that can report whether the traffic is a bot or not by comparing current traffic passing through the network with these signatures (Alder et al., 2007; Goebel and Holz, 2007).

Anomaly-based IDS analyses abnormal traffic in the network and, normal network traffic is separated from an unusual one based on this analysis (Gu et al., 2008; Han et al., 2012; Stinson and Mitchell, 2007).

Machine learning-based IDS explores hidden relationships within the data that can be used to create models for prediction and classification (Bijalwan et al., 2016; Bilge et al., 2012; Chen et al., 2017; Khanchi et al., 2018; Livadas et al., 2006; Masud et al., 2008; Selvam and Sumathi, 2015; Saad et al., 2011; Shin et al., 2012; Zhao et al., 2013).

DNS-based IDS is similar to anomaly-based IDS, except that in these methods, anomalies in DNS traffic are checked (Choi et al., 2009).

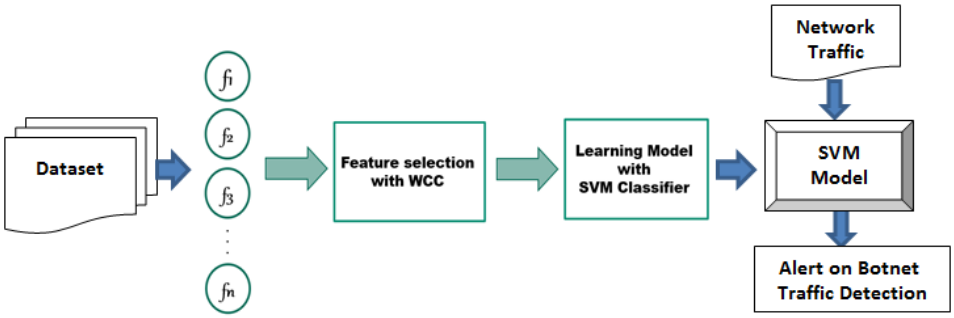
In hybrid methods, two or more botnet detection approaches are combined aiming to cover the weaknesses of one another (Shin et al., 2012).

Table 1 compares previous research on botnet detection based on different criteria. Among the methods mentioned above, only three methods (Bilge et al., 2012; Selvam and Sumathi, 2015; Saad et al., 2011) have all the capabilities, including feature selection and real-time. Some of the previous works (Bilge et al., 2012; Selvam and Sumathi, 2015) have used heuristic methods to select features. Since heuristic methods may be captured in local optimisations, it is preferable to use metaheuristic algorithms for feature selection. Most of the work done has not considered feature selection and model creation efficiently and faces limitations in the real-time detection of botnets. In this paper, an efficient method for feature selection is suggested, resulting in a real-time model with more precision, accuracy, and sensitivity.

Table 1 The work performed to botnet detection

Method	*Category*				*Criteria*					
	Signature-based	*Anomaly-based*	*DNS-based*	*Machine learning*	*Detecting unknown bot*	*Independent from structures and protocols*	*Detecting encrypted bots*	*FPR low*	*Real-time*	*Feature selection*
Alder et al. (2007)	✓				×	×	×	×	×	×
Goebel and Holz (2007)	✓				×	×	×	×	×	×
Han et al. (2012)		✓			✓	✓	✓	✓	✓	×
Stinson and Mitchell (2007)		✓			✓	×	✓	✓	×	×
Gu et al. (2008)		✓			✓	×	✓	✓	×	×
Choi et al. (2009)			✓		✓	×	×	×	✓	×
Shin et al. (2012)		✓		✓	✓	✓	✓	✓	×	×
Masud et al. (2008)				✓	✓	✓	✓	✓	×	×
Livadas et al. (2006)				✓	✓	×	×	×	×	✓
Bilge et al. (2012)				✓	✓	✓	✓	✓	✓	✓
Saad et al. (2011)				✓	✓	✓	✓	✓	✓	✓
Zhao et al. (2013)				✓	✓	✓	✓	✓	×	✓
Chen et al. (2017)				✓	✓	✓	✓	✓	×	×
Khanchi et al. (2018)				✓	✓	✓	✓	✓	✓	×
Selvam and Sumathi (2015)				✓	✓	✓	✓	✓	✓	✓
Bijalwan et al. (2016)				✓	✓	✓	✓	✓	×	×

Figure 1 Real-time botnet detection system (see online version for colours)



3 The proposed method

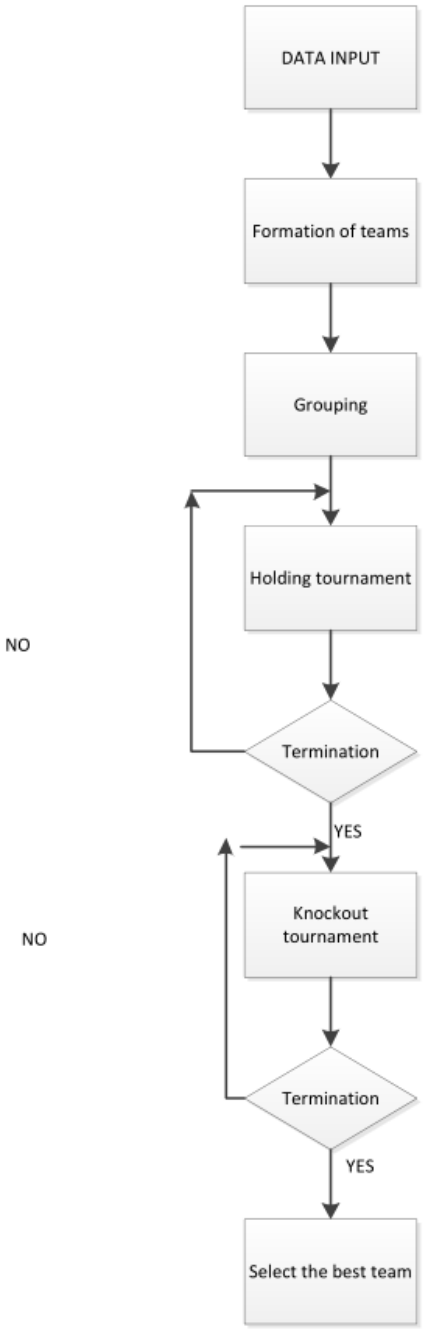
The proposed method aims to detect botnets relying on the identification of effective features. Selection of effective features and removal of unnecessary or low-effective features, in addition to precision, improves the speed of learning algorithm. In the proposed method, a subset of appropriate features is selected using the WCC algorithm, which is presented in Subsection 3.1. The reason for selection of this algorithm is its suitable convergence, stability and appropriate elapsed time compared with other optimisation algorithms such as ACO, PSO, and GA. After selecting the features, their efficiencies in detecting the botnets are evaluated based on the model obtained through the SVM classifier. SVM is trained based on the features selected by the WCC algorithm, and the accuracy of the proposed model is used to evaluate the effectiveness of the selected features. In the end, based on the set of selected features, a real-time model is created to detect botnets. Figure 1 shows the diagram of the proposed system.

3.1 WCC algorithm for feature selection

The WCC algorithm was presented in 2016 (Masoudi-Sobhanzadeh and Motieghader, 2016). This algorithm has been inspired by the rules of the sport in the world championships. The WCC algorithm starts by randomly creating an initial population of teams. Each team consisting of some players represents a solution to the problem. Each player in the feature selection problem is equivalent to a feature, and each team is considered to be a set of features. The teams are randomly grouped. Grouping

is done in such a way that the number of teams in each group is equal. The teams in each group compete periodically (local optimisation phase), and ultimately, highly qualified teams will reach the elimination stage (global optimisation phase) and compete there. The losing teams are set aside in the elimination tournament. At the end of the elimination stage, one team will remain, and it is selected as the champion of the tournament or the final solution to the problem. Competency of teams is measured by a predetermined function called the scoring function, as described in Subsection 3.2.

Figure 2 Flowchart of WCC algorithm for feature



The WCC algorithm includes four operators of shooting, attacking, passing and crossing. Each operator is described below (Masoudi-Sobhanzadeh and Motieghader, 2016):

1 Shooting: In this operator, the current team selects some of its values (features) randomly and sends them to the opposite team. The opposing team recalls the scoring function by receiving new values, and if these values improve the scoring function, they will be accepted; otherwise they will be ignored.

2 Attacking: In this operator, a series of values are generated randomly and sent to the opposite team. In other words, some features that are not currently in the team are randomly selected from the entire set of features and they are added to the opposing team. If these new values improve the scoring function, they will be accepted. Otherwise they will be ignored.

3 Passing: In this operator, two points are randomly selected, and their values are displaced. New values will be accepted provided that they improve the scoring function.

4 Crossing: This operator replaces the values of a team in a permutation form. As with the operators mentioned, new values will be accepted in this operator as long as the scoring function is improved.

In the WCC algorithm, competition is based on the duration of the game. The duration of a game is the number of team attempts to perform the operators mentioned above. It is also important to note that each existing operator is selected randomly. At the end of the game, the most qualified team will be selected as the winner of the game (the answer to the problem). The flowchart of the WCC algorithm for feature selection is shown in Figure 2.

3.2 Scoring function

In the WCC algorithm, there is also a referee who evaluates the teams' operations and gives them a score. The score is a value that indicates how much a developed answer is close to the optimal solution to the problem. A function called the scoring function is defined to calculate the score of a team depending on the problem. The scoring function plays the main role in converging and finding an optimal response in the optimisation algorithm. In the present study, the following steps are taken to assess the score of a team:

- The features selected by the team are given as inputs to SVM classifier.
- SVM is trained based on the input features on a labelled dataset which includes normal traffic and bot traffic.
- The model created on a test dataset detects the normal traffic from the bot traffic.
- The accuracy of the botnet detection on the test dataset is considered as the team's score. Accuracy is the percentage of samples in the test dataset that has been truly classified by the model.

Figure 3 illustrates the pseudo code of the proposed method for botnet detection.

Figure 3 Pseudo-code of the proposed method for detection of the botnets

```
Random generation of initial teams
Evaluate the score for each team // Calculate accuracy value
Random grouping of initial teams
For i = 1 to size (teams) do
    For j = 1 to size (teams) do
        If teams (i) is rival teams (j)
            First ← teams (i)
            Second ← teams (j)
            For t = 1 to matchtime do
                copy second team's values
                First ← apply roles (shooting, attacking, passing, crossing) on copied values
                If score (first) is improved then
                    teams (i) ← update current values with ameliorated values
                End
            End
        End
    End
End
While size (teams) > 1 do
        a = random number from 1 to size (teams)
        b = random number from 1 to size (teams)
        First ← teams (a)
        Second ← teams (b)
        For t = 1 to matchtime do
            copy seconds team's values
            First ← apply roles (shooting, attacking, passing, crossing) on copied values
            copy teams (a) team's values
            Second ← apply roles (shooting, attacking, passing, crossing) on copied values
            teams (a) ← first;
            teams (b) ← second;
            If score (first) > score (second) then
                teams (b) = []
            else
                teams (a) = []
            End if
        End // finish matchtime
End // finish while
Return teams
```

4 Evaluation results

In this section, the proposed method for botnet detection is evaluated and compared with other methods. The criteria of accuracy, precision, sensitivity, specificity, and false positive rate (FPR) were used to assess the proposed method. The confusion matrix is used to compute these parameters (Table 2).

Table 2 Confusion matrix

		Predicted	
		Positive	*Negative*
Actual	Positive	TP	FN
	Negative	FP	TN

Each of the matrix parameters is as follows:

- *TP:* The number of bot attacks detected truly by the proposed method.
- *TN:* The number of normal traffic accurately identified as normal.
- *FP:* The number of normal traffic incorrectly classified as the bot traffic.
- *FN:* The number of bot attacks detected falsely by the proposed method as the normal ones.

Concerning the confusion matrix parameters, the evaluation criteria are defined as follows:

1 Accuracy: It is the most popular and general criterion for evaluating botnet detection methods, which shows how much the algorithm has detected the total traffic truly. The accuracy criterion is obtained using equation (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

2 Precision: This criterion shows the proportion of correctly detected bot attacks out of the total number of traffic flows classified as a bot by a classifier and is calculated by relation (2).

$$precision = \frac{TP}{TP + FP} \tag{2}$$

3 Sensitivity: This criterion, also known as the recall, indicates correctly detected bot attacks and is calculated by relation (3).

$$sensitivity = \frac{TP}{TP + FN} \tag{3}$$

4 Specificity: This criterion states that how much the algorithm has detected the normal traffic that is not a member of the botnet, and is calculated with equation (4).

$$specificity = \frac{TN}{TN + FP} \tag{4}$$

5 Root mean square error (RMSE): It represents the error rate of the model and is calculated with equation (5). X

is the actual value and y is the estimated value of the model, and n is the total number of observed data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}} \quad (5)$$

6 FPR: This criterion shows that how much the algorithm has identified normal traffic as a bot, and is calculated with equation (6).

$$FPR = \frac{FP}{TN + FP} \quad (6)$$

In the following, the datasets used to evaluate the proposed method are described first and then the results of the tests are presented. The proposed method was implemented in MATLAB 2017. The specifications of the system in which the proposed method was run include 256 GB RAM, 500 cores with 2.2 GH power, and 8 GB of the internal memory and 12 cores were allocated to this program.

4.1 Dataset

In this paper, the dataset of the University of California (https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT), one of the newest and most complete datasets available for the detection of botnets, has been used. This dataset is a collection of real traffic data, gathered from nine internet of things (IoT) devices and includes one million samples, each of which has 115 features. Attacks on this data set are in the following three categories:

- Benign attacks: This group of attacks is benign because they do not incur a lot of damage to the network.
- Mirai attacks: This type of attack is a malware that propagates itself in the network and misuses the net for various purposes.
- Gafgyt attacks: These are attacks from Trojan horse viruses and are used more to steal information.

The specification of the used dataset is shown in Table 3.

Table 3 Specification of the dataset

Dataset characteristics	Multivariate, sequential
Attribute characteristics	Real
Associated tasks	Classification, clustering
Number of instances	1,000,000
Number of attributes	115
Missing values	N/A

4.2 Results

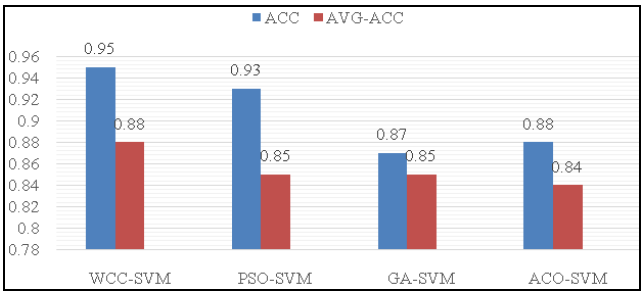
In Table 4, the efficiency of the proposed method has been compared with the GA, PSO, and ACO algorithms based on the before mentioned criteria. As can be seen, the proposed method is superior to the above methods in all criteria.

Table 4 Comparison of the efficiency of the proposed method compared to other methods

AL_NAME	*Sensitivity*	*Specificity*	*Precision*	*FPR*
WCC-SVM	0.95	0.98	0.95	0.01
GA-SVM	0.87	0.94	0.89	0.05
PSO-SVM	0.93	0.95	0.94	0.04
ACO-SVM	0.88	0.91	0.91	0.08

In Figure 4, the best accuracy (ACC) and the average of accuracy (AVG-ACC) of the algorithms have been compared over 30 runs of algorithms.

Figure 4 Comparison of the best accuracy (ACC) and the average of accuracy (AVG-ACC) of the proposed method compared to other methods during 30 runs (see online version for colours)



The results of Table 4 are related to the best results from 30 different algorithm runs, and the results of Figure 4 are related to the comparison of the highest accuracy and average of accuracy in 30 algorithm runs. As can be seen, the WCC has led to better results than other algorithms.

Figure 5 shows the receiver operating characteristic (ROC) curve related to the above algorithms. Figure 5 consists of two parts of the ROC and the ROC space. The larger is the area under the curve of a classifier, the more optimal is the final efficiency of it. It is observed that the proposed method has yielded better results.

In the next experiment, different feature selection algorithms have been compared based on convergence, the average convergence, and stability. The results of this comparison are shown in Figure 6. The horizontal axis shows the number of iterations of the feature selection algorithm and the vertical axis indicates the accuracy and RMSE of the botnet detection based on the selected features.

Figures 6(a) and 6(d) show the convergence of algorithms regarding accuracy and RMSE. As can be seen, the results of the WCC Algorithm are better than that of the other three algorithms, and in the 40th iteration, the accuracy of 95% and the RMSE of less than 0.2 have been obtained. The results of the PSO are better than that of the genetic and ACO algorithms. Also, by comparing ACO and GAs, it can be stated that both algorithms have achieved the same result regarding RMSE criterion, but regarding accuracy, ACO performed better than the GA.

Figure 5 The ROC curve of the proposed method compared to that of other methods (see online version for colours)

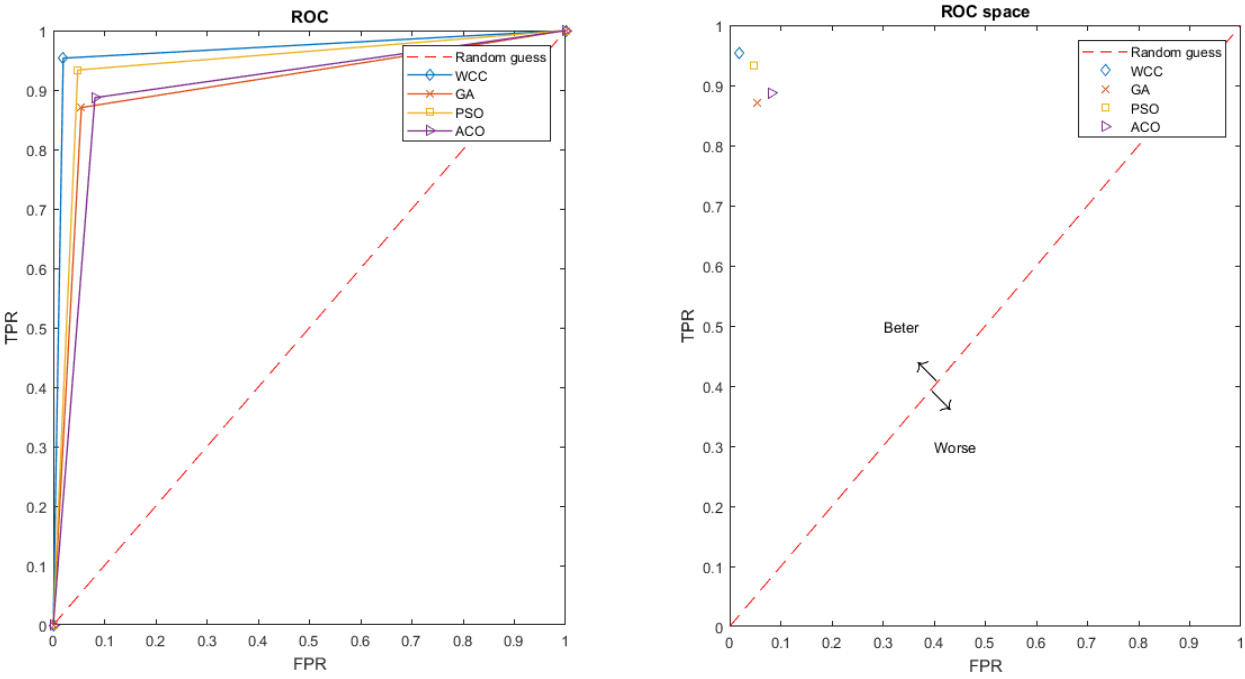
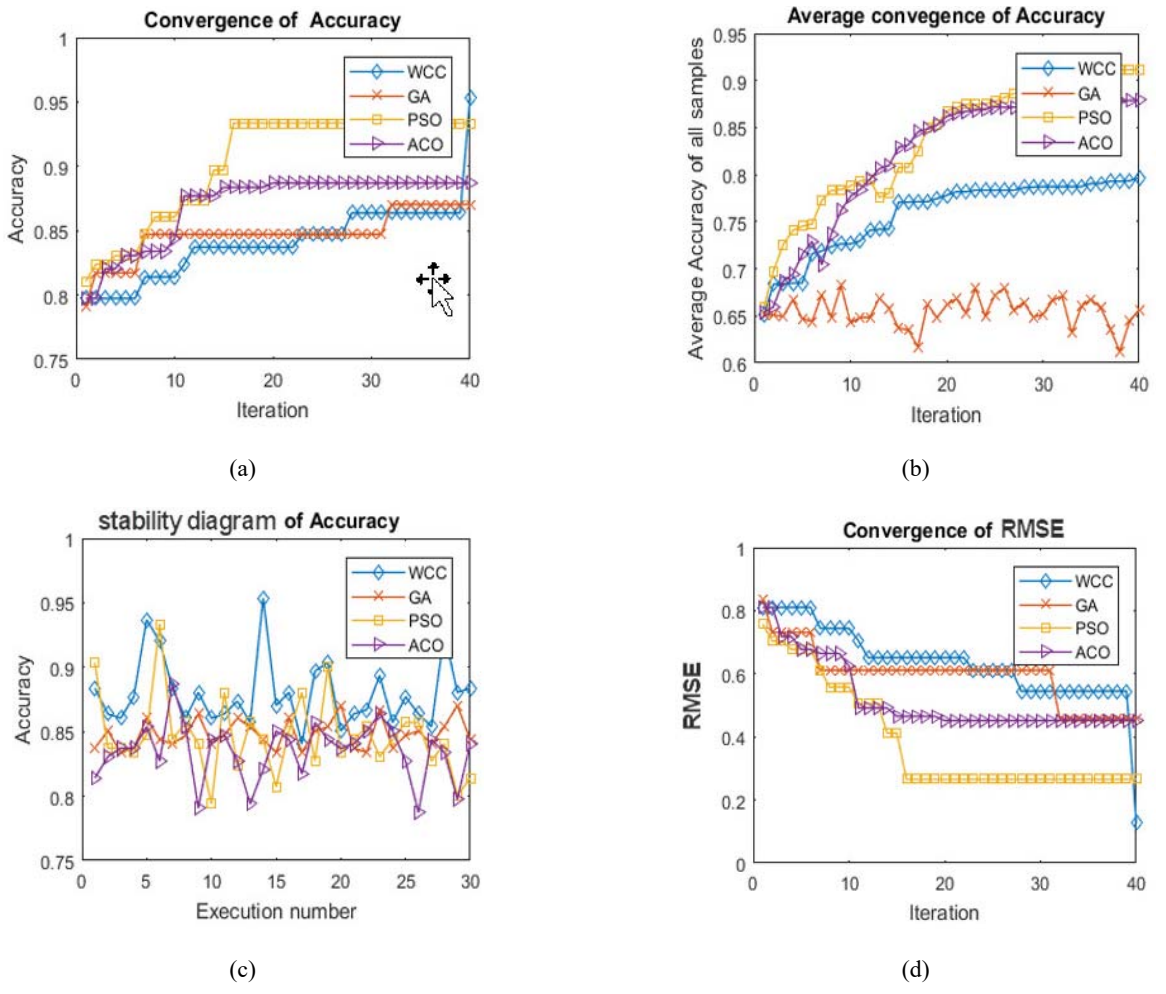


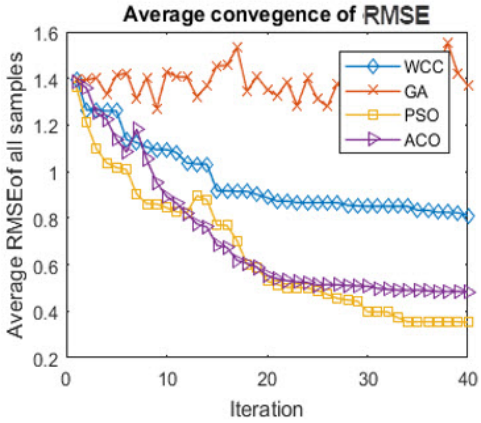
Figure 6 Convergence and stability of the proposed method compared to those of other methods (see online version for colours)



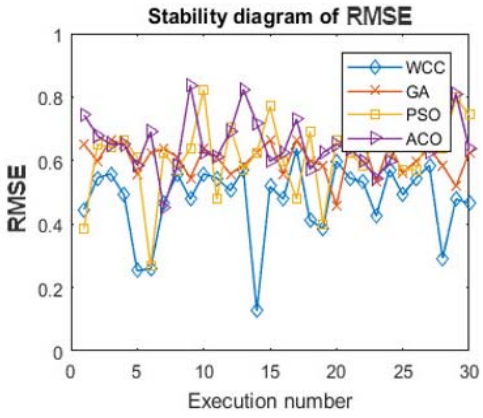
(a) (b)

(c) (d)

Figure 6 Convergence and stability of the proposed method compared to those of other methods (continued) (see online version for colours)



(e)



(f)

Figures 6(b) and 6(e) show the average convergence of the algorithms. As can be seen, ACO and PSO algorithms have converged more rapidly, but their best solution is less accurate than the best solution of the WCC algorithm.

Figures 6(c) and 6(f) are related to the stability of the algorithms that have been depicted over 30 different runs. Given that optimisation algorithms are randomly scored and their operators are also used randomly, different results are generated at each run of algorithms. The graphs of the first and second columns on the left side of Figure 6 relate to the best performance of the algorithms among 30 different runs. An algorithm that produces better results and also has fewer fluctuations is more appropriate and more efficient than other algorithms. As can be seen, the WCC algorithm has better results than three other algorithms.

Table 5 shows the average training time for 30 individual executions of feature selection algorithms in minutes. As seen in Table 5, the WCC algorithm has less training time than other algorithms. It should be noted that Table 5 illustrates the time it takes to build the model offline.

Table 5 Average training time of algorithms for model construction

AL_NAME	*WCC-SVM*	*PSO-SVM*	*GA-SVM*	*ACO-SVM*
Training time (min.)	91.09	145.41	214.90	222.89

Table 6 The number of features selected by algorithms

AL_NAME	*WCC-SVM*	*PSO-SVM*	*ACO-SVM*	*GA-SVM*
Number of selected features	13	25	28	29

Table 6 shows the number of selected features of the algorithms for the best response. Also, Table 7 shows the online time of algorithms for botnet detection in 7867 samples of data. As can be seen, the proposed method has chosen fewer features, resulting in less online time to intrusion detection.

Table 7 Runtime of algorithms to detect botnets

AL_NAME	*WCC-SVM*	*PSO-SVM*	*GA-SVM*	*ACO-SVM*
Runtime (sec.)	4.17	7.37	7.91	8.01

In the final experiment, the proposed method for detection of bot attacks has been compared with several machine learning algorithms commonly used for botnet detection including SVM, decision tree (DT), and neural network (NN) without feature selection. These three algorithms have been used successfully in previous studies to detect botnets (Alauthaman et al., 2018; Chowdhury et al., 2017; Kondo and Naoshi, 2007; Livadas et al., 2006).

For evaluation, 20% of the initial dataset (including 200,000 records) not used in the construction of the botnet detection model has been selected as a test dataset. The results of this comparison are shown in Figure 7. As shown in Figure 7, the proposed method is superior to other methods for both accuracy and RMSE.

Figure 7 Comparison of the proposed method with other methods regarding accuracy and RMSE (see online version for colours)

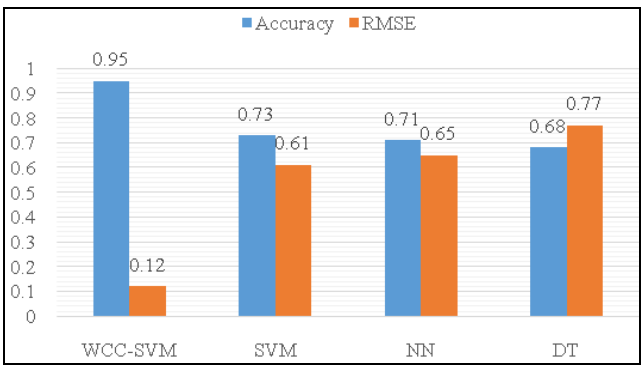
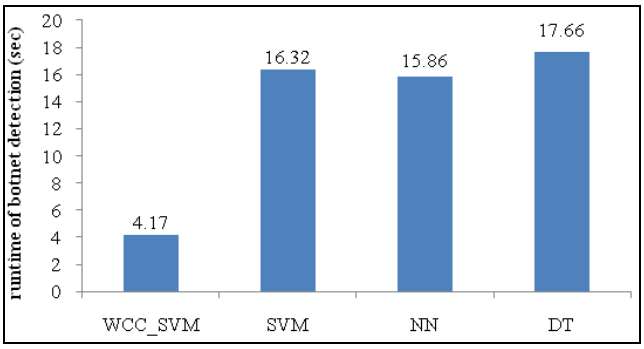


Figure 8 shows that the proposed method requires less runtime to detect the bot traffic compared with the three algorithms of SVM, NN, and DT.

In summary, the results of the evaluations show that the proposed method is more efficient than other methods of botnet detection, both regarding accuracy and runtime. Therefore, the constructed model is suitable for real-time prediction of botnets.

Figure 8 Comparison of the runtime of the proposed algorithm with that of other methods



5 Conclusions

In this paper, a novel method has been proposed for the real-time detection of botnets. The proposed method initially develops an efficient model regarding accuracy and runtime to detect the bot traffic from the normal one. To construct the model, effective features in detecting bot attacks are selected using the WCC algorithm, and, the botnet prediction model is created based on the selected features with the training of SVM classifier on a dataset. Then, this model is used for real-time detection of botnets.

Several criteria such as convergence, stability and classification criteria such as accuracy, precision, sensitivity, specificity, and FPR were used to evaluate the efficiency of the proposed method. The results of the evaluations showed that the feature selection based on the WCC algorithm compared with other common methods of feature selection, such as PSO, ant colony optimisation, and GA, increased the accuracy and reduced the runtime of the model made to detect botnets. Also, the proposed method was more efficient in detecting botnets compared to common machine learning algorithms.

References

Alauthaman, M., Aslam, N., Zhang, L., Alasem, R. and Hossain, M.A. (2018) 'A P2P botnet detection scheme based on decision tree and adaptive multilayer neural networks', *Neural Computing and Applications*, Vol. 29, No. 11, pp.991–1004.

Alauthman, M. (2016) *An Efficient Approach to Online Bot Detection Based on A Reinforcement Learning Technique*, Doctoral dissertation, Northumbria University.

Alder, R., Burke, J., Keefer, C., Orebaugh, A., Pesce, L. and Seagren, E.S. (2007) 'Chapter 4 – introducing snort', *How to Cheat at Configuring Open Source Security Tools*, pp.181–212, Syngress, Burlington.

Bijalwan, A., Chand, N., Pilli, E.S. and Krishna, C.R. (2016) 'Botnet analysis using ensemble classifier', *Perspectives in Science*, Special issue on 'Engineering and Material Sciences', Vol. 8, pp.502–504.

Bilge, L., Balzarotti, D., Robertson, W., Kirda, E. and Kruegel, C. (2012) 'Disclosure: detecting botnet command and control servers through large-scale netflow analysis', *Proceedings of the 28th Annual Computer Security Applications Conference*, ACM, pp.129–138.

Chen, W., Luo, X. and Zincir-Heywood, A.N. (2017) 'Exploring a service-based normal behaviour profiling system for botnet detection', in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May, pp.947–952.

Choi, H., Lee, H. and Kim, H. (2009) 'BotGAD: detecting botnets by capturing group activities in network traffic', in *Proceedings of the Fourth International ICST Conference on Communication System Software and Middleware*, June, 2pp.

Chowdhury, S., Khanzadeh, M., Akula, R., Zhang, F., Zhang, S., Medal, H., Marufuzzaman, M. and Bian, L. (2017) 'Botnet detection using graph-based feature clustering', *Journal of Big Data*, Vol. 4, No. 1, 14pp.

Goebel, J. and Holz, T. (2007) 'Rishi: identify bot contaminated hosts by IRC nickname evaluation', *First Workshop on Hot Topics in Understanding Botnets 2007*, 8pp.

Gu, G., Zhang, J. and Lee, W. (2008) 'BotSniffer: detecting botnet command and control channels in network traffic', in *15th Annual Network & Distributed System Security Symposium*, San Diego.

Han, F., Chen, Z., Xu, H., Wang, H. and Liang, Y. (2012) 'A collaborative botnets suppression system based on overlay network', *International Journal of Security and Networks*, Vol. 7, No. 4, pp.211–219.

Khanchi, S., Vahdat, A., Heywood, M.I. and Zincir-Heywood, A.N. (2018) 'On botnet detection with genetic programming under streaming data label budgets and class imbalance', *Swarm and Evolutionary Computation*, Vol. 39, pp.123–140.

Kondo, S. and Sato, N. (2007) 'Botnet traffic detection techniques by C&C session classification using SVM', in *International Workshop on Security*, October, Springer, Berlin, Heidelberg, pp.91–104.

Limarunothai, R. and Munlin, M. (2015) 'Trends and challenges of botnet architectures and detection techniques', *Journal of Information Science & Technology*, Vol. 5, No. 1, pp.51–57.

Livadas, C., Walsh, R., Lapsley, D.E. and Strayer, W.T. (2006) 'Using machine learning techniques to identify botnet traffic', in *Proceedings 2006 31st IEEE Conference on Local Computer Networks*, November, pp.967–974.

Masoudi-Sobhanzadeh, Y. and Motieghader, H. (2016) 'World competitive contests (WCC) algorithm: a novel intelligent optimization algorithm for biological and non-biological problems', *Informatics in Medicine Unlocked*, Vol. 3, pp.15–28.

Masud, M.M., Al-Khateeb, T., Khan, L., Thuraisingham, B. and Hamlen, K.W. (2008) 'Flow-based identification of botnet traffic by mining multiple log files', in *2008 First International Conference on Distributed Framework and Applications*, IEEE, pp.200–206.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D. and Elovici, Y. (2018) 'N-BaIoT – network-based detection of IoT botnet attacks using deep autoencoders', *IEEE Pervasive Computing*, Vol. 17, No. 3, pp.12–22.

Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., Felix, J. and Hakimian, P. (2011) 'Detecting P2P botnets through network behavior analysis and machine learning', in *2011 Ninth Annual International Conference on Privacy, Security and Trust*, IEEE, pp.174–180.

Schiller, C. and Binkley, J.R. (2011) *Botnets: The Killer Web Applications*, Elsevier, Syngress Publishing, Rockland, MA.

Selvam, D.V.V.N. and Sumathi, M.V. (2015) 'General framework for detection of botnet using random forest in real time', *International Journal of Scientific & Engineering Research*, Vol. 6, No. 4, pp.469–474.

Shin, S., Xu, Z. and Gu, G. (2012) 'EFFORT: rfficient and effective bot malware detection', in *2012 Proceedings IEEE INFOCOM*, IEEE, pp.2846–2850.

Silva, S.S., Silva, R.M., Pinto, R.C. and Salles, R.M. (2013) 'Botnets: a survey', *Computer Networks*, Vol. 57, No. 2, pp.378–403.

Stinson, E. and Mitchell, J.C. (2007) 'Characterizing bots' remote control behavior', in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, Springer, Berlin, Heidelberg, pp.89–108.

Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A. and Garant, D. (2013) 'Botnet detection based on traffic behavior analysis and flow intervals', *Computers & Security, Special issue: 27th IFIP International Information Security Conference*, Vol. 39, pp.2–16.