# Feature Ranking in Intrusion Detection by Hybrid Algorithm with Support Vector Machine and Analytic Hierarchy Process

[1]Dan Liu, [2]Zhonghuan Tian, [3]Bin Luo, *[4]Jingbo Xia
[1,2,3]*College of science, Huazhong Agricultural University*
*4,Corresponding Author *College of science, Huazhong Agricultural University, jingbo_xia@yahoo.cn*

## *Abstract*

*Evaluation of feature contribution and its ranking in intrusion detection data set is a hot issue for intrusion detection system (IDS) researchers. Machine learning algorithm support vector machine (SVM) and decision algorithm analytic hierarchy process (AHP) is combined to assess feature ranking in KDDcup99 data set in our experiment. It is discover that features dst_host_srv_count, dst_host_same_src_port_rate, dst_host_count, dst_host_rerror_rate, dst_host_same_srv_rate , which are representatives of features in host based network traffic statistical characteristics groups are of more importance in KDDcup99 data set than other features.*

**Keywords**: *Feature Ranking, Intrusion Detection, Support Vector Machine*

## 1. Introduction

With the development of computer science especially network technique, computer system has developed from a dedicated host to complicated and connected open system, which bring people great convenience in sharing information and resources but arise a series numbers of security problems .

Network intrusion aims to destroy network system's confidentiality, integrity and availability, and it is hard to be dealt with by traditional firewall.. Under this circumstance, intrusion detection system (IDS) is increasingly becoming the focus of security market. IDS, by analyzing the connection records in data packets, can precisely distinguish intrusion affairs. By now, IDS has two common models: The first model is anomaly detection model. Anomaly detection assumes intrusion activity is different from system's normal activity, by constructing normal activity database and comparing unknown activities, the suspicious activity and affairs could be found. While the second model is misuse detection or signature-based detection, which aims to collect possible intrusion affairs and activities. By analyzing and the detected activities, IDS could distinguish whether these activities are intrusion or not. However, misuse detection could only distinguish intrusion activity base on known intrusion affairs, hence it is unable to recognize unknown intrusion activities. Currently, anomaly detection model based IDS is mainly used.

Moreover, IDS has to process and analyze massive data, always with size over thousands M bytes. Therefore, to reduce the dimension of data set and to extract some representative features from the data set to decrease the computing scale and to improve the efficiency and prediction accuracy of IDS have become the focus of IDS researchers. Many data-mining based intelligent algorithm, such as k-nearest neighbor (KNN), support vector machine (SVM), artificial neural networks (ANN), self-organizing maps (SOM), decision tress, genetic algorithm (GA), fuzzy logic [1], have been introduced into IDS in order to achieve improvement and optimization.

For the feature selection of IDS, there are single feature selection method such as feature removal method and sole feature method, as well as hybrid method such as gradually feature removal (GFR) method [2], Bi-Layer behavioral-based feature selection approach [3], gravitational search algorithm (GSA) [4], MOGFID method [5]. Some researchers assess the importance between each feature and choose some representative features to represent the whole data set while some reduce the weak relevant features to reduce the data set and computing scale [6-8]. For example, Yinhui Li [2] extract 19 important features in KDDcup99 data set, Heba F. Eid [3] decrease the 41 features in KDDcup99 to 20 to represent the whole data set. Mansour Sheikhan [4] selected 26 features in KDDcup99 data set while Tsang [5] chose 25 important features.

The purpose of this paper is to put forward a novel and efficient hybrid algorithm of SVM and AHP so as to assess the different feature's ranking in KDDcup99 data set. We find the 41 features in KDDcup99 data set are of different ranking, features dst_host_srv_count, dst_host_same_src_port_rate,

dst_host_count, dst_host_rerror_rate, dst_host_same_srv_rate, which are representatives of features in host based network traffic statistical characteristics groups are of more importance in KDDcup99 data set than other features.

This paper is organized as follow. Section 2 introduces materials and methods used in our experiment. Experiment results are shown in section 3. Conclusion and discussion and future work are provided in section 4.

## 2. Materials and methods

### 2.1 Data set & Data processing

#### 2.1.1 Data set

The KDDcup99 data set (downloaded from http://www.sigkdd.org/kddcup/index.php? section=1999 &method=data), which includes nine weeks of raw TCP data simulation of US Air Force environment, is regarded as standard data for IDS researchers. This data set has two versions, namely, the full data set (4898431 records, 743M uncompressed) and the 10% subset (494307 records, 75M uncompressed). For efficiency in program implementation, the 10% subset is used in our experiment.

For every connection record in KDDcup99 data set, there are 41 features, which are classified into 4 categories, basic features with of individual TCP connections (feature 1 to 9), content features within a connection suggested by domain knowledge (feature 10 to 22), time based network traffic statistical characteristics (feature 23 to 31), and host based network traffic statistical characteristics (feature 32 to 41). See figure 1. The whole records in KDDcup99 data set is distributed into 5 classes which are normal, denial of service (DOS), unauthorized access from a remote machine (R2L), unauthorized access to local supervisor privileges (U2R) and probing, surveillance and other probing according to their visit behaviors.

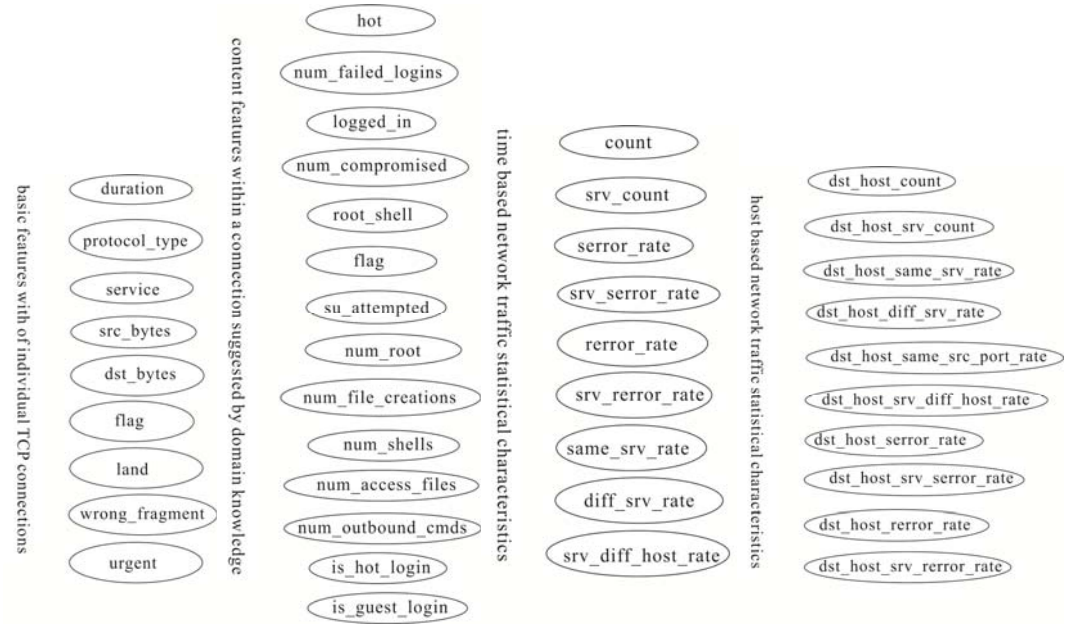


**Figure 1:** features categories in KDDcup99 data set

#### 2.1.2 Data processing

There are 70.5% repetitions in the 10% KDDcup99 data set which is redundancy of our experiment, so we delete the repetition of the data set and reduce the quantity of the data set from 494307 to 145831. For each class of the data set, the quantity changes are shown in table 1.

**Table 1:** 10%_KDDcup99 data set's construction and experimental data set construction

| Class | Normal | DOS | Probe | R2L | U2R |
|---|---|---|---|---|---|
| 10%_KDDcup99 data set | 97564 | 391458 | 4107 | 1126 | 52 |
| Experimental data set | 88089 | 54562 | 2129 | 999 | 52 |

Taking the data set size into consideration, part of the reduced data set is chosen to make up the final data set for SVM. Noticing the huge number of class Normal and DOS, we choose 10% data of class Normal and DOS by interval sampling. Meanwhile, data of Probe, R2L, U2R are fully remained. In this way, we get the final data set whose total number is 17444.

KDDcup99 data set has 41 features, in which No. 2, 3 and 4 is character. For each character, distributed by upper case and lower case, the difference between each letter and upper case A or lower case a is calculated, summarize all the differences and the character feature is transformed into number. For example, the feature "abcd" will be transformed into 6. In addition, all data are scaled into [0, 1].

## 2.2 Algorithm of SVM and AHP

### 2.2.1 SVM classifier

SVM is a popular-used classification technique, which constructs hyper plains in a higher or infinite-dimensional space, which can be used for classification, regression or other tasks. SVM tries to produce a model, which is based on training data provided and kernel functions, to predict target data's values only given test data's attributes. The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.

When a training data set of instance-label pairs $(x_i, y_i)$, $i = 1, 2, \cdots\cdots n$, where $x_i \in R^m$ and $y_i \in \{1, -1\}$ is given, SVM aims to find the solution of the following optimization problem:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i (w^T \varphi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned}$$

In the above formula, function $\varphi$ map the training vectors $x_i$ into a higher dimensional space in which SVM will find a linear separating hyper plain with the maximal margin and realize the classification. $C > 0$ is called the penalty parameter of the error term, which aims to avoid over-fitting. In addition, $K(x_i, x_j) \equiv \varphi(x_i)^T \varphi(x_j)$ is the kernel function, including linear kernel $K(x_i, x_j) = x_i^T x_j$, polynomial kernel $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$, radial basis function (RBF) kernel $K(x_i, x_j) = \exp(-\gamma \left\| x_i^T x_j \right\|^2)$, and sigmoid kernel $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$, where $\gamma > 0,\ d, r$ are parameters.

### 2.2.2 AHP algorithm

The Analytic Hierarchy Process (AHP) was proposed by Thomas Saaty in the 1970. By organizing and analyzing complex decisions, which aims to rank the importance of elements in decision system.. The procedure of the normal AHP algorithm can be summarized as:

Step 1. Model the problem as a hierarchical structure containing the decision goal, the alternatives for reaching it, and the criteria for evaluating the importance of alternatives.

Step 2. Obtain the pair wise comparison matrix. Priorities among the elements of the hierarchy is established by making a series of judgments based on pair wise comparisons of the elements. Denote

$a_{ij}$ as the comparison value of element $B_i$ and $B_j$, the matrix $A=\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$ is defined as the Pair wise comparison matrix. All elements in matrix A have these features: (1). $a_{ij} > 0$ (2). $a_{ij} = \frac{1}{a_{ji}}, \quad (i, j = 1, 2, .....n)$.

Step 3. Check the consistency of the comparison matrix. Suppose the largest eigenvalue of the matrix *A* is $\lambda_{\max}$.

(1). Calculate the CI (Consistency index) of the matrix *A*, where $CI = \frac{\lambda_{\max} - n}{n-1}$,

(2). Search for the appropriate RI (Average random consistency index) of the matrix *A*.

For $n = 1, 2, .....9$, RI is given as:

| *n* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| RI | 0 | 0 | 0.58 | 0.90 | 1.12 | 1.24 | 1.32 | 1.41 | 1.45 |

For $n > 9$, RI could be calculated like this: Construct 500 Pair wise comparison matrixes randomly, and get the mean largest eigenvalue of the 500 matrixes, $\lambda'_{\max}$. $RI = \frac{\lambda'_{\max} - n}{n-1}$.

(3). Calculate the CR (Consistency ratio) of the matrix A. $CR = \frac{CI}{RI}$. Only in the case that CR<0.10, the matrix A is acceptable.

Step 4. Plus all elements of each row, we can get the $A_1, A_2 \cdots A_n$ represent the summation of each row. Calculate $\frac{A_i}{A_1 + A_2 + \cdots\cdots + A_n}, (i = 1, 2, \cdots\cdots n)$, we can get the weight $w_1, w_2 \cdots w_n$ for each element. The bigger the weight is, the more important the element is.

## 2.3 Proposed hybrid strategy of SVM and AHP in computing feature's weight

### 2.3.1 Hybrid algorithm

Step 1. Assume *j*=1, *j* represent which feature is being computing. Assume $\{x_1, x_2 \cdots\cdots x_{41}\}$ represent the 41 features of the data set.

Step 2. Delete $x_j$ in training data set and test data set, and find the best *C* and $\gamma$ of the 40 features data set for SVM classifier.

Step 3. Train SVM classifier and calculate the prediction accuracy with *C* and $\gamma$, and assume the deleted-data accuracy is $accu_j$, *j*++.

Step 4. Repeat step 2 and 3, until *j*=41.

Step 5. Let $a_{ij} = \frac{accu_j}{accu_i}$, $(i, j = 1, 2, \cdots\cdots 41)$, the pair wise comparison matrix in AHP is calculated as $\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$.

Step 6. Follow AHP algorithm and get the weight $w_i$ $(i = 1, 2, \cdots\cdots 41)$ for each feature-deleted data set.

Inspired by priorities in AHP, we let $a_{ij} = \frac{accu_j}{accu_i}$ and define the pair wise comparison matrix. The reason for this definition is that $accu_i$ refers to the prediction accuracy by SVM without feature *i*, while $accu_j$ refers to the prediction accuracy by SVM without feature *j*, when one feature is poor-relevant to the data set and prediction, the corresponding accuracy will be higher than those tight-relevant features. In addition, the comparison matrix is of strong consistency.

#### 2.3.2 Parameter searching method for *C* and $\gamma$ in SVM

RBF kernel function is chosen as the kernel function in our experiment and the main steps of SVM implementation is as following: scale the data set, choose the best parameter *C* and $\gamma$, train the SVM classifier, and finally use the trained SVM classifier to classify the test data.

The choice for parameter *C* and $\gamma$ is of great importance on prediction accuracy for SVM classifier. K-fold cross validation (K-CV) method is used to optimize the SVM classifier and get the best parameter *C* and $\gamma$. K-CV method used in optimization of SVM classifier means equally divide raw data set into k groups and for each subset, it will be used as test data set for SVM classifier while the rest k-1 groups are used as SVM rain data set. Consequently, k prediction accuracy will be gotten and the average accuracy would be classification symbol for the SVM classifier. It is straight forward that the *C* and $\gamma$ combination towards the highest average prediction accuracy among k groups is the best parameter combination for SVM classifier.

However, when more than one groups of combination has the same average prediction, the group which has smallest *C* should be chosen because parameter *C* is a penalty parameter for error term, a big *C* would lead to over fitting phenomenon and decrease the generalization of the SVM classifier. When the smallest *C* combines with several groups of $\gamma$, the first searched group of *C* and $\gamma$ is chosen as the best parameter for SVM classifier.

To decrease the scale of calculation, a wide range such as $\{2^{-10}, 2^{-9}, \cdots\cdots 2^{10}\}$ is set for roughly search *C* and $\gamma$, after getting the best *C* and $\gamma$, a narrow range such as $\{2^{-4}, 2^{-3.5}, \cdots\cdots, 2^{2.5}, 2^{3}\}$ is set accordingly and the searched *C* and $\gamma$ in this round will be seen as the best parameters for SVM classifiers.

## 3. Results

As discussed in Material and methods, the experimental data set consist of 10% data in normal/DOS class, and the whole data in class R2L, U2R, Probe. Here, sample pair (i, j) is used to describe which connection record in class normal and DOS is chosen. In class normal the connection record whose number is $10n + i$, $i = 1, 2\cdots\cdots$ is chosen while in class DOS record whose number is $10n + j$, $j = 1, 2\cdots\cdots$ is chosen. For each class of data set, the former 70% is chosen as SVM train data set and the rest 30% is chosen as the SVM test data set. In the experiment, 9 groups of sample pairs are chosen, which are (5, 7), (1, 9), (2, 2), (3, 3), (4, 4), (6, 6), (8, 8), (7, 1), (9, 5). The average result for 9

groups could be seen in table X. Feature listed in order of weight is: 33, 36, 32, 40, 34, 14, 30, 22, 10, 18, 29, 26, 27, 19, 3, 15, 5, 8, 9, 13, 21, 22, 6, 16, 11, 7, 1, 4, 17, 24, 25, 28, 38, 23, 39, 31, 41, 35, 37, 12, 2. Features' weight, ranking and feature-deleted prediction accuracy could be seen in table 2 and 3.

**Table 2:** Feature weight and ranking

| Feature name | Feature number | Weight | Ranking | Feature name | Feature number | Weight | Ranking |
|---|---|---|---|---|---|---|---|
| dst_host_srv_count | 33 | 0.02433 | 1 | is_hot_login | 21 | 0.02181 | 22 |
| dst_host_same_src_port_rate | 36 | 0.02336 | 2 | flag | 6 | 0.02181 | 23 |
| dst_host_count | 32 | 0.02329 | 3 | num_root | 16 | 0.02181 | 24 |
| dst_host_rerror_rate | 40 | 0.02256 | 4 | num_failed_logins | 11 | 0.02181 | 25 |
| dst_host_same_srv_rate | 34 | 0.02227 | 5 | land | 7 | 0.02181 | 26 |
| root_shell | 14 | 0.02214 | 6 | duration | 1 | 0.02181 | 27 |
| diff_srv_rate | 30 | 0.02209 | 7 | src_bytes | 4 | 0.02181 | 28 |
| is_guest_login | 22 | 0.02203 | 8 | num_file_creations | 17 | 0.02180 | 29 |
| hot | 10 | 0.02191 | 9 | srv_count | 24 | 0.02179 | 30 |
| num_shells | 18 | 0.02189 | 10 | serror_rate | 25 | 0.02178 | 31 |
| same_srv_rate | 29 | 0.02186 | 11 | srv_rerror_rate | 28 | 0.02177 | 32 |
| srv_serror_rate | 26 | 0.02184 | 12 | dst_host_serror_rate | 38 | 0.02176 | 33 |
| rerror_rate | 27 | 0.02183 | 13 | count | 23 | 0.02174 | 34 |
| num_access_files | 19 | 0.02182 | 14 | dst_host_srv_serror_rate | 39 | 0.02173 | 35 |
| service | 3 | 0.02181 | 15 | srv_diff_host_rate | 31 | 0.02172 | 36 |
| su_attempted | 15 | 0.02181 | 16 | dst_host_srv_rerror_rate | 41 | 0.02171 | 37 |
| dst_bytes | 5 | 0.02181 | 17 | dst_host_diff_srv_rate | 35 | 0.02167 | 38 |
| wrong_fragment | 8 | 0.02181 | 18 | dst_host_srv_diff_host_rate | 37 | 0.02158 | 39 |
| urgent | 9 | 0.02181 | 19 | logged_in | 12 | 0.02150 | 40 |
| num_compromised | 13 | 0.02181 | 20 | protocol_type | 2 | 0.02087 | 41 |
| num_outbound_cmds | 20 | 0.02181 | 21 | | | | |

**Table 3:** average feature-deleted prediction accuracy

| Feature-deleted | Prediction accuracy | Feature-deleted | Prediction accuracy |
|---|---|---|---|
| 1 | 80.3587 | 22 | 79.5564 |
| 2 | 83.9733 | 23 | 80.6261 |
| 3 | 80.3502 | 24 | 80.4245 |
| 4 | 80.3672 | 25 | 80.4754 |
| 5 | 80.3545 | 26 | 80.2292 |
| 6 | 80.3566 | 27 | 80.2844 |
| 7 | 80.3587 | 28 | 80.4860 |
| 8 | 80.3545 | 29 | 80.1931 |
| 9 | 80.3545 | 30 | 79.3442 |
| 10 | 79.9745 | 31 | 80.6834 |
| 11 | 80.3587 | 32 | 75.2754 |
| 12 | 81.5006 | 33 | 72.1426 |
| 13 | 80.3545 | 34 | 78.7138 |
| 14 | 79.1892 | 35 | 80.8596 |
| 15 | 80.3523 | 36 | 75.0504 |
| 16 | 80.3566 | 37 | 81.2501 |
| 17 | 80.3884 | 38 | 80.5476 |
| 18 | 80.0743 | 39 | 80.6431 |
| 19 | 80.3205 | 40 | 77.7820 |
| 20 | 80.3545 | 41 | 80.7131 |
| 21 | 80.3545 | none | 80.3629 |

The SVM parameter searching results could be seen in figure 2 and 3.

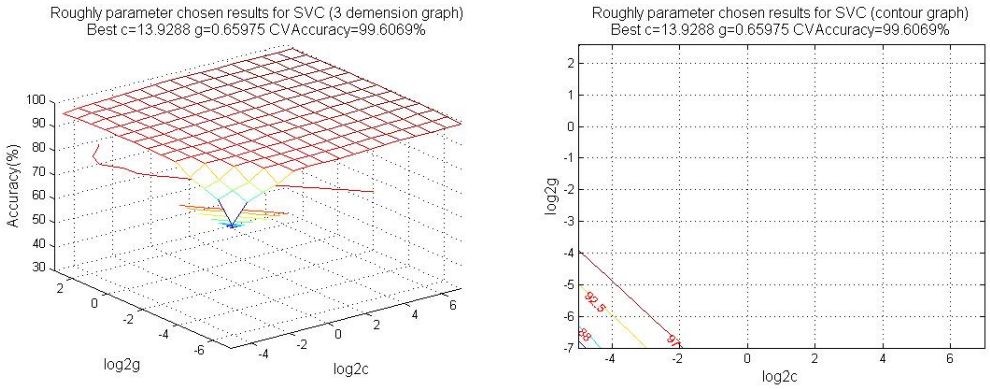


**Figure 2:** Parameter searching results for the 7th sample pairs

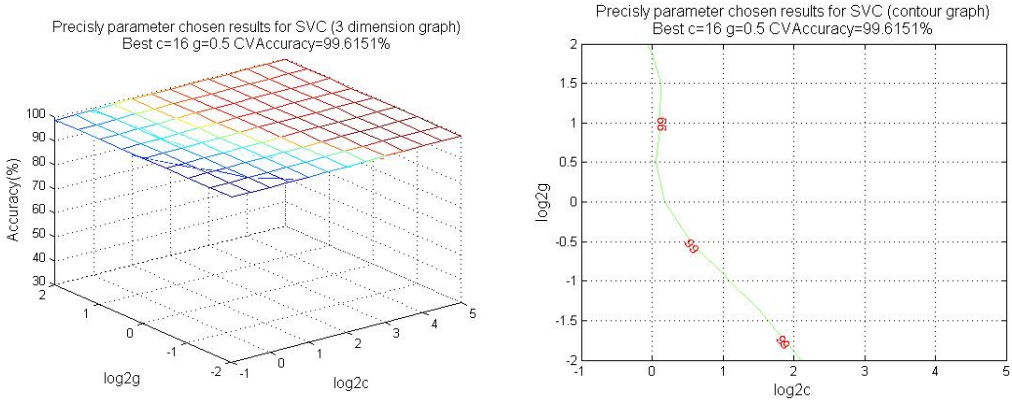


**Figure 3:** precisely parameter chosen results for the 7th sample pairs

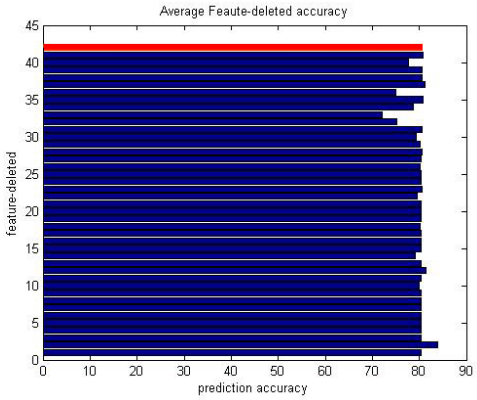


**Figure 4:** average feature-deleted accuracy

Feature-deleted average prediction accuracy could be seen in figure 4. The red bar in figure 4 refers no feature-deleted experimental data set's prediction accuracy.

## 4. Conclusion and discussion

By analyzing the experimental data set, it is found that all connection record in feature 21st (is_hot_login) and 22nd (is_guest_login) is 0.00. This phenomenon means that for the 5 classes, the connection record is the same which means feature 21st and 22nd contribute nothing to the classification, so feature 21st and 22nd is called the standard features. With feature 21st and 22nd, 41

features in figure 1 could be classified into two group, the former 20 features are strong-relevant or important features, while other 19 features are not, which could be seen in table 4.

**Table 4:** classification of results

| Feature category | Feature numbers | Feature list |
|---|---|---|
| Important | 20 | 33, 36, 32, 40, 34, 14, 30, 22, 10, 18, 29, 26, 27, 19, 3, 15, 5, 8, 9, 13 |
| Standard | 2 | 21, 22 |
| Others | 19 | 6, 16, 11, 7, 1, 4, 17, 24, 25, 28, 38, 23, 39, 31, 41, 35, 37, 12,2 |

Among the important features, the category basic features with of individual TCP connections has 4, category content features within a connection suggested by domain knowledge has 6, category time based network traffic statistical characteristics has 4, category host based network traffic statistical characteristics has 6. It is worth mentioning that the most important 5 features, which are 33$^{rd}$, 36$^{th}$, 32$^{nd}$, 40$^{th}$ and 34$^{th}$, are all exist in category host based network traffic statistical characteristics

In order to evaluate the results, we compared our results with other researchers' experimental results, which are feature removal method, sole feature method, hybrid method, GFR method, MOGFIDS and GSA, comparison result could be seen in table 5.

**Table 5:** result comparison with other algorithms

| Algorithm | Feature number | Important feature list |
|---|---|---|
| Feature removal [2] | 10 | 8, 10, 14, 31, 32, 33, 35, 36, 37, 40 |
| Sole feature [2] | 10 | 6, 7, 23, 24, 25, 29, 30, 31, 32, 38 |
| Hybrid method [2] | 10 | 10, 14, 23, 24, 25, 31, 32, 33, 36, 38 |
| GFR method [3] | 19 | 35, 33, 2, 14, 36, 10, 4, 32, 40, 29, 8, 37, 31, 38, 34, 25, 27, 15, 19 |
| MOGFIDS [4] | 25 | 2, 5, 6, 7, 8, 9, 11, 12, 13, 14, 17, 22, 23, 25, 30, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 |
| GSA [5] | 26 | 1, 3, 5, 6, 10, 13, 14, 16, 17, 18, 19, 23, 24, 25, 26, 27, 28, 29, 31, 35, 36, 37, 38, 39, 40, 41 26 |
| Our algorithm | 20 | 33, 36, 32, 40, 34, 14, 30, 22, 10, 18, 29, 26, 27, 19, 3, 15, 5, 8, 9, 13 |

Compared with the former 10 features in our experiment, feature removal method, sole feature method, hybrid method have 6, 4, 5 repetition, respectively. Compared with GFR method, MOGFIDS and GSA, 13, 12 and 12 features are repeated, respectively.

Among the feature list, especially the artificial intelligent data mining algorithm, there are some common features, feature 14$^{th}$, 36$^{th}$, 40$^{th}$ appears in all four former algorithms while feature 5$^{th}$, 8$^{th}$, 10$^{th}$, 13$^{th}$, 19$^{th}$, 25$^{th}$, 26$^{th}$, 27$^{th}$, 29$^{th}$, 32$^{nd}$, 33$^{rd}$, 34$^{th}$, 35$^{th}$, 37$^{th}$, 38$^{th}$ appear 3 times in four algorithms.

In this paper, a new feature selection algorithm, SVM plus AHP is put forward to deal with the KDDcup99 data set. By calculating features' weight and ranking, 2 standard features and 19 important features are found in KDDcup99 data set. Features 33$^{rd}$, 36$^{th}$, 32$^{nd}$, 40$^{th}$ and 34$^{th}$, existed in category host based network traffic statistical characteristics are the most important five features in the experiment, which indicates that the category of host based network traffic statistical characteristics are worthy of more focus in intrusion detection.

To improve feature-deletion process, trying more numbers of features deleted at one time and finding a group of features representing the whole 41 features to predict intrusion will be considered in our future work.

## Acknowledgements

This research is partly supported by National Natural Science Foundation of China (Grant No. 61202305 ) and National Creative Innovation Plan of College Students (Grant No. 201210504090).

## References

[1] Chif-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, Wei-Yang Lin , "Intrusion detection by machine learning: A review", Experts systems with Applications, Vol. 36, pp. 11994–12000, 2009.

[2] Yinhui Li, Jingbo Xia, Silan Zhang, "An efficient intrusion detection system based on support vector machine and gradually feature removal method", Experts systems with Applications, Vol. 39, pp. 424-430, 2012.

[3] Heba F.Eid, Mostafa A. Salama, Aboul Ella Hassanien, Tai-hoon Kim, "Bi-Layer Behavioral-Based Feature Selection Approach for Network Intrusion Classification", Security Technology, Communications in Computer and Information Science, Vol. 259, pp. 195-203, 2011.

[4] Mansour Sheikhan, Maryam Sharifi Rad, "Gravitational search algorithm–optimized neural misuse detector with selected features by fuzzy grids–based association rules mining", Neural Computing and Applications, DOI 10.1007/s00521-012-1204-y, 2012.

[5] Chi-Ho Tsang, Sam Kwong, Hanli Wang, "Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection", Pattern recognition, Vol. 40, pp. 2373-2391, 2007.

[6] Shafi, K., & Abbass, H. A. "An adaptive genetic-based signature learning system for intrusion detection", Expert Systems with Applications, 36(10), 12036–12043, 2009.

[7] Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. "Intrusion detection by machine learning: A review", Expert Systems with Applications, 36, 11994–12000, 2009.

[8] Tsai, C. F., & Lin, C. Y. "A triangle area based nearest neighbors approach to intrusion detection", Pattern Recognition, 43(1), 222–229, 2010.