Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 986428, 9 pages
http://dx.doi.org/10.1155/2014/986428

Research Article

Botnet Detection Using Support Vector Machines with Artificial Fish Swarm Algorithm

Kuan-Cheng Lin,[1] Sih-Yang Chen,[1] and Jason C. Hung[2]

[1] *Department of Management Information Systems, National Chung Hsing University, Taichung 40227, Taiwan*
[2] *Department of Information Management, Overseas Chinese University, Taichung 40721, Taiwan*

Correspondence should be addressed to Kuan-Cheng Lin; kuanchenglin@gmail.com

Received 21 January 2014; Accepted 4 March 2014; Published 29 April 2014

Academic Editor: Young-Sik Jeong

Because of the advances in Internet technology, the applications of the Internet of Things have become a crucial topic. The number of mobile devices used globally substantially increases daily; therefore, information security concerns are increasingly vital. The botnet virus is a major threat to both personal computers and mobile devices; therefore, a method of botnet feature characterization is proposed in this study. The proposed method is a classified model in which an artificial fish swarm algorithm and a support vector machine are combined. A LAN environment with several computers which has infected by the botnet virus was simulated for testing this model; the packet data of network flow was also collected. The proposed method was used to identify the critical features that determine the pattern of botnet. The experimental results indicated that the method can be used for identifying the essential botnet features and that the performance of the proposed method was superior to that of genetic algorithms.

1. Introduction

Because of the advancements and innovations in technology, the applications of the Internet of Things (IoT) [1] are rapidly growing, such as cloud computing [2] and smart phone applications. The IoT is not a new type of technology; it is the extension of existing technologies; for example, tens of thousands of smart phones are connected by Wi-Fi, 3G networks, or radio-frequency identification; therefore, using smartphones is a type of IoT, and the development of IoT will be a major trend in the future.

However, because of the recent information explosion, information security has become a crucial topic, even in relation to the IoT. Botnets [3–6] are a recent major threat; when a computer has been infected by a botnet virus, it still functions normally, but the attacker can control the infected computer to threaten the victim by achieving distributed denial of service (DDoS) [7], sending spam, engaging in phishing, or embezzling personal or company data. Botnets are typically composed of three components: a bot herder, a bot client, and a command and control server. The bot herder is the attacker and the bot client is the victim that is infected by the botnet virus; the command and control server (C & C) is the control server of a botnet and also a communication tool between a bot herder and a bot client. A bot herder typically uses Internet Relay Chat (IRC) protocol to communicate with the command and control server and a bot client. IRC protocol provides real-time one-on-one or group chat room service through a connection to an IRC server, and every chat room is called a channel. A bot herder uses IRC channels to send specific command codes, which are already determined by the bot herder who sent the virus, to a bot client. When a bot client recognizes the specific command code designed by a bot herder, the bot client achieves the movement according to the received command code.

Because botnet viruses are always changing, in both pattern and attack methods, detecting and protecting against these viruses have become extremely difficult. Most botnet-detecting studies have applied basic Internet virus detection methods such as Honeynet and anomaly-based, signature-based, or machine-learning techniques [8]. The anomaly-based and signature-based methods are the most commonly used. In the anomaly-based method, when the detection system observes that the traffic in the user network exhibits

unusual actions, it determines that the user might be the victim of a botnet virus. The advantage of using the anomaly-based method is that unknown botnets can be detected; the disadvantage is that the rate of misjudgment might be high. In the signature-based method, an unusual packet database is typically built, and when the system detects that the Internet packets of a user conform to the database, the user might be infected by a botnet virus. The advantage of using this method is a high detection rate; however, the database must be frequently updated. Because both these methods possess disadvantages, they were not used in this research; instead, the machine-learning method was adopted for detecting botnet viruses. A method that can be used to detect unknown botnet viruses and has a high detection rate was developed by using feature selection, which was used to identify the critical features of botnet viruses.

Feature selection is used for identifying the critical features of a large amount of multidimensional data and subsequently using those features for analysis. For example, if there are 10 computers in an office and a few of them are infected with an Internet virus, the monthly Internet package data of this office must be collected, which is an extremely large data set because it contains thousands of packet transfer records, and every record has multiple features, such as a host IP address, MAC address, and the protocol type. These data must be analyzed, which subsequently reveals the affected computers as those with several feature anomalies. When the relationship between certain features and viruses is identified, those features must be used with precaution in the future.

This example is an application of feature selection. In a large subset of features, the feature subset most representative or most related to a goal must be identified because although every feature is different, some irrelevant features exist, and certain features are noised or redundant. If all these unnecessary features are considered, the complexity of and space necessary for calculations increase, and the correlation between the feature subset and the goal decreases. Therefore, the purpose of feature selection is to filter unnecessary features and to identify the feature subset that is most related to the goal. Moreover, as the feature number increases, the number of possible relevant feature subsets grows exponentially. When the number of features expands to such a large number that people cannot process it, such problems are called a curse of dimensionality. Conducting a search for all the possible feature subsets involves an excessive amount of time and calculation space, which is not cost-effective; therefore, an efficient and effective optimization algorithm must be used for determining the most suitable feature subset by using limited time and calculation space.

The applications of classification and clustering are widely used in various fields, such as recommendation systems [9], voice communication systems [10], and data mining. Applying feature selection can increase the efficiency of classification and clustering, and increasing classification accuracy and performance through feature selection is imperative. Classification refers to classifying data into appropriate categories. Multiple classification methods can be used, such as a decision tree [11], support vector machine (SVM) [12, 13], or neural network [14, 15]. All these methods are types of supervised learning. Recently, using an SVM has become increasingly common because SVM can achieve high classification with small training sets [13]. The main purpose of the SVM is to establish an optimal hyperplane to classify data and build a classification model.

The metaheuristic algorithm is widely used in various optimization problems, such as feature selection [16, 17] and schedule management [18]. Various metaheuristic algorithms are inspired by natural mechanisms; for example, genetic algorithms (GAs) [19] were inspired by gene mutation and crossover, and particle swarm optimization [20, 21] was inspired by the movement of flocks of birds. Various metaheuristic algorithms exist, such as cat swarm optimization [22], ant colony optimization [23], and artificial fish swarm algorithm (AFSA) [24], which simulates the foraging of fish swarm.

In [25], the results indicated that the AFSA exhibited excellent performance in function optimization, and the potential of applying the AFSA in optimization problems was also revealed. Furthermore, in [26], the researchers proposed a type of feature selection and back-propagation network for botnet detection; however, using an AFSA combined with an SVM classifier might yield superior performance. In this study, a classified model was proposed combining an AFSA algorithm and an SVM. The proposed method was used to identify the critical features determining the pattern of a botnet. The findings indicated that the proposed method can be used to identify the essential botnet features, accurately classifying botnet detection.

Section 2 introduces the SVM, GA, AFSA, and feature characterization of the botnet virus. Section 3 introduces the proposed botnet detection method, using the SVM and the AFSA. Section 4 presents the experiment results and Section 5 provides a conclusion and suggestions for future studies.

2. Background

2.1. Support Vector Machine. The SVM was proposed by Cortes and Vapnik [27]. It is a supervised learning model based on structural risk minimization [27] and the Vapnik–Chervonenkis dimension [28]. An SVM is typically applied in machine learning [29] and for solving classification or regression problems; therefore, the main purpose of an SVM is identifying the optimal hyperplane to analyze various classification data. The optimal hyperplane possesses the maximal margin associated with the various classification data, as shown in Figure 1. Two black points and three white points exist on the maximal margin line, which represent two types of classification data; these points are called support vectors.

These support vectors can be used for classifying new data. When the data is not linearly separable, the kernel function must be used to map the data into the Vapnik-Chervonenkis dimensional space. Three types of kernel function (Φ) exist: radial basis functions (RBFs), polynomials, and sigmoids. Using the appropriate kernel function for transforming the data is imperative for increasing the

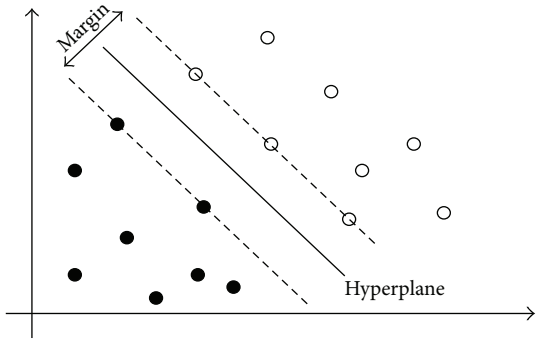

Figure 1: The optimal hyperplane.

classification speed. The three kernel functions are described as follows.

RBF kernel:

$$\Phi\left(x_i - x_j\right) = \exp\left(-\gamma \left\|x_i - x_j\right\|\right). \quad (1)$$

Polynomial kernel:

$$\Phi\left(x_i - x_j\right) = \left(1 + x_i \cdot x_j\right). \quad (2)$$

Sigmoid kernel:

$$\Phi\left(x_i - x_j\right) = \tanh\left(kx_i \cdot x_j - \delta\right). \quad (3)$$

2.2. Genetic Algorithm. The GA was first proposed by J. Holland in 1975, and the main concept of GAs is the simulation of survival of the fittest through crossover and mutation. In this algorithm, chromosomes, which are composed of series genes, play an essential role. Every chromosome has its own fitness value, and the chromosomes that contain high fitness values have a high chance of survival. In this study, an SVM classification accuracy value was used as the fitness value. The GA process is outlined as follows.

(1) *Initialization*. Encode the optimization problem to integrate with GA, create the fitness function and initial N chromosome randomly, and include the gene and the parameters.

(2) *Evaluate Fitness*. Use the fitness function to evaluate the fitness of every chromosome.

(3) *Reproduction*. Determine the reproduction rate of every chromosome based on its fitness value; if the fitness value is high, the reproduction rate is high as well. Use the roulette wheel selection method to select the reproduction chromosomes.

(4) *Crossover*. Randomly match two chromosomes from the reproduction pool and create a new generation of chromosomes by completing the crossover step by applying one-point crossover based on the probability of crossover rate.

(5) *Mutation*. Randomly select dimensions to achieve simple mutation based on the probability of mutation rates; this can increase the opportunities of identifying enhanced solutions.

(6) *Stop the Algorithm If Terminal Criteria Are Satisfied*. If the terminal criteria are satisfied, stop the algorithm and output the optimal solution. Otherwise, start from (2) for the next iteration until the terminal criteria are satisfied.

2.3. Artificial Fish Swarm Algorithm

2.3.1. Conception. The AFSA is an optimization algorithm that simulates the behavior of fish swarm, such as foraging and movement. For example, the position of most fish in a pond is typically the position at which the most food can be obtained. The AFSA includes three main steps, which are Follow, Swarm, and Prey. In the AFSA, these three steps are repeated to determine the optimal solution. Similar to other bioinspired algorithms, the AFSA is used to determine the optimal or most satisfactory solution in a limited time by continually searching for possible solutions using a metaheuristic. In the AFSA, the position of every fish is considered a solution, and every solution has a fitness value that is evaluated using the fitness function. The fitness function changes when different goals are established.

2.3.2. Process. The F_i represent fish i, and C_i represent the center of F_i as mentioned in Table 3. The process of the AFSA is outlined as follows.

(1) *Initialization*. Encode the optimization problem to integrate with AFSA, create the fitness function and initial N fish randomly, and include the position and parameters.

(2) *Evaluate Fitness*. Use the fitness function to evaluate the fitness of every fish.

(3) *Movement of Fish Swarm*. Process the Follow, Swarm, and Prey movements of every fish and determine the optimal solution.

Follow. At this step, the F_i are compared with neighbors based on the optimal fitness value; if the optimal fitness of its neighbor is superior and the crowded degree of this fish is not greater than the maximal crowded degree, then the F_i moves to the position of the neighbor fish, which indicates that the feature subset of the F_i is replaced by that of the neighbor fish. This also indicates that the Follow step is completed. If the Follow step fails, then implement Swarm or Follow for the next fish.

Swarm. At this step, the F_i are compared based on the fitness value of their own, C_i; if the fitness value of the C_i is superior and the crowded degree of the C_i is not greater than the maximal crowded degree, then the F_i moves to the C_i; this indicates that the feature subset of the F_i is replaced by that of the C_i and that the Swarm step is completed. If the Swarm step fails, implement Prey or Follow for the next fish.

Prey. At this step, the F_i randomly changes its own feature subset, indicating that if a feature is 0 and it is chosen to change randomly, this feature becomes 1 and the value of the changed features is not greater than what is visible. If the fitness of the changed feature subset is greater than that of the original, then the changed feature subset replaces the original feature subset which indicates that the Prey step

Table 1: Features of the botnet dataset.

Feature number	Feature name	Feature content
F_1	Total_count	The number of different destination IP address.
F_2	Source_conut	The number of different source IP address.
F_3	Port_count	The number of different port.
F_4	Low_port	The lowest port number.
F_5	High_port	The highest port number.
F_6	TCP_count	The number of different TCP servers
F_7	UDP_count	The number of different UDP servers
F_8	ICMP_count	The number of different ICMP servers
F_9	AvgLength	Average length of packets
F_{10}	StddevLength	The standard deviation of packet length.
F_{11}	Time_Regulartiy	The time regularity of packet sending.
F_{12}	Info_Char	The ASCII content of packets

is completed. If the Prey step fails, the algorithm repeats this step until the repeated number reaches the maximal try number.

(4) *Stop the Algorithm If Terminal Criteria Are Satisfied*. If the terminal criteria are satisfied, then stop the algorithm and output the optimal solution; otherwise, start from (2) for the next iteration until the terminal criteria are satisfied.

2.4. Feature Characterization. To build a botnet detection system, a botnet network data set must be collected. By referencing [26], a local area network (LAN) simulation was built to collect the packet data of network flow; the computers used in this LAN were affected by a botnet virus. The software VirtualBox was used to simulate 10 computers, and the operating systems of those virtual computers included Windows XP, Windows 7, and Linux; subsequently, the computers were connected to the Internet through a Linux router. On these computers, normal user behaviors were simulated, such as playing online games, browsing websites, and watching videos. The packet data of this LAN was collected for 3 weeks, and the packets included the packet between the C & C server and the botnet virus.

Three data sets (Botnet1, Botnet2, and Botnet3) were obtained using various simulated LANs, and each one was infected by a distinct IRC botnet virus. And the duration of each data set was 1 week, the feature number of every data set was 12, and the instances in every data set were 223. The features of each data set, referenced from [26, 30], are shown in Table 1.

Details regarding the features of AvgLength, StddevLength, Time_Regularity, and Info_Char are described as follows.

AvgLength. This feature is the average length of every packet and is calculated by using (4). The variable x is the packet length and N is the total number of packets:

$$\text{AvgLength} = \frac{\text{SUM}(x_i)}{N}. \tag{4}$$

StddevLength. This feature is the standard deviation of the average length of every packet and is calculated by using (5). The variable x is the packet length, μ is the average length of every packet, and N is the total number of packets:

$$\text{StddevLength} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}. \tag{5}$$

Time_Regularity. Because a bot client typically transmits a status report packet to a bot herder, knowing the transmission time regularity of each packet was necessary. This feature is the transmission time regularity of specific packets. A transmission time regularity counter is defined as γ, and if the total number of packets is N, then the total number of γ is N-1, and a set is an array, (i.e., $\gamma = \{\gamma_2, \gamma_3, \ldots, \gamma_n\}$). For example, γ_2 is the transmission time counter that counts the packet number, and the interval time is 2 seconds. Subsequently, the frequency array α and the infrequency array β were defined. The variable t is a constant value between 0 and 1 which was set as 0.5 in this study. The feature Time_Regularity is calculated by using (6):

$$\begin{aligned} \gamma_i &> \frac{2t\sum \gamma_i}{N}, \quad \text{then } \alpha_j = \gamma_i, \\ \gamma_i &\le \frac{2t\sum \gamma_i}{N}, \quad \text{then } \beta_k = \gamma_i, \\ \text{TimeRegularity} &= \text{avg}(\alpha) * (\text{avg}(\alpha) - \text{avg}(\beta)). \end{aligned} \tag{6}$$

Info_Char. Because the specific command that a bot herder uses to control the computer of a bot client typically contains symbols, determining the weight of the symbols in the packets is necessary. This feature is the American Standard Code for Information Interchange (ASCII) counter, and 95 counters exist; each counter counts the number of times relevant ASCII characters appear in all packets. For example, a counter was defined as C; therefore, C_{10} is the counter that counts the number of times the ASCII number 10 appears, even as a decimal, or with the symbol #. The feature Info_Char is calculated by using (7):

$$\text{Info\_Char} = \text{Max}(C_i). \tag{7}$$

3. The Proposed Method

Both the GA and AFSA are metaheuristic algorithms; however, they employ distinct optimization mechanisms. The GA has demonstrated success in numerous applications, but

```
Random initialize Fish Swarm.
WHILE (is terminal condition reached)
  FOR (i = 0; i < NumFish; i++)
    Measure fitness for Fish.
    DO step Follow
    IF (Follow Fail) THEN
      DO step Swarm
        IF (Swarm Fail) THEN
          DO step Prey
        END
    END
  End FOR
End WHILE
Output optimal solution.
```

Pseudocode 1: Pseudocode of AFSA.

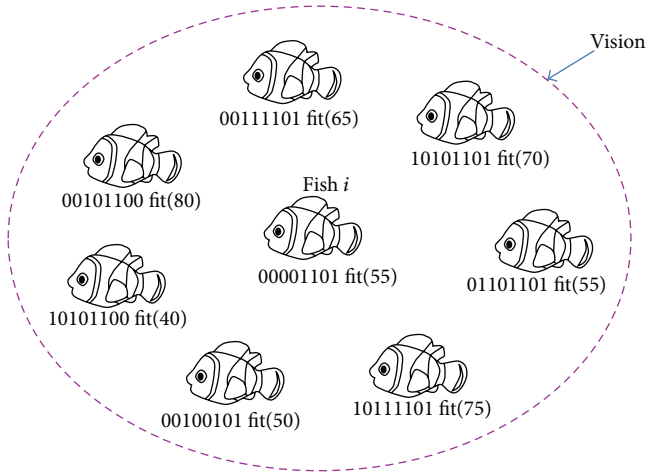

Figure 2: Initiation step of AFSA.

Table 2: Representation of a solution set.

C	γ	F_1	$\cdots F_i \cdots$	F_n

a previous study [25] indicated that AFSA yields superior optimization performance. In this study, the SVM was employed as the classifier, using the AFSA and the GA to perform feature selection. Classifiers can establish a classified model and use it to assign data to the correct categories. First, the data must be divided into multiple components, and every record of this data must have the correct category label. Several pieces of data were regarded as training data and the rest were regarded as test data; subsequently, the training data were input into the classifiers, which was the SVM, to establish the classified model, and then the test data were used to verify this model and obtain accurate classifications. Various components of the data were used to alternately perform these steps, which comprised the cross-validation process. For example, the first portion of the data was used as the test data and the remaining data were used as training data; whereas in the next round, the second portion of the data was used as the test data and the remaining data were used as training data. The pseudocode of AFSA is shown in Pseudocode 1.

In this study, the solution set comprises two parts: (1) the SVM parameters (e.g., C and γ) and (2) the feature subset. In the second part, binary codes were used to represent feature selection; 0 indicated that the feature was not selected and 1 indicated that it was selected. Table 2 shows the solution set.

The feature subset $F(10101)$ indicates that the first, third, and fifth features were selected, whereas the second and fourth features were not selected. Data input into the SVM without preprocessing indicate that every feature is selected and the classification accuracy is likely unreliable. Thus, the AFSA must be used to conduct feature selection. Incorporating the AFSA with the SVM enables the algorithm to identify a superior feature subset such as $F(10101)$. Only data relevant to the selected features are input into the SVM to establish the classification model; this facilitates analyzing whether the classification accuracy is improved. Thus, feature selection is attained and performing the aforementioned steps enables excluding unnecessary data.

At the initial steps of the AFSA, the algorithm assigns a random feature subset to every fish, and the SVM is used to obtain the classification accuracy based on the fitness of every fish. Subsequently, Follow, Swarm, and Prey processes are implemented to obtain the optimal solution. The definitions of the parameters, referenced from [31], are presented in Table 3.

The steps involved in the AFSA-SVM method are presented as follows.

(1) Initiation: randomly assign a feature subset to N fish. Define all parameters including vision, maximal crowded degree, and maximal trial number. For example, Figure 2 shows that eight fish were initiated; each fish has its own feature and the circle represents the vision of fish i.

(2) Evaluate the classification value as a fitness value of the feature subset of each fish by using the SVM as shown in Figure 2.

(3) Starting with the first fish, implement the Follow step. If Follow is successful, perform step 6; otherwise perform step 4. For example, in Figure 2, the fitness value of fish i is 55; by contrast, the best fitness neighbor exhibits a value of 80. Thus, the best fitness neighbor demonstrates a superior fitness value, indicating that a superior fish is located in the vision of fish i. Therefore, the Follow step is successful and fish i moves to the location of the best fitness neighbor, replacing its feature subset as shown in Figure 3.

(4) Implement the Swarm step for the same fish. If successful, perform step 6; otherwise perform step 5. For example, in Figure 2, calculate the center subset by using (3) in Table 3 and then use the SVM to evaluate its fitness value, comparing the fitness value of fish i and the center subset. If the fitness value of the center subset is the highest, the Swarm step

Table 3: Parameters of AFSA.

Parameter name	Definition
Distance	The distance between F_i, F_j is obtained through formula (1). Those two fish have the same number of features, k, and if the first feature of F_i is 0 and the first feature of F_j is 0, then the distance between F_i, F_j will remain the same. But if the first feature of F_i is different from the first feature of F_j, the distance between F_i, F_j will be plus one. The distance between two fish is the sum of the differences of every feature: $\text{distance}(F_i, F_j) = \sum_{k=1}^{k} \lvert F_i(k) - F_j(k) \rvert$ (1)
Vision	The visibility of a fish and also the maximum distance that this fish can move. In other words, it is the maximum number of features that one fish can be change
Neighbor	The neighbor of F_i is all the fish that are in F_i's vision; if the distance between F_k and F_i is greater than 0 and less than or equal to vision, F_k is the neighbor of F_i. It is obtained through formula (2): $\text{Neighbor}(F_i) = \{F_k \mid 0 < \text{distance}(F_i, F_k) \le \text{vision}\}$ (2)
Center	The center of F_i is the center of F_i's neighbor. It can be considered as a fish; the center feature is obtained through formula (3); if more than half F_i's neighbors' feature i are 0, then the center of F_i's feature i will be 0, and vice versa: $F_{\text{center}}(i) = \begin{cases} 0, & \sum_{k=1}^{k} F_k(i) < \frac{k}{2} \\ 1, & \sum_{k=1}^{k} F_k(i) \ge \frac{k}{2} \end{cases}$ (3)
Crowded degree	The crowded degree of F_i is to represent the density of F_i's position; it is obtained through formula (4): $\text{Crowded Degree}(F_i) = \frac{\text{Neighbors of } F_i}{\text{Total number of Fishes}}$ (4)
The maximum crowded degree	The limited number of crowded degree: if the crowded degree of F_i is greater than the limited number, then other fish cannot approach F_i.
The maximum trial number	The maximum number can perform the Prey movement

is successful and fish i moves to the center subset, replacing the feature subset.

(5) Implement the Prey step for the same fish. After the Prey step, perform step 6. For example, in Figure 2, the feature subset of fish i is 00001101. The features randomly change each time the Prey step is executed. The number of changed features must be less than vision and the number of times Prey is executed must be less than the maximal trial number. After changing the feature subset, evaluate the fitness value by using the SVM and compare it with the original feature subset of fish i; if the changed feature subset exhibits superior fitness, the Prey step is successful and the feature subset is replaced with the original feature subset.

(6) Determine if the current fish is the last in the fish swarm. If no, then begin from step 3 and perform the steps for the next fish; if yes, then perform step 7.

(7) Determine the fitness of every fish; if excellent fitness is observed, then update the optimal solution and perform step 8.

(8) Determine if the terminal criteria are satisfied and stop the algorithm; otherwise start from step 3 to begin the next iteration. Figure 4 shows the AFSA flow chart.

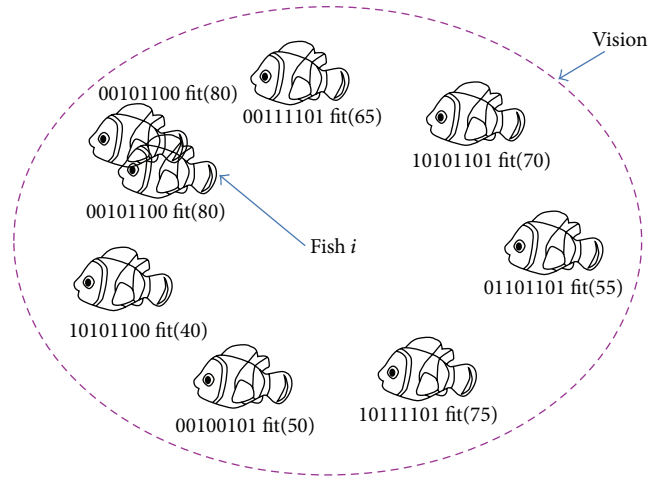

Figure 3: Follow step of AFSA.

4. Experimental Results

To estimate the performance of feature selection using the AFSA combined with an SVM, the performance of the AFSA was compared with that of a GA, including the classification accuracy, the number of features of the optimal solution subset, and the time spent applying each algorithm to perform calculations. For both the AFSA and GA, the terminal

Table 4: The experimental results of AFSA and GA, 5-fold cross-validations.

Datasets	AFSA-SVM			GA-SVM		
	No. of selected features	Average accuracy rate (%)	Executed time (sec)	No. of selected features	Average accuracy rate (%)	Executed time (sec)
Botnet1	6	97.76	19843	7.2	97.30	22831
Botnet2	5.6	98.22	21460	6.8	96.87	22868
Botnet3	6	99.56	22436	7.8	99.11	21583

condition of each fold was when the optimal solution was not updated after 1 hour. The algorithm parameters used in this study are presented as follows.

AFSA. The number of fish was 30, the maximal number of trials was 30, and the maximal crowded degree was 0.5.

GA. The genetic number was 20, and the mutation rate was 0.05.

The computer used to implement the AFSA and GA algorithms was a desktop computer. The operating system was Microsoft Windows 7, the coprocessor was a 2.66-GHz Intel Core 2 Quad Processor Q8400, the amount of memory was 2 GB, and the algorithms were coded using Dev C++. The classifier used was the Library for Support Vector Machines [32] and the RBF kernel function.

4.1. Experiment 1. Simulated botnet data sets were collected as mentioned in Section 2.4, and Table 4 shows the experimental results for each data set classified using the AFSA and the GA and a fivefold cross-validation process. The results are the average of the fivefold. The average classification accuracy, number of selected features of the optimal solution subset, and total time between the AFSA and GA were compared. The AFSA was more accurate than the GA was for all data sets, indicating that an increased botnet detection rate can be obtained. The number of selected features of the AFSA was also less than the number of selected features of the GA; thus, the amount of processed data involved in botnet detection was reduced, thereby reducing the detection time. Ultimately, the total time the AFSA spent was less than that of the GA, except for the data set Botnet3; based on these results, the AFSA can be used to obtain higher classification rates, identify the optimal feature subset by using less selected features, and spend less time performing calculations than using the GA can.

To determine the critical features, the total number of selected features in the optimal subset output by using AFSA-SVM was calculated and the results are presented in Table 5. If the number of selected features is high, it indicates that the feature is critical for classifying the input data when using SVM. Thus, the features that exhibit high counts are the features critical to botnet detection.

The results in Table 5 revealed that Features 9 and 11, AvgLength and Time_Regularity, are the features most often selected from the optimal feature subset, followed by Feature 12, Info_Char. Because of idle time, the bot herder was not always controlling the computer of the bot client; however, the computer of the bot clients still sent a status report packet to the bot herder regularly; therefore, AvgLength is a critical feature. Furthermore, the transmission time interval exhibited a regular pattern in sending the status report packet, which is why Time_Regularity is such a critical feature. Moreover, because the specific commands sent by the bot herder typically contain specific symbols, identifying the

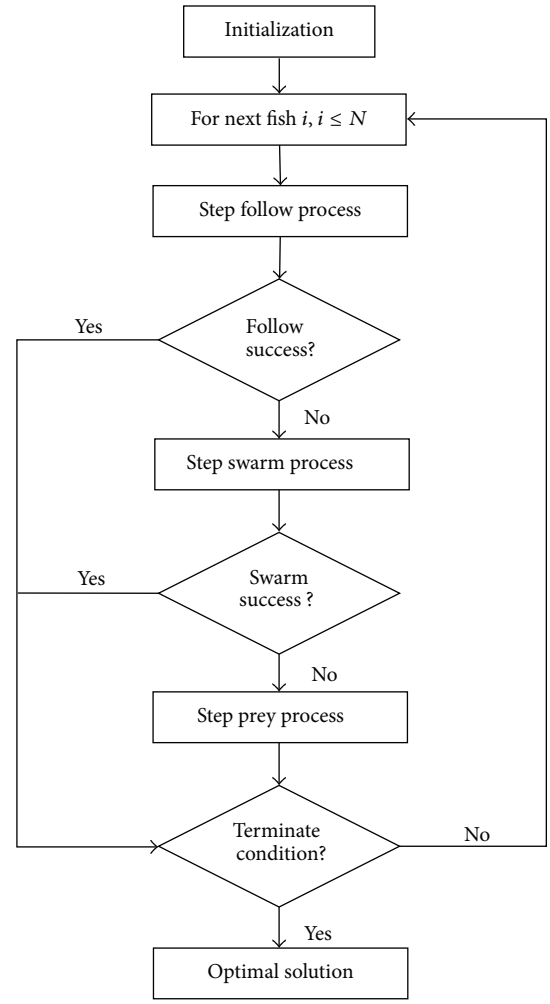

Figure 4: Flow chart of the proposed method.

Table 5: Count of selected feature by using 5-fold cross-validations.

Count of selected feature											
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}
10	13	12	12	14	13	10	8	19	16	19	18

Table 6: The experimental results of AFSA and GA, 10-fold cross-validations.

Datasets	AFSA-SVM			GA-SVM		
	No. of selected features	Average accuracy rate (%)	Executed time (sec)	No. of selected features	Average accuracy rate (%)	Executed time (sec)
Botnet1	4	100	3934	5.8	97.31	25662
Botnet2	4.4	99.11	13505	6.2	99.56	5234
Botnet3	5	100	10256	6.5	97.29	16523

Table 7: Count of selected feature by using 10-fold cross-validations.

Count of selected feature											
F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}
27	27	30	23	32	30	19	23	34	34	39	31

specific symbols that the bot herder uses may help identify a computer that is infected.

4.2. Experiment 2. Tenfold cross-validation was subsequently used, and the terminal condition of each fold was changed as if the optimal solution had not been updated after 1 hour or the classification accuracy was 100%. The results are shown in Table 6. Whether the optimal feature subset falls into the local optimal can be determined. The execution time can be substantially reduced, yielding increased classification accuracy and fewer selected features compared with using fivefold cross-validation. When using the tenfold cross-validation method, the training data grow, enabling the population to comprise additional samples; however, population growth may substantially increase the convergence rate.

The total number of selected features in the optimal subset by using tenfold cross-validations was shown in Table 7. The results shown in Table 7 indicate that Features 9, 10, and 11, representing AvgLength, StddevLength, and Time_Regularity, respectively, were most often selected from the optimal feature subset when using 10-fold cross-validation; this was similar to the results of using fivefold cross-validation, excepting Feature 10 (StddevLength). The classification rate increased when the selected number of StddevLength increased. Therefore, the StddevLength feature was critical to botnet detection. StddevLength represented the standard deviation of the packet length number; the bot clients regularly sent status report packets to the bot herder. These packets were typically short and consistent in length; thus, the StddevLength was the vital feature in botnet detection.

5. Conclusion and Future Work

In this study, a feature selection method for detecting botnet viruses is proposed, which is the AFSA-SVM method. Based on the experimental results, using the AFSA yielded only slightly higher classification accuracies than using the GA, but less time was spent to obtain a lesser number of feature subsets. In practical applications, classification accuracy is typically the first priority, but in certain processes, such as botnet virus detection, detection speed is as crucial as accuracy. To obtain the desired detection speed, the data required for processing must be reduced under the premise that the accuracy level is the same; therefore, in this scenario, the AFSA-SVM method is superior.

The result also shows that both GA and AFSA can still be applied for identifying the critical features of botnet, filtering unnecessary features, and using these algorithms in various applications easily. In our research, an IRC botnet was collected as the data set; however, in real world situations, botnet viruses are constantly changing, and an increasing number of botnet viruses are using peer to peer (P2P) or other protocols as the attack method. Therefore, in future studies, the proposed method must be tested for detecting P2P protocols or other types of botnet viruses. Finally, a feature-selection-based detection system for detecting botnet viruses can hopefully be constructed in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] Y. Pan and J. Zhang, "Parallel programming on cloud computing platforms—challenges and solutions," *Journal of Convergence*, vol. 3, no. 4, pp. 23–28, 2012.

[3] K. Wang, C.-Y. Huang, S.-J. Lin, and Y.-D. Lin, "A fuzzy pattern-based filtering algorithm for botnet detection," *Computer Networks*, vol. 55, no. 15, pp. 3275–3286, 2011.

[4] H. Choi and H. Lee, "Identifying botnets by capturing group activities in DNS traffic," *Computer Networks*, vol. 56, no. 1, pp. 20–33, 2012.

[5] W. T. Strayer, D. Lapsely, R. Walsh, and C. Livadas, "Botnet detection based on network behavior," *Advances in Information Security*, vol. 36, pp. 1–24, 2008.

[6] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proceedings of the 6th ACM SIGCOMM on Internet Measurement Conference (IMC '06)*, pp. 41–52, October 2006.

[7] M. S. Obaidat and F. Zarai, "Novel algorithm for secured mobility and IP traceability for WLAN networks," *Journal of Convergence*, vol. 3, no. 2, pp. 1–8, 2012.

[8] M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection," in *Proceedings of the 3rd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE '09)*, pp. 268–273, June 2009.

[9] R. Pan, G. Xu, B. Fu, P. Dolog, Z. Wang, and M. Leginus, "Improving recommendations by the clustering of tag neighbours," *Journal of Convergence*, vol. 3, no. 1, pp. 13–20, 2012.

[10] A. Bhattacharya, W. Wu, and Z. Yang, "Quality of experience evaluation of voice communication: an affect-based approach," *Human-Centric Computing and Information Sciences*, vol. 2, article 7, 2012.

[11] J. R. Quinlan, *C4.5: Programs for Machine Learning*, The Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers, San Mateo, Calif, USA, 1993.

[12] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[13] M. Abdel Fattah, "The use of MSVM and HMM for sentence alignment," *Journal of Information Processing Systems*, vol. 8, no. 2, 2012.

[14] G. P. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man and Cybernetics C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, 2000.

[15] K. Sarkar, M. Nasipuri, and S. Ghose, "Machine learning based keyphrase extraction: comparing decision trees, naïve Bayes, and artificial neural networks," *Journal of Information Processing Systems*, vol. 8, no. 4, pp. 693–712, 2012.

[16] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 1–4, pp. 131–156, 1997.

[17] A. James and S. Dimitrijev, "Ranked selection of nearest discriminating features," *Human-Centric Computing and Information Sciences*, vol. 2, article 12, 2012.

[18] S. Farzi, "Efficient job scheduling in grid computing with modified artificial fish swarm algorithm," *International Journal of Computer Theory and Engineering*, vol. 1, no. 1, pp. 13–18, 2009.

[19] C.-L. Huang and C.-J. Wang, "A GA-based feature selection and parameters optimizationfor support vector machines," *Expert Systems with Applications*, vol. 31, no. 2, pp. 231–240, 2006.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.

[21] B. Singh and D. Lobiyal, "A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks," *Human-Centric Computing and Information Sciences*, vol. 2, article 13, 2012.

[22] K.-C. Lin and H.-Y. Chien, "CSO-based feature selection and parameter optimization for support vector machine," in *Proceedings of the Joint Conferences on Pervasive Computing (JCPC '09)*, pp. 783–788, December 2009.

[23] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.

[24] X.-L. Li, Z.-J. Shao, and J.-X. Qian, "Optimizing method based on autonomous animats: fish-swarm Algorithm," *System Engineering Theory and Practice*, vol. 22, no. 11, pp. 32–38, 2002.

[25] H. Chen, S. Wang, J. Li, and Y. Li, "A hybrid of artificial fish swarm algorithm and particle swarm optimization for feedforward neural network training," in *Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering*, 2007.

[26] J. L. Liao and K. C. Lin, *A Study of Feature Selection Integrated with Back-Propagation Network for Botnet Detection*, National Chung Hsing University, Taichung, Taiwan, 2013.

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[28] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[29] R. Malhotra and A. Jain, "Fault prediction using statistical and machine learning methods for improving software quality," *Journal of Information Processing Systems*, vol. 8, no. 2, pp. 241–262, 2012.

[30] C. Langin, H. Zhou, S. Rahimi, B. Gupta, M. Zargham, and M. R. Sayeh, "A self-organizing map and its modeling for discovering malignant network traffic," in *Proceedings of the IEEE Symposium on Computational Intelligence in Cyber Security (CICS '09)*, pp. 122–129, Nashville, Tenn, USA, April 2009.

[31] T. Liu, Y.-B. Hou, A.-L. Qi, and X.-T. Chang, "Feature optimization based on Artificial Fish-swarm Algorithm in intrusion detections," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, pp. 542–545, April 2009.

[32] C. C. Chang and C. J. Lin, "LIBSVM: A Library for Support Vector Machines," http://www.csie.ntu.edu.tw/~cjlin/libsvm/.