# A Novel Feature-Selection Approach Based on Particle Swarm Optimization Algorithm for Intrusion Detection Systems (Workshop Paper)

Jianzhen Wang(✉) and Yan Jin

Business College of Shanxi University, Taiyuan 030031, Shanxi, China
aawangjz@163.com, px_happy@163.com

**Abstract.** This paper proposes a feature selection approach, based on improved Discrete Particle Swarm Optimization (DPSO), to solve the "dimension disaster" problem in data classification; it is named Progressive Binary Particle Swarm Optimization (PBPSO). This feature selection approach is highly problem-dependent and influenced by the locations of particles. It adopts the principle of "partial retention - change - reduction of duplication - update" in the process of selection, and defines a new fitness function describing the correlation between the features and class labels. Experimentation was conducted using of the KDDCup99 data set to evaluate our proposed PBPSO. The experimental results show that 14 features were selected from the original data space with 41 features. Three classic classifiers, namely J48, Naive Bayes and ID3, were then used to further evaluate the performance of the selected features. The classification accuracy rates on the different classifiers achieved using the selected feature subset are similar to those achieved using the original feature set. The training time is, however, significantly reduced. In comparison with other similar algorithms, including Genetic Algorithm GA and Greedy Algorithm FGA. The results show that the PBPSO extracts fewer features, achieves slightly higher classification accuracy, and less time consuming in terms of model training. It has been demonstrated that the PBPSO enhances the practicability of certain classification algorithms in handling high-dimensional data.

**Keywords:** Feature selection · Discrete particle swarm algorithm · Correlation analysis · Correct classification rate · Modeling efficiency

## 1 Introduction

With the tremendous growth of computer network technology, various issues of network security are arising accordingly. Firewall is the first gate for network security. Traditional firewall, based on the static security technology, is unable to prevent attacks from the inter-network, attacks bypassing firewall, new attacks and is unable to

Supported by Shanxi Provincial Department of Science and Technology, Natural Science Fund Project, 2016; Shanxi Scholarship Council, Scholarship Fund Project, 2016; State Administration of Foreign Experts, Overseas Training Fund Program.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2019
Published by Springer Nature Switzerland AG 2019. All Rights Reserved

X. Wang et al. (Eds.): CollaborateCom 2019, LNICST 292, pp. 455–465, 2019.
https://doi.org/10.1007/978-3-030-30146-0_32

effectively defense virus [1]. IDS, Intrusion Detection System, is the second gate for network security, which is able to dynamically monitor inter-network or host computer, to detect various attacks in time and response, and to compensate for the defect of the static security technology.

According to the technology approach adopted, IDS is comprised of misuse detection and anomaly detection [2, 3]. Currently, misuse detection is the major monitoring approach for IDS, and the attacking feature database is its core component. misuse detection can detect attacks of known types with low rate of false alarm and high rate of under reporting, and do nothing for the new unknown attacks, which requires constantly upgrading of feature database to guarantee the completeness of system detection ability. The completeness and accuracy of attacking feature database of misuse detection directly affect the performance of IDS based on misuse detection. Anomaly detection, based on normal behavior, builds feature outline and cannot detect the specific type of attacks with high rate of false alarm and low rate of under reporting, but is able to detect the new unknown attacks. Anomaly detection mostly adopt the statistical analysis or based on the approach of rule description, builds behavior feature outline of normal user in the system. These two detection approaches are complementary in the aspects of applicable objects and ways of detection, and so on. The effective combination of both can offset the defects of each other, improving the performance of intrusion detection.

From key point of computer network or within the host computer, IDS collects data and make classification. Only a small proportion of massive feature of data source has something to do with attacking classification. These massive and redundant features cause enormous drop of classification efficiency and classification accuracy of attacking classifier. Caruana pointed out that variety of feature significantly affected the quality of inductive learning [4]. Extracting effective feature subset from massive features not only improves the performance of learning algorithm, but increases the training speed of algorithm as well. The smaller a feature subset is, the more representative the selected feature might be, the higher the quality of generated rule is. Consequently, feature selection is critical to increase the classification efficiency of network attacking [5], and is the core issue of Intrusion Detection System [6]. The feature selection of intrusion detection in fact is strategy that we are seeking for an optimal point on the aspects of reduction of rate of false alarm and rate of underreporting and improvement of system performance. Effective feature selection is the focus of this paper.

## 2 Feature-Selection Approach

Choosing some most effective features from a set of features to achieve the objective of reducing the dimension of feature space, this process is called feature selection. The basic task of feature selection is to find out a feature subset from a known features set, in order to describe the known samples in a consistent way [7]. The quantity of feature selection of intrusion detection should be appropriate, neither too many nor too few. Too many features mean strong particularity which is easy to cause under reporting, intensive computation and poor real-time affect the system efficiency. Too few features mean strong universality which is easy to cause false alarm. Before and after feature selection, the distribution of category should be as consistent as possible. The classification accuracy cannot drop obviously.

According to whether depend on machine learning algorithm, feature selection can be divided into two models: filter model and wrapper model [8]. In the filter model, feature selection approach is independent of any inductive learning algorithm. It can select feature subset by using a preprocessing pattern, selected feature subset is independent of any particular algorithm, and then inductive learning. In the wrapper model, feature selection approach directly optimizes the certain particular algorithm, by evaluating algorithms' generalization ability for the selected feature subset in each step to achieve feature selection. Wrapper model regards feature selection as the searching issue within all possible feature space.

Filter model has higher computation efficiency than wrapper model, and is a classification algorithm which is independent of the ultimate choice. But because the adopted evaluation function is likely to lead to incorrect orientation, causing feature selection deviate from the ultimate objective, and less effective. Wrapper model wraps learning algorithm into feature selection algorithm, classification accuracy of subset ultimately selected is pretty higher. But when data size is higher, it needs to take massive computing resources. Feature selection is actually a combinatorial optimization issue, which has proven to be NP difficult problem [9, 10]. Search algorithm and evaluation function are two main components of feature selection algorithm. Adopting some heuristic search algorithm to select a features subset of suboptimal can reduce the computational workload. After the generation of feature subset, use an evaluation function to calculate whether the feature subset is good or bad, and compare the result with previous best one. After several iterations, the best fitted features subset is the ultimate selected result.

This paper focuses on the issue of feature selection for intrusion detection. The data set contains massive data and computation is time-consuming, it is inappropriate to use wrapper model. Therefore, this paper adopts filter model to select feature subset, takes advantage of particle swarm algorithm with fast convergence to search within the feature space, simultaneously introduce the correlation analysis to guide the search of algorithm.

## 3 Particle Swarm Optimization Algorithm

### 3.1 Basic Particle Swarm Algorithm

PSO, Particle Swarm Optimization [11] is a global random search algorithm based on swarm intelligence proposed by James Kennedy, an American social psychologist and Russell Eberhart, an electrical engineer in 1995 through simulating birds' migrating and flocking behavior during the process of foraging.

First of all, PSO initialize a group of random particles (random solution), and then find the optimal solution through iteration. In the each iteration, the particles update themselves through tracking two extreme values. One is the optimal solution found by the particle itself, which is called individual extreme value Pi, the other one is the optimal solution found by the whole population, which is called global extreme value Pg. Assume size of particles population is N, particles' current position is represented as $X_i^k = (x_1^k, \cdots, x_n^k, \cdots, x_N^k)$, $x_n^k \in [l_n, u_n]$, $1 \leq n \leq N$, $l_n$ and $u_n$ represent the upper and lower bound of nth dimension respectively. Particles' current speed is $V_i^k = (v_1^k, \cdots, v_n^k, \cdots, v_N^k)$, $V_i^k$ is limited between the maximum value $V_{\max}^k = (v_{\max,1}^k, \cdots, v_{\max,n}^k, \cdots, v_{\max,N}^k)$ and minimum value $V_{\min}^k = (v_{\min,1}^k, \cdots, v_{\min,n}^k, \cdots, v_{\min,N}^k)$. After particles find the two extreme values above, then update speed and position of themselves according to formulas (1) and (2).

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \tag{1}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{2}$$

Where ω is inertia weight, c1, c2 are constant, called learning factors, used to adjust the relative importance of individual extreme value and global extreme value. r1, r2 are random numbers uniformly distributed among (0, 1). $P_i^k, P_g^k$ represent position of individual extreme value and position of global extreme value for particles' kth iteration. Conditions for iteration termination are that either reach the maximum time of iteration or meet the minimum threshold of fitness. The first part of formula (1) is particles' speed in previous step, indicating particles' current state, the second part is the reflection of particle itself, which is the cognitive part, particles adjust their speed and position for next step by thinking the position of themselves, in this way particles have sufficiently strong global search capability to avoid falling into a local minimum, the third part represents that particles update their next step through the information exchange with other particles.

### 3.2 Discrete Binary Particle Swarm Optimization Algorithm

In 1997, Kennedy and Eberhart proposed Discrete Binary Particle Swarm Optimization (BSPO) of PSO algorithm, which make it into the field of combinatorial optimization [12]. BPSO adopts binary encoded form, constrains each dimension of $\mathrm{x}_i^k$ and $\mathrm{p}_i^k$ as 1 or 0 within BSPO model, but there is no such constraint for speed $\mathrm{v}_i^k$. The Sigmoid function of speed (formula 3) indicates the possibility of position state changes. Formula for speed update of BPSO is unchanged, but formula for position update is altered to formula (4). Where rand() is random numbers among [0, 1]. Within the discrete binary model, $v_{max}^k$ is preserved, playing the role of constraining the ultimate possibility of $X_i^k$ is 1 or 0. In fact, normally $v_{max}^k i$ s set between ±4.0, so that there is at least one opportunity making $\mathrm{S}\left(v_{max}^k\right) \approx 0.0180$, meaning under the condition of upcoming state changing.

$$\mathrm{S(v)} = \frac{1}{1 + e^{-v}} \tag{3}$$

$$\begin{aligned} &\text{if } \left(\text{rand}() \prec \mathrm{S}(v_i^k)\right) \text{ then } x_i^k = 1 \\ &\text{else } X_i^k = 0 \end{aligned} \tag{4}$$

PSO can be directly used to solve continuous problem, while intrusion detection feature selection is non-continuous problem. Consequently, this paper uses BPSO Algorithm, each particle has its own position and speed, and particles' position represents one possible solution for the problem. The implement of algorithm requires the re-definition of particle coding scheme, particles' position and speed, and the updating rules.

## 4 Feature-Selection Approach Based on BPSO for IDS

### 4.1 Data Set Selection

This paper adopts KDDCup99 data set [13]. This dataset is a 9 weeks data selected from a simulated U.S. Air Force LAN, which has great similarity with real-world scenarios and is widely used to test the efficiency of intrusion detection system. KDDCup99 data set is divided into marked training dataset and unmarked testing dataset. According to Table 1, training dataset has different probability distribution with testing dataset. Testing dataset contains some types of attacking which do not appear in the training dataset, making intrusion detection more realistic. According to Table 2, training dataset contains one kind of normal identifier class-Normal and 22 kinds of training attacking types. There are another 14 kinds of attacking types appearing only in the testing dataset. In KDDCup99 data set, each connected record contains 41 fixed feature attribute and one kind of class identifier. Identifier is used to indicate that this connected record is either normal or a certain specific attacking type. Within 41 fixed feature attributes, nine of them are discrete and the rest is continuous.

**Table 1.** Distribution of connection types in 10% KDDCup99 data set

| Attack classes | Training data | | Testing data | |
|---|---|---|---|---|
| | Number | Percentage | Number | Percentage |
| Normal (0) | 97278 | 19.691066 | 60593 | 19.481463 |
| Probe (1) | 4107 | 0.83134 1 | 4166 | 1.339425 |
| DoS (2) | 391458 | 79.239142 | 229853 | 73.900826 |
| U2R (3) | 52 | 0.010526 | 228 | 0.073305 |
| R2L (4) | 1126 | 0.227926 | 16189 | 5.204981 |

### 4.2 Algorithm Design

Traditional dimension reduction method is a non-supervisory algorithm. Despite dimension reduction of data source can simplify the data source, the low-dimensional data is not completely equivalent to the original data. The key of taking advantage of particle swarm algorithm to solve the issue of feature selection rests on the selection of coding scheme and fitness function [14, 15]. Coding is determined by the essence of the issue and the space where the particle is, the fitness function reflects the connection between practical issues and optimizing algorithm.

**Table 2.** Sample size in 10% KDDCup99 training dataset

| Attack classes | Attack label name |
|---|---|
| Normal | Normal97278 |
| DOS | Back2203, Land21, Neptune107201, Pod264, Smurf280790, Teardrop979 |
| Probe | Ipsweep1247, Nmap231, Portsweep1040, Satan1589 |
| R2L | Ftp_write8, Guess_passwd53, Imap12, Multihop7, Phf4, Warezclient1020, Warezmaster20, Spy2 |
| U2R | Buffer_overflow30, Loadmodul9, Perl3, Rootkit10 |
| Total | 494,020 |

### 4.2.1 Coding Scheme

Coding scheme is that for a given network states data set D including N-dimensional feature, the goal of feature selection is to select a feature subset R making objective function optimal [16].This paper, based on the analysis of KDDCup99 standard data set, determines to make natural number coding for particles. Using natural number to represent feature attributes, in this way, coding is simple and easy to implement. Several of features compose of a feature subset, each feature subset is a natural number coding sequence. The natural number coding sequence corresponds to a particle of the population, while each particle corresponds to a feasible solution, the combination of different features attributes of 41 features attributes.

### 4.2.2 Definition of Particles Position, Speed, and the Updating Rules Within PSO Algorithm

Each particle's position, within PSO algorithm, is a potential solution [17, 18]. Particles' position is the natural number sequence between 1 and 41, numbers of sequence dimension represent numbers of feature attributes. Dimensions of each particle vary from 2 to 41, respectively representing various attributes from 2 to 41. The dimensions are determined by a parameter. For each dimension, use PSO algorithm to optimize, finally make comparison for optimal solutions corresponding each dimension obtained, then select best fitness as the ultimate solution, realize extracting effective features. In the sequence which is represented by particles' position, because sequence of various features in feature subset has nothing to do with ultimate result. To prevent repetitive combination, particles' position coding is arranged from small to big, like (3, 9, 16, 38).

Particles' speed determines the updating way of particles' position. This study proposes rules for position updating: "partially preserved-alter-compare and duplicate checking-updating". In this way, the rules ensure all possibility of combinations of different feature attributes, and avoid the repetitive search, improving the convergence rate for the algorithm. For instance, a particle's current position is (2, 4, 5), preserve 2 unchanged in the next step, and then compare historical position, randomly altering the rest coding and there is no repetition with historical position. In this instance, if there is historical position like (2, 7, 9), there will not be such combination like (2, 7, 9) when 4 and 5 is altered, coding varying from 1 to 41. At initial, the particles' population is randomly generated, the size of particles' population is 30, and the maximum times of iteration are 1000.

### 4.2.3 Definition of Fitness Function

In the feature selection within filter model, selection of fitness function is very significant [19]. This paper adopts correlation evaluation method, uses the correlation of feature attributes and attacking types to evaluate particles' pros and cons, furthermore define the fitness function. Particles' position within PSO algorithm represents the independent variable in correlation function, and particles' speed is the alteration of attribute number, altering and updating particles' coding in the search process of algorithm iteration. Correlation of various feature attributes and attacking types is the fitness function, and take advantage of joint probability of each attribute and identifier class to measure how great or small the correlation is, use discrete PSO algorithm search to find the most correlated set of solutions as feature subset this study extracts. Fitness function of particles is computed according to formulas (5) and (6). Where i and j contain all attributes of feature attributes set, C is identifier class. The bigger the value of fitness function is, the greater the correlation of the feature attribute and the class is, the higher the fitness of particles is.

$$f = \frac{\sum_{j} \rho\left(A_j, C\right)}{\sum_{i} \sum_{j} \rho\left(A_i, A_j\right)} \tag{5}$$

$$\rho(A, B) = \frac{P(AB) - P(A)P(B)}{\sqrt{P(A)(1 - P(A))P(B)P(1 - P(B))}} \tag{6}$$

## 4.3 Experimental Result

Based on the ideas above to program, PSO tool-kit is developed in Matlab [20]. After the running of PSO algorithm main programming, it returns optimal solutions and its corresponding fitness. The experimental result extracts 14 feature attributes to constitute feature subset, out from 41 fixed feature attributes within each connected record in KDDCup99 Data set, reducing feature dimensions, see Table 3.

**Table 3.** Features subset using BPSO

| Features number | Features subset |
|---|---|
| 14 | Duration, Protocol_type, Service, Flag, Src_bytes, Dst_bytes, Land, Wrong_fragment, Count, Srv_count, Dst_host_count, Dst_host_diff_srv_rate,Dst_host_same_src_port_rate, Dst_host_srv_diff_host_rate |

## 5 Algorithm Validation Test

Select part of data from 10% KDDCup99 training data set to test algorithm. To keep the classification information of the original data, adopt sampling method without replacement. For classes more than 10,000 samples, randomly select 10 percent of them as the sample, for classes less than 10,000 samples, select them all, ultimately 22 classes are generated, and 57,141 samples, in total, constitute experimental sample collection, see Table 4.

**Table 4.** Details of sample dataset

| Attack classes | Attack label name |
|---|---|
| Normal | Normal9800 |
| DOS | Back2203, Land21, Neptune10720, Pod264, Smurf28000, Teardrop879 |
| Probe | Ipsweep1247, Nmap231, Portsweep1040, Satan1589 |
| R2L | Ftp_write8, Guess_passwd53, Imap12, Multihop7, Phf4, Warezclient1020, Warezmaster20, Spy2 |
| U2R | Buffer_overflow30, Loadmodul9, Perl3, Rootkit10 |
| Total | 57,141 |

Weka, based on Java, is an intelligent analysis environment used for data mining and knowledge discovery [21]. In this experimental sample collection, use classic classification like J48, ID3 and Naive Bayes provided by Weka platform to build model training learning, respectively taking advantage of original feature attributes of KDDCup99 data set and 14 features attributes after feature selection. Adopting 10-fold cross-validation verify the influence classification result of feature selection algorithm this study proposed has on various classification algorithm [22, 23]. It turns out that after feature selection based on BPSO algorithm, there's slight increase for the rate of correct classification, significantly reducing the time to build a classifier, see Tables 5 and 6.

**Table 5.** Classification results of features collection

| Classification | Sample number | Features number | Correctly classified rate | Classifier build time |
|---|---|---|---|---|
| J48 | 57141 | 41 | 99.8740% | 12.33 |
| ID3 | 57141 | 41 | 91.9983% | 7.08 |
| NaiveBayes | 57141 | 41 | 96.8411% | 2.23 |

Furthermore, analyze feature subset respectively extracted by discrete BPSO algorithm, Genetic Algorithm (GA) and First Greedy Algorithm (FGA) [24, 25], see Table 7. Adopt Weka platform built-in J48 decision tree classifier and Naive Bayes classifier respectively to make classification, verifying operating performance of feature

**Table 6.** Classification results of features subset

| Classification | Sample number | Features number | Correctly classified rate | Classifier build time |
|---|---|---|---|---|
| J48 | 57141 | 14 | 99.8782% | 2.77 |
| ID3 | 57141 | 14 | 91.9983% | 4.95 |
| NaiveBayes | 57141 | 14 | 96.1341% | 0.55 |

selection algorithm based on BPSO and other feature selection algorithm. Experimental result show that the algorithm of this paper can achieve relatively higher rate of correct classification by using fewer numbers of features, and has a big advantage on running time as well, which is an effective feature selection method, according to Tables 8 and 9.

**Table 7.** Features subset of different algorithms

| Algorithms | Features number | Features subset |
|---|---|---|
| BPSO | 14 | duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_ fragment, srv_count, dst_host_count, dst_host_diff_srv_rate, dst_host_same_src_port_rate, count, dst_host_srv_diff_host_rate |
| GA | 19 | protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, num_shells, srv_count, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_srv_count, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_ host_rerror_ rate, count, hot |
| FGA | 15 | protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, root_shell, count, srv_count, diff_srv_ rate, dst_ host_srv_count, dst_ host_srv_ diff_host_rate, dst_host_rerrorr_ rate |

**Table 8.** Classification results of features subset using J48

| Algorithms | Sample number | Features number | Correctly classified rate | Time of build model |
|---|---|---|---|---|
| BPSO | 57141 | 14 | 99.8782% | 2.77 |
| GA | 57141 | 19 | 99.8530% | 4.16 |
| FGA | 57141 | 15 | 99.9057% | 5.31 |

**Table 9.** Classification results of features subset using NaiveBayes

| Algorithms | Sample number | Features number | Correctly classified rate | Time of build model |
|---|---|---|---|---|
| BPSO | 57141 | 14 | 96.1341% | 0.55 |
| GA | 57141 | 19 | 96.0029% | 0.83 |
| FGA | 57141 | 15 | 96.1457% | 0.76 |

## 6 Conclusion

First of all, analyzes the key issues of feature selection of intrusion detection systems and determines the feature selection mode. Based on the analysis of the basic particle swarm optimization algorithm and the discrete particle swarm optimization algorithm, Proposed an improved feature selection method BPSO-P based on discrete particle swarm optimization. Design the BPSO-P algorithm group natural number coding scheme, define the particle position, velocity and update rules, define the fitness function of evaluating the particle's advantages and disadvantages, using the correlation evaluation method, according to the correlation degree of the particle, 14 feature attributes were selected from all 41 features of KDDCUP99 data set. BPSO-P's coding method and location update rule in this paper can be used not only in the feature selection of intrusion detection, but also can process other similar discrete problems. It has certain application value.

Finally, 57141 data were extracted from the KDDCUP99 data set to construct the experimental sample space. The classification model was built using the classic classifier in the Weka platform. The ten-fold cross-validation method was used to verify the effectiveness of the algorithm. The experimental results show that BPSO-P feature selection algorithm proposed in this paper is applicable to a variety of classifiers, which can slightly improve the correct classification ratio, and the modeling time of the classifier is significantly shortened. Compared with other feature selection algorithms, BPSO-P can obtain a higher correct classification ratio by using fewer feature numbers, which is an effective feature selection method.

## References

1. Daoyuan, H., Jinghua, S.: Network security (2004)
2. Anderson, J.P.: Computer security threat monitoring and surveillance (1980)
3. Spafford, E.H.: Crisis and aftermath. Commun. ACM **32**(6), 678–687 (1989)
4. Caruana, R., Freitag, D.: Greedy attribute selection. In: ICML, pp. 28–36. Citeseer (1994)
5. Cox, I.J., Miller, M.L., Bloom, J.A., Honsinger, C.: Digital Watermarking, vol. 53. Springer, Heidelberg (2002)
6. Tataru, R. L., El Assad, S., D'eforges, O.: Improved blind DCT watermarking by using chaotic sequences. In: 2012 International Conference for Internet Technology and Secured Transactions, pp. 46–50. IEEE (2012)
7. Dash, M., Liu, H.: Feature selection for classification. Intell. Data Anal. **1**(1), 131–156 (1997)
8. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artif. Intell. **97**(1), 273–324 (1997)
9. John, G.H., Kohavi, R., Pfleger, K., et al.: Irrelevant features and the subset selection problem. In: Machine Learning: Proceedings of the Eleventh International Conference, pp. 121–129 (1994)
10. Chen, B., Hong, J., Wang, Y.: The problem of finding optimal subset of features. Chin. J. Comput.-Chin. Edn. **20**, 133–138 (1997)

11. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, New York, NY, vol. 1, pp. 39–43 (1995)
12. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 5, pp. 4104–4108. IEEE (1997)
13. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.-A.: A detailed analysis of the KDD cup 99 data set. In: Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications (2009)
14. Gong, S., Gong, X., Bi, X.: Feature selection method for network intrusion based on gqpso attribute reduction. In: 2011 International Conference on Multimedia Technology (ICMT), pp. 6365–6368. IEEE (2011)
15. Zhang, X.Q., Gu, C.H.: A method to extract network intrusion detection feature. J. South China Univ. Technol. (Nat. Sci. Edn.) **1**, 019 (2010)
16. Li, Y., Fang, B., Guo, L., Chen, Y.: Network anomaly detection based on TCM-KNN algorithm. In: Proceedings of the 2nd ACM symposium on Information, computer and communications security, pp. 13–19. ACM (2007)
17. Zhang, Y., Wang, L., Sun, W., Green, R.C., Alam, M., et al.: Distributed intrusion detection system in a multi-layer network architecture of smart grids. IEEE Trans. Smart Grid **2**(4), 796–808 (2011)
18. Wang, X., Sun, L.: Ant algorithm inspired immune intrusion detector generation algorithm. In: 2011 International Conference on Network Computing and Information Security (NCIS), vol. 2, pp. 124–127. IEEE (2011)
19. Shitao, C., Guolong, C., Wenzhong, G., Yanhua, L.: Feature selection of the intrusion detection data based on particle swarm optimization and neighborhood reduction. J. Comput. Res. Dev. **7**, 018 (2010)
20. Su, J.-R., Li, B.-Y., Wang, X.-K.: Particle swarm optimization using average information of swarm. Jisuanji Gongcheng yu Yingyong (Comput. Eng. Appl.) **43**(10), 58–59 (2007)
21. Group, W.M., et al.: The waikato environment for knowledge analysis. http://www.cs.waikato.ac.nz/ml/weka-2007
22. Xu, J., You, J., Liu, F.: A fuzzy rules based approach for performance anomaly detection. In: Proceedings of the 2005 IEEE Networking, Sensing and Control, pp. 44–48. IEEE (2005)
23. Amudha, P., Karthik, S., Sivakumari, S.: A hybrid swarm intelligence algorithm for intrusion detection using significant features. Sci. World J. (2015)
24. Singh, A., Banafar, H., Pippal, R.S.: Intrusion detection on KDD99cup dataset using K-means, PSO and GA: a review. Probe **300**, 300 (2015)
25. Lin, S.-W., Ying, K.-C., Lee, C.-Y., Lee, Z.-J.: An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Appl. Soft Comput. **12**(10), 3285–3290 (2012)