Expert Systems with Applications 39 (2012) 424–430

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An efficient intrusion detection system based on support vector machines and gradually feature removal method

Yinhui Li [a], Jingbo Xia [a,*], Silan Zhang [a], Jiakai Yan [a], Xiaochuan Ai [b], Kuobin Dai [c]

[a] College of Science, Huazhong Agricultural University, Wuhan, Hubei, China
[b] College of Science, Navy Engineering University, Wuhan, Hubei, China
[c] College of Math and Info Science, Huanggang Normal University, Huanggang, Hubei, China

ARTICLE INFO

Keywords:
Intrusion detection
Support vector machine
Feature reduction

ABSTRACT

The efficiency of the intrusion detection is mainly depended on the dimension of data features. By using the gradually feature removal method, 19 critical features are chosen to represent for the various network visit. With the combination of clustering method, ant colony algorithm and support vector machine (SVM), an efficient and reliable classifier is developed to judge a network visit to be normal or not. Moreover, the accuracy achieves 98.6249% in 10-fold cross validation and the average Matthews correlation coefficient (MCC) achieves 0.861161.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

With the rapid development and popularity of Internet, the security of networks has been a focus in the current research. Nowadays, much attention has been paid to intrusion detection system (IDS) which is closely linked to the safe use of network services. However, it is not easy to discern the attack and the normal network visit. To overcome this problem, various artificial intelligence methods are developed, such as fuzzy logic (Chimphlee, Addullah, Sap, Srinoy, & Chimphlee, 2006; Tsang, Kwong, & Wang, 2007), *K*-nearest neighbor (Li & Guo, 2007; Tsai & Lin, 2010), support vector machine, SVM (Joseph, Das, Lee, & Seet, 2010; Khan, Awad, & Thuraisingham, 2007), artificial neural networks, ANN (Wang, Hao, Ma, & Huang, 2010), Naïve Bayes networks (Amor, Benferhat, & Elouedi, 2004), principal component analysis, PCA (Wang & Battiti, 2006), decision tree (Depren, Topallar, Anarim, & Ciliz, 2005; Xiang, Yong, & Meng, 2008) and genetic algorithm, GA (Mukkamala, Sung, & Abraham, 2004; Shafi & Abbass, 2009). For explicit review of current development, refer to the recent review (Tsai, Hsu, Lin, & Lin, 2009; Wu & Banzhaf, 2010).

Among the methods mentioned above, SVM is an effective one, which is a well-known classifier tool based on small sample learning. Since SVM has manifested its robustness and efficiency in the network action classification, it therefore becomes a popular method widely used in IDS, as shown in Tsai's review (Tsai et al., 2009).

In general, IDS deals with tremendous amount of data which contain redundant and irrelevant features causing excessive training and predicting time. Methods for feature reduction are divided into two categories: filter method and wrapper method (Kohavi & John, 1997). Wrapper method is kind of feature removal method by evaluating the resultant probability of error, to select the critical features. Meanwhile, filter methods analyze the sole features independent of the classifier and decide which features should be kept. Generally, wrapper methods generally perform better than filter methods.

In order to get better performance, hybrid feature selection method is also considered, which combines wrapper and filter methods. Unfortunately, the performance of the hybrid methods is far from perfect. On the other hand, due to the increasing application of artificial intelligence, various machine learning methods merged into the research of feature reduction in IDS. Chebrolu, Abraham, and Thomas (2005) investigated the performance of two feature selection algorithms involving Bayesian networks (BN) and Classification Regression Trees (CRC), and developed the ensemble of both methods. Furthermore, Tsang et al. (2007) used genetic-fuzzy rule mining approach to evaluate the importance of IDS features.

Recently, Li, Wang, Tian, Lu, and Young (2009) proposed a wrapper-based feature selection algorithm aiming at building lightweight IDS by using a modified random mutation hill climbing as search strategy to specify a candidate subset for evaluation. The illuminative sense of this research is the confirmation of the effectiveness of traditional wrapper and filter method with proper feature selection strategy. The focus of this paper is this area. Our approach in selecting critical features is an improved wrapper-based feature reduction method, called gradually feature removal method, short for GFR method.

The aim of this research is twofold: the first is to establish a desirable IDS model with high efficiency and accuracy, by formulating a pipeline of data processing and data mining based

* Corresponding author.
E-mail addresses: xjb@mail.hzau.edu.cn, jingbo_xia@yahoo.cn (J. Xia).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.07.032

on certain machine learning method, including *k*-means algorithm for data clustering, ant colony optimization (ACO) method for small training data set construction and SVM for classifier, etc.; the second is to develop the feature reduction method, GFR method, and select critical features in IDS, so as to reduce the training and predicting time in IDS classifier with the least sacrifice of accuracy.

In this paper, we put forward a pipeline of the data preprocess and data mining in IDS, and also give an effective strategy in feature reduction, GFR method. The experimental result shows the efficiency and reliable of this IDS pipeline. Moreover, 19 most important features related to the intrusion are discovered.

2. Materials and methods

2.1. Data set

The experimental data used in this paper is a benchmark database downloaded from KDDcup99 (http://kdd.ics.uci.edu/databases/kddcup99). This database contains a standard set of network visit data, which includes a wide variety of intrusion simulation in the US military network environment. KDDcup99 data consist of two data sets, which are the full data set (18 M, 743 M Uncompressed) and the 10% subset (2.1 M, 75 M Uncompressed). The latter is chosen to be the experimental data set as our object. Each data consists of 41 features, as listed in Fig. 1.

The classification of the attack behavior is a 5-class problem, and each network visit belongs to one of the following behavior: normal, denial of service (DOS), unauthorized access from a remote machine (R2L), unauthorized access to local supervisor privileges (U2R), probing, surveillance and other probing.

2.2. SVM classifier

Due to the popularity of support vector machine in the current research, we briefly introduce SVM in this section. SVM is an efficient tool widely used in the multiclass classification. By computing the hyper plain of a given set of training samples, a support vector machine builds up a mechanism to predict which category a new sample falls into.

The parameter need to be checked is γ and C, where γ is used in the kernel function, radial basis function (RBF): $K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$, and C is the penalty parameter for the model.

For the RBF kernel function, the learning model equals to solve the following convex quadratic programming (QP) problem,

$$\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to:

$$0 \leqslant \alpha_i \leqslant C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0,$$

where labels $y_i = +1, -1$ stands for the positive label and negative label, respectively. The explicit information could refer to Vapnik (1998).

2.3. Data preprocessing

One notes that the redundancy in the KDD99 data set is amazingly high. Obviously, such a high redundancy certainly influences the use of data. By deleting the repeated data, the size of data set is reduced from 494,021 to 145,586.

Furthermore, in order to make the data set more efficient, *K*-means clustering (Ball & Hall, 1967) is used to reduce the data set. *K*-means is a popular clustering algorithm which aims to partition different data samples into certain clusters by evaluating the smallest distance between data and clusters. First, k samples are chosen to present for different cluster. Second, each sample is assigned into the nearest cluster. Afterwards, update the central of the cluster. Repeat the second step until all of sample data are partitioned into a suitable cluster.

By clustering the data into 5 clusters, the intersection of the original data and clustered data are remained. The size of data set is reduced from 145,586 to 116,266. The final data set is named as compact data set. In detail, the process of the data pretreatment is illustrated in Fig. 2.

2.4. Construction of small training data set

After streamlining the raw data, a small sample data set need to be chosen from the database so as to represent for the whole data set. In order to test the robustness of small training data, the SVM tool is used, which is also a classifier depending on small sample learning.

Here, an effective training data set is a subset of the original data set, in which the samples own the ability of enough representation

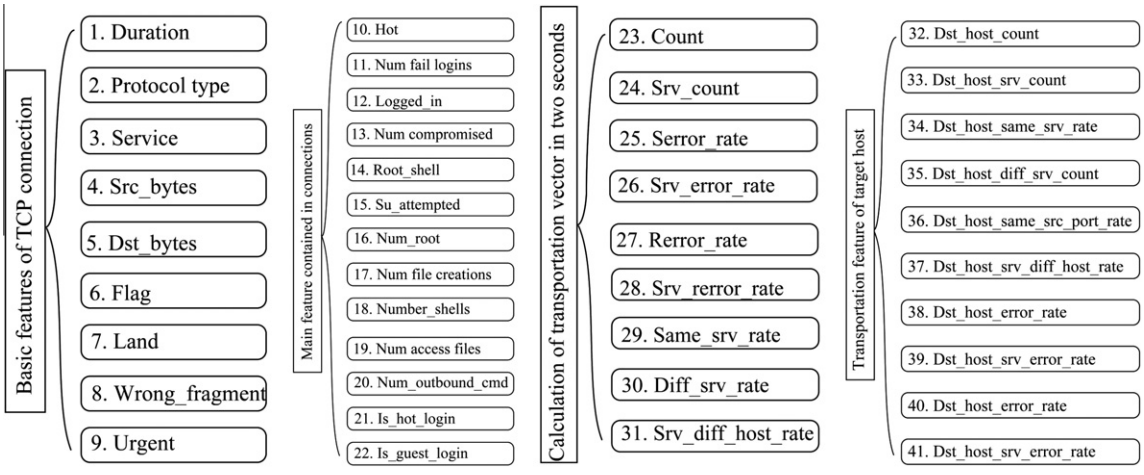

Fig. 1. Features of KDD99 data with 41 dimensions.

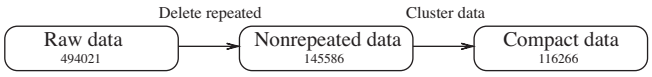

Fig. 2. The pretreatment of sample data.

for the whole data set. In common, randomly separation of small subset could act as the training data set in the process of classifier training phase. However, the effectiveness of random chosen training data set is unclear, which in turn influence the efficiency of classifier, especially in the case of large scale unknown prediction. Instead of randomly choosing strategy, ACO method is used to choose a proper one subset as the training data set.

In the ACO algorithm, each ant maps to a choice of training data set. For 116255 samples in the compact data set, 550 ones are chosen to construct the training data set. Accordingly, the chances of subset chosen amount to C_{116255}^{550}.

First, ten ants $Ant_i^{(0)}$, $(i = 1, 2, \ldots, 10)$ randomly chooses a subset among C_{116255}^{550} cases. The effectiveness of each ant is evaluated via SVM classifier. From the value of prediction accuracy, ants are sorted from the "elitist" ant to the "worst" ant, $Ant_1^{(1)}, \ldots, Ant_{10}^{(1)}$.

Afterwards, a random threshold value Tv_i^k is generated to each ant $Ant_i^{(k)}$, where $0 \leqslant Tv_i^k \leqslant 1$. According to different threshold value, $Ant_i^{(k)}$ make different choice. In detail,

$$Ant_i^{(k)} = \begin{cases} Ant_i^{(k-1)}, & \text{if } i = 10, \quad \text{or } i \neq 10 \quad \text{and} \quad -\frac{3}{90}i + \frac{1}{2} \leqslant Tv_i^k \leqslant \frac{3}{90}i + \frac{1}{2}; \\ Ant_{random}, & \text{if } i \neq 10 \quad \text{and} \quad Tv_i^k < -\frac{3}{90}i + \frac{1}{2}; \\ Ant_{neighbor}, & \text{if } i \neq 10 \quad \text{and} \quad Tv_i^k > \frac{3}{90}i + \frac{1}{2}. \end{cases}$$

As illustrated in Fig. 3, the hatched region is encircled by 3 functions, l_1, l_2 and l_3, where

$$l_1 : y = f_1(x) = \begin{cases} -\frac{3}{90}x + \frac{1}{2}, & \text{if } 0 \leqslant x < 9; \\ 0, & \text{if } 9 \leqslant x < 10. \end{cases}$$

$$l_2 : y = f_1(x) = \begin{cases} \frac{3}{90}x + \frac{1}{2}, & \text{if } 0 \leqslant x < 9; \\ 1, & \text{if } 9 \leqslant x < 10. \end{cases}$$

$$l_3 : x = 10.$$

As shown in Fig. 3, the hatched region represents for the elitist learning choice, and the upper region and the lower region stands for self-learning region and variation region, respectively. One notes that $Ant_{10}^{(k)}$ will always be $Ant_1^{(k-1)}$, which means the worst ant turns to be a better ant. By doing this, the effectiveness of the ants colony will not decrease.

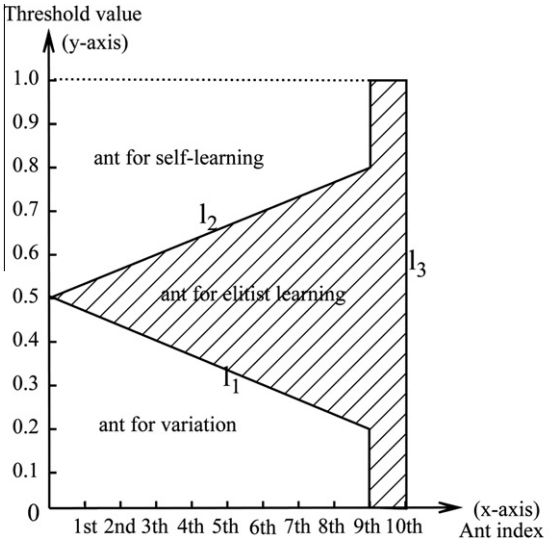

Fig. 3. The threshold value region for three ant move strategies.

After the ACO process, a proper training dataset is derived, which compromise 550 samples with 350 normal visit samples, 70 samples of DOS, 50 samples of R2L, 10 samples of U2R and 70 probing samples.

2.5. Feature reduction strategy

As depicted in 2.1, each network visit behavior in KDD99 database maps to a mathematical vector with 41 features. For the sake of efficiency, streamline the mathematical vector is vital for machine learning method in IDS. Accordingly, feature selection is done with four different strategies, the feature removal method, the sole feature method, the hybrid method and the proposed GFR method.

Algorithm 2.5.1 (*Feature removal method*).

Step 1: Assume $X = \{x_1, x_2, \ldots, x_{41}\}$, stands for the mathematical feature of the sample data.
Step 2: Delete x_i $(i = 1, 2, \ldots, 41)$ from X and update $X^{(i)} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{41})$ as the new feature vector.
Step 3: A classifier is undertaken to evaluate the importance of the provided feature x_i.
Step 4: Sort the classifier accuracy related to $X^{(i)}$, the order of the vital feature is obtained.

Algorithm 2.5.2 (*Sole feature method*).

Step 1: Assume $X = \{x_1, x_2, \ldots, x_{41}\}$, stands for the mathematical feature of the sample data.
Step 2: Assume $X^{(i)} = (x_i)$ as the one dimension feature vector.
Step 3: Classifier is constructed to evaluate the importance of x_i again.
Step 4: The order of the vital feature is obtained via the sort of the classification accuracy.

Algorithm 2.5.3 (*Hybrid method for feature selection*).

Step 1: Run 2.5.1, get features $x_{j1}, x_{j2}, \ldots, x_{j41}$, sorted by importance.
Step 2: Run 2.5.2, get features $x_{k1}, x_{k2}, \ldots, x_{k41}$, sorted by importance.
Step 3: $l = 1$. Choose l common features in the intersection with the utmost importance.
Step 4: l++. Repeat step 3, until a proper subset of $\{x_i\}$ is chosen.

Direct observation shows that the idea of the sole feature method (2.5.2) is based on the reverse idea of the feature removal method (2.5.1). From the point view of statistics, the intersection of the features in above both methods is more reliable than mere one method. Henceforth, a hybrid method is formed by using the same features with high importance order both in the feature removal method and the sole feature method. In detail, the hybrid method is described as 2.5.3.

Unfortunately, the hybrid method does not work well, as shown in the experimental result in the next section. It may stem from the low accuracy of the sole feature method. Instead, a gradually feature removal method, GFR method, is proposed.

Algorithm 2.5.4 (*Gradually feature removal method, GFR method*).

Step 1: Let $N = 41$, $j = 1$, where N denote as the dimension of the feature scale, j is used to count the chosen critical feature.
Step 2: Assume $X = (x_1, x_2, \ldots, x_N)$, stands for the current feature of the sample data.
Step 3: Delete $x_i (i = 1, 2, \ldots, N)$ from X and update $X^{(1,i)} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{41})$ as the new feature vector.
Step 4: A classifier is undertaken to evaluate the effective of the features combination, $X^{(1)}, X^{(2)}, \ldots, X^{(N)}$. Record the ith x_i with the best performance of $X^{(j)}, \bar{x}_j = x_i$.
Step 5: Delete x_i from $X = (x_1, x_2, \ldots, x_N), N - -, j + +$. Go to step 2.
Step 6: Repeat step 2 to step 5, until a series of $\bar{x}_j$ is obtained.
Step 7: Evaluate the accuracy and efficiency performance of classifier with feature subset of $\{\bar{x}_j\}$. Choose the most balanced one.

The outcome of the comparison of above four algorithms is listed in the next section. Statistical result shows the advantage of the proposed GFR method.

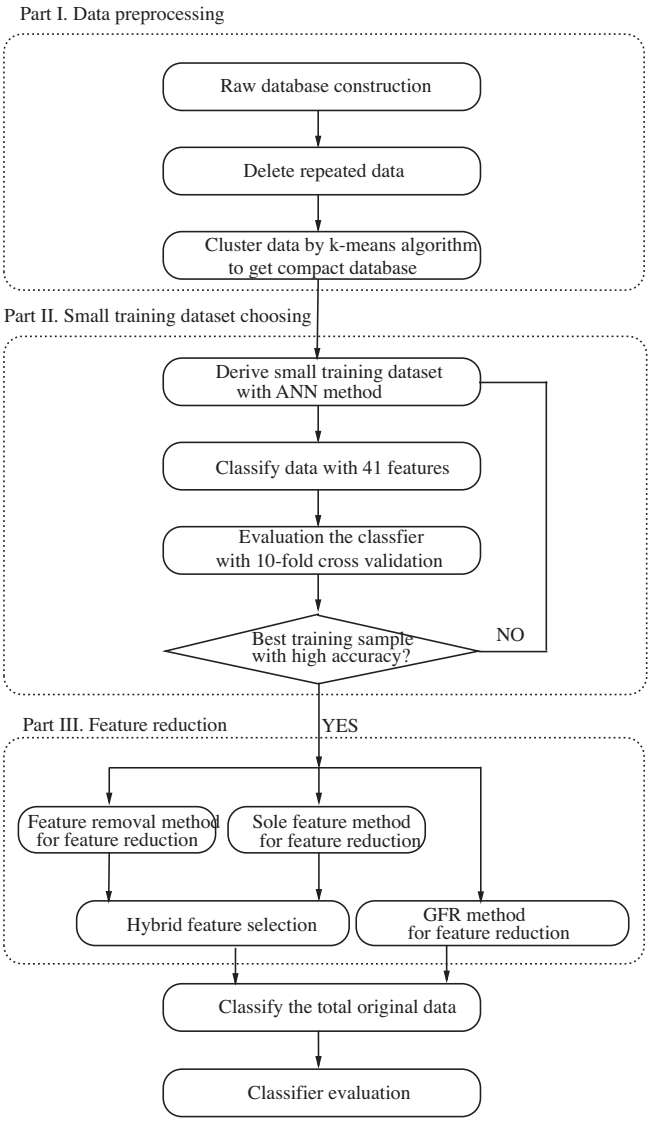

Fig. 4. Flowchart of the algorithm of IDS with reduced features.

2.6. Pipeline of the IDS with feature reduction

The following stepwise procedure is employed so as to implement the pipeline IDS.

Step a: Data preprocessing. In this part, repeated data are deleted from the KDD99 database. By using K-means clustering method, compact data set is constructed.
Step b: Small training dataset construction. In this part, small training dataset is chosen by an artificial intelligence method, ACO algorithm, which ensure the robustness of the chosen small training subset.
Step c: Feature reduction. Dataset is trained and tested with 41 features and 4 different feature reduction strategies. Finally, 19 critical features are chosen by GFR method.

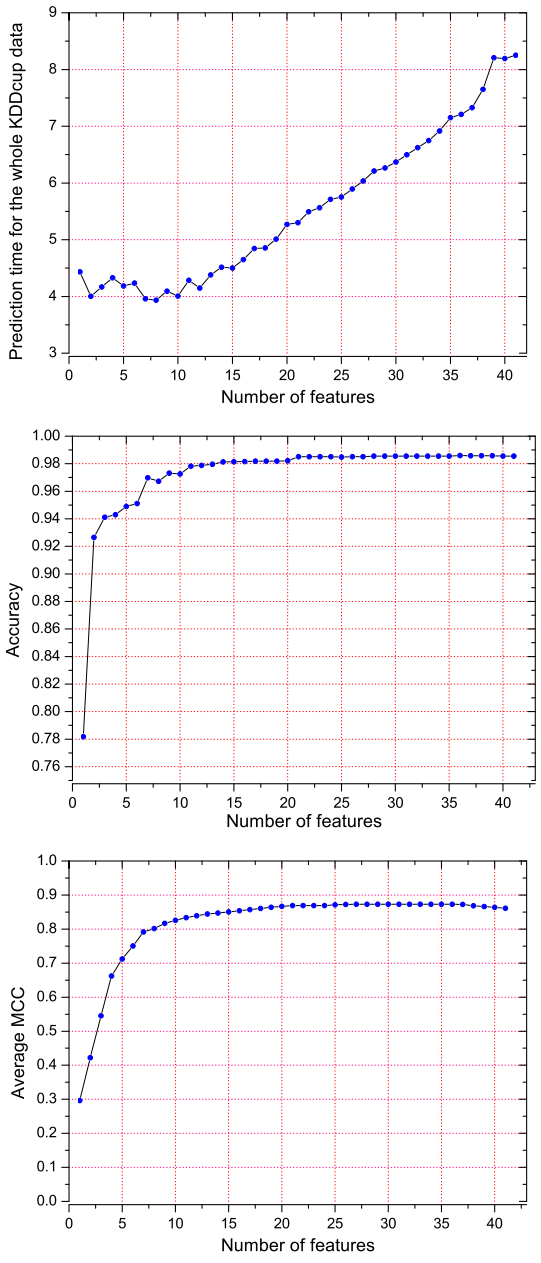

Fig. 5. Accuracy, avg MCC and prediction time with chosen GFR features.

Step d: Grid search of the parameters and in SVM. Evaluate the possibility of occasionality for an unknown network visit to be normal visit or not. Thus an IDS system is constructed.

The total procedure of the new IDS pipeline is listed in Fig. 4.

3. Results

3.1. Evaluation criteria for prediction

During a 5-class problem in IDS, the dataset is separated into five classes and each sample faces five different possibilities. The following measurements are often used to evaluate the efficiency of the classifier:

- True positive (TP_i): The number of sample that is correctly classified into the ith class;
- False positive (FP_i): The number of samples being wrongly classified into the ith class;

Table 3
Chosen feature from four feature reduction method.

Algorithm	Feature number	Feature list
2.5.1	10	8, 10, 14, 31, 32, 33, 35, 36, 37, 40.
2.5.2	10	6, 7, 23, 24, 25, 29, 30, 31, 32, 38.
2.5.3	10	10, 14, 23, 24, 25, 31, 32, 33, 36, 38.
GFR method	19	2, 4, 8, 10, 14, 15, 19, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40.

Table 1
The list of the gradually removed features in GFR method.

Round	Remained feature
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 (28)
2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 (22)
3	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 (12)
4	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 (18)
5	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 (26)
6	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41 (39)
7	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (5)
8	1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (6)
9	1, 2, 3, 4, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (9)
10	1, 2, 3, 4, 7, 8, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (17)
11	1, 2, 3, 4, 7, 8, 10, 11, 13, 14, 15, 16, 19, 20, 21, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (20)
12	1, 2, 3, 4, 7, 8, 10, 11, 13, 14, 15, 16, 19, 21, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (21)
13	1, 2, 3, 4, 7, 8, 10, 11, 13, 14, 15, 16, 19, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (11)
14	1, 2, 3, 4, 7, 8, 10, 13, 14, 15, 16, 19, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (24)
15	1, 2, 3, 4, 7, 8, 10, 13, 14, 15, 16, 19, 23, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (7)
16	1, 2, 3, 4, 8, 10, 13, 14, 15, 16, 19, 23, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (30)
17	1, 2, 3, 4, 8, 10, 13, 14, 15, 16, 19, 23, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41 (41)
18	1, 2, 3, 4, 8, 10, 13, 14, 15, 16, 19, 23, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (13)
19	1, 2, 3, 4, 8, 10, 14, 15, 16, 19, 23, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (16)
20	1, 2, 3, 4, 8, 10, 14, 15, 19, 23, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (1)
21	2, 3, 4, 8, 10, 14, 15, 19, 23, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (23)
22	2, 3, 4, 8, 10, 14, 15, 19, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (3)
23	2, 4, 8, 10, 14, 15, 19, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (19)
24	2, 4, 8, 10, 14, 15, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (15)
25	2, 4, 8, 10, 14, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (27)
26	2, 4, 8, 10, 14, 25, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (25)
27	2, 4, 8, 10, 14, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40 (34)
28	2, 4, 8, 10, 14, 29, 31, 32, 33, 35, 36, 37, 38, 40 (38)
29	2, 4, 8, 10, 14, 29, 31, 32, 33, 35, 36, 37, 40 (31)
30	2, 4, 8, 10, 14, 29, 32, 33, 35, 36, 37, 40 (37)
31	2, 4, 8, 10, 14, 29, 32, 33, 35, 36, 40 (8)
32	2, 4, 10, 14, 29, 32, 33, 35, 36, 40 (29)
33	2, 4, 10, 14, 32, 33, 35, 36, 40 (40)
34	2, 4, 10, 14, 32, 33, 35, 36 (32)
35	2, 4, 10, 14, 33, 35, 36 (4)
36	2, 10, 14, 33, 35, 36 (10)
37	2, 14, 33, 35, 36 (36)
38	2, 14, 33, 35 (14)
39	2, 33, 35 (2)
40	33, 35 (33)
41	35 (35)

Table 2
Confusion matrix obtained with 19 features in training and testing.

	Normal	DOS	R2L	U2R	Probing
Normal	0.99507	0.00102696	0.00232489	0.000102468	0.00147554
DOS	0.0203529	0.976785	0.00118376	7.8795e-05	0.00159972
R2L	0.0953954	0	0.903403	0.0012012	0
U2R	0.211538	0	0.25	0.538462	0
Probing	0.0690286	0.012717	0.00333177	0.000328484	0.914594

Table 4
Performance comparison of IDS system with feature reduction.

	$log_2\gamma$	log_2C	Training time (s)	Testing time (s)	Accuracy (%)	MCC_{avg}
2.5.1	8	0.5	0.073433	3.74173	97.0890	0.799040
2.5.2	16	0.5	0.117039	5.00390	95.0078	0.495120
2.5.3	4	0.5	0.094401	4.21113	96.4591	0.734857
GFR method	4	0.5	0.118356	4.63227	98.6249	0.861161
Total features	2	0.5	0.159150	7.80018	98.6750	0.868684

- True negative (TN_i): The number of outer samples that is correctly classified;
- False negative (FN_i): The number of ith class samples which is wrongly classified into the other classes;
- $ACCuracy = \frac{\sum(TP_i+TN_i)}{\sum(TP_i+TN_i+FP_i+FN_i)}$;
- MCC is the Matthews correlation coefficient, which performs well even in the unbalanced classes. $MCC_i = \frac{TP_i\times TN_i - FP_i\times FN_i}{\sqrt{(TP_i+FN_i)(TP_i+FP_i)(TN_i+FP_i)(TN_i+FN_i)}}$;
- $MCC_{avg} = \frac{\sum MCC_i}{5}$.

On the other hand, classifier is evaluated with 10-fold cross validation, which is a technique for estimating the performance of a classifier. First, the original samples are randomly partitioned into 10 subsets. Secondly, one subset is singled out to be the testing data and the remaining 9 subsets are treated as training data. Afterwards, the cross validation process repeat 10 times and the estimation accuracy of the classifier can be evaluated by the average accuracy of the ten estimations.

The 10-fold cross validation is more popular in the circumstances of huge data set, compared with the Leave-one-out cross-validation. The latter is usually very time expensive according to the high complexity of training times.

3.2. Experimental results

The experiment run in a Pentium-Pro 2.93 GHz computer with 1.80 G memory running Fedora Linux 9. The code for data processing and data mining is written in C++, and the SVM package is Libsvm 2.91, which is developed by Chang and Lin (2001).

By gradually remove the less important features, GFR method decide the importance order of 41 features. The explicit results are obtained in Fig. 5, where the feature in the bracket of each line is the chosen removal features. Therefore, it is inferred that the order of the critical features is 35, 33, 2, 14, 36, 10, 4, 32, 40, 29, 8, 37, 31, 38, 34, 25, 27, 15, 19, 3, 23, 1, 16, 13, 41, 30, 7, 24, 11, 21, 20, 17, 9, 6, 5, 39, 26, 18, 12, 22, 28.

Begin with the 35th feature, feature is gradually added into the SVM classifier to find the proper magnitude of feature set. To better understand the choice of necessary features, the accuracy and avgMCC value with different feature combination is listed in the Fig. 5 (see Table 1).

From the results presented in Fig. 5, feature set of the preceding 19 features performs well with balanced performance and proper feature magnitude. In detail, the feature chosen is 2, 4, 8, 10, 14, 15, 19, 25, 27, 29, 31, 32, 33, 34, 35, 36, 37, 38, 40. By using these 19 features, the accuracy of the SVM classifier achieves 98.6249% in 10-fold cross validation, and the average MCC achieves 0.861161. Moreover, the explicit confusion matrix is shown in Table 2.

In order to evaluate the advantage of GFR method, the other three feature reduction algorithms are also undertaken. In the 2.5.1, 10 important features are chosen. Similarly, 2.5.2 chooses other 10 critical features. Furthermore, by choosing the common preceding ones both appear in the above two algorithm, 10 critical features are derived in 2.5.3. The explicit list of chosen feature is listed in Table 3 as below.

Among the feature list, several common features are notable which exist in most circumstances. Concretely, the common features are 10th, 14th, 31th, 32th, 33th and 36th features that stand for hot, root_shell, srv_diff_host_rate, dst_host_count, dst_host_srv_count and dst_host_same_src_port_rate, respectively. The proportion of the transportation features of target host is high. On the other hand, the common features themselves prove the consistency of these feature reduction methods.

The comparison of the IDS with four feature reduction method and the IDS with original features are listed in Table 4.

From the result of Table 3, the average MCC of GFR method is higher than that of other three algorithms. One notes that the average MCC of 2.5.2 is only 0.495120, which is much higher than other method. This may stem from the inferiority of the filter-based method, since 2.5.2 (sole feature method) is based on a filter strategy. Instead, both of 2.5.1 and GFR method are wrapper-based method, which shows great advantage in the efficiency of classifier. In addition, the essence of GFR method is originated from the thorough precise wrapper strategy. Henceforth, GFR performs better than 2.5.2. Furthermore, the MCC value of GFR method is slightly lower than that of total feature SVM classifier while the training and predicting time is greatly reduced.

4. Conclusion

In this paper, a pipeline of IDS via a series of machine learning strategies is proposed with the following steps: construct a compact data set by clustering redundant data into a compact one; select a proper small training data set with the method of ACO; reduce the feature dimension from 41 to 19 so as to seize the key feature of the network visit; obtain the classifier with SVM and undertake a thorough prediction to the total KDD cup data set.

The accuracy of this IDS pipeline achieves 98.6249%, and MCC value achieves 0.861161. The result show that this IDS pipeline is a reliable one, which performs well in accuracy and efficiency.

One emphasis is to put on the research of feature reduction method. By giving a precise wrapper-based feature reduction method, the GFR method is proposed. This method owns reasonable property in precise feature selection and shows advantage in the experimental result.

Another interesting aspect to further develop is how to choose the proper small training data set. Though this problem is dealt with ACO in this research, there are still some challenges to be explored. Our future work will focus on the following aspects: (1) Current strategy of small training data setting is not adaptive to complex program in multiple classification problem, especially in the unbalanced circumstances. The more complex model and associated improved approach of training data setting for that will be considered; (2) To what extent does a sole feature contribute to the identification of the network visit? The strategy of critical features combination, which is related with the network behaviors and distinguish the goal of attackers and normal users, will be considered.

Acknowledgements

This work was supported by the Funding of Natural Science Research in Hubei Province, 2009CDZ017, and the Doctoral funding of Huazhong Agricultural University 52204-06031.

References

Amor, N. B., Benferhat, S., & Elouedi, Z. (2004). *Naïve Bayes vs decision trees in intrusion detection systems*. *SAC' 04: Proceedings of the 2004 ACM Symposium on Applied Computing*. New York, NY, USA: ACM Press.

Ball, G., & Hall, D. (1967). A clustering technique for summarizing multivariate data. *Behavioral Science, 12*, 153–155.

Chang, C. C., & Lin, C. J. (2001). Training nu-support vector classifiers: Theory and algorithms. *Neural Computation, 13*(9), 2119–2147.

Chebrolu, S., Abraham, A., & Thomas, J. P. (2005). Feature deduction and ensemble design of intrusion detection systems. *Computers and Security, 24*(4), 295–307.

Chimphlee, W., Addullah, A. H., Sap, M. N. M., Srinoy, S., & Chimphlee, S. (2006). Anomaly-based intrusion detection using fuzzy rough clustering. In *Paper presented at the international conference on hybrid information technology (ICHIT' 06)*.

Depren, O., Topallar, M., Anarim, E., & Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications, 29*, 713–722.

Joseph, J. F. C., Das, A., Lee, B. S., & Seet, B. C. (2010). CARRADS: Cross layer based adaptive real-time routing attack detection system for MANETS. *Computer Networks, 54*(7), 1126–1141.

Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB Journal, 16*, 507–521.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*, 273–324.

Li, Y., & Guo, L. (2007). An active learning based TCM-KNN algorithm for supervised network intrusion detection. *Computer and Security, 26*, 459–467.

Li, Y., Wang, J. L., Tian, Z. H., Lu, T. B., & Young, C. (2009). Building lightweight intrusion detection system using wrapper-based feature selection mechanisms. *Computers and Security, 28*(6), 466–475.

Mukkamala, S., Sung, A. H., & Abraham, A. (2004). Modeling intrusion detection systems using linear genetic programming approach. *Proceedings of innovations in applied artificial intelligence, 17th international conference on industrial and engineering applications of artificial intelligence and expert systems (IEA/AIE). Lecture notes in computer science* (Vol. 3029). Springer.

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

Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications, 36*, 11994–12000.

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

Tsang, C. H., Kwong, S., & Wang, H. (2007). Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recognition, 40*, 2373–2391.

Vapnik, V. (1998). *Statistical learning theory*. New York: John Wiley.

Wang, W., & Battiti, R. (2006). Identifying intrusions in computer networks with principal component analysis. In *Paper presented at the proceedings of the first international conference on availability, reliability and security (ARES' 06)*.

Wang, G., Hao, J., Ma, J., & Huang, L. (2010). A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert Systems with Applications, 37*, 6225–6232.

Wu, S. X., & Banzhaf, W. (2010). Use of computational intelligence in intrusion detection systems: A review. *Applied Soft Computing, 10*(1), 1–35.

Xiang, C., Yong, P. C., & Meng, L. S. (2008). Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. *Pattern Recognition Letters, 29*(7), 918–924.