Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection System

You Chen[1,2], Yang Li[1,2], Xue-Qi Cheng[1], and Li Guo[1]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080
[2] Graduate School of the Chinese Academy of Sciences, Beijing 100039
{chenyou, liyang, chengxueqi, guoli}@software.ict.ac.cn

Abstract. The Intrusion detection system deals with huge amount of data which contains irrelevant and redundant features causing slow training and testing process, higher resource consumption as well as poor detection rate. Feature selection, therefore, is an important issue in intrusion detection. In this paper we introduce concepts and algorithms of feature selection, survey existing feature selection algorithms in intrusion detection systems, group and compare different algorithms in three broad categories: filter, wrapper, and hybrid. We conclude the survey by identifying trends and challenges of feature selection research and development in intrusion detection system.

Keywords: intrusion detection, feature selection, filter, wrapper, hybrid.

1 Motivation and Introduction

Intrusion Detection System (IDS) plays vital role of detecting various kinds of attacks. The main purpose of IDS is to find out intrusions among normal audit data and this can be considered as classification problem. One of the main problems with IDSs is the overhead, which can become prohibitively high. As network speed becomes faster, there is an emerging need for security analysis techniques that will be able to keep up with the increased network throughput [1]. Therefore, IDS itself should be lightweight while guaranteeing high detection rates. Several literatures have tried to solve that by figuring out important intrusion features through feature selection algorithms. Feature selection is one of the important and frequently used techniques in data preprocessing for IDS [2], [3]. It reduces the number of features, removes irrelevant, redundant, or noisy data, and brings the immediate effects for IDS.

In terms of feature selection, several researches have proposed identifying important intrusion features through wrapper filter and hybrid approaches. Wrapper method exploits a machine learning algorithm to evaluate the goodness of features or feature set. Filter method does not use any machine learning algorithm to filter out the irrelevant and redundant features rather it utilizes the underlying characteristics of the training data to evaluate the relevance of the features or feature set by some independent measures such as distance measure, correlation measures, consistency measures [4], [5]. Hybrid method combines wrapper and filter approach. Even though

H. Lipmaa, M. Yung, and D. Lin (Eds.): Inscrypt 2006, LNCS 4318, pp. 153 – 167, 2006.

a number of feature selection techniques have been utilized in the fields of web and text mining, and speech recognition, however, there are very few analogous studies in intrusion detection field.

2 General Procedure of Feature Selection

In this section, we explain in detail the four key steps as shown in Fig. 1[36].

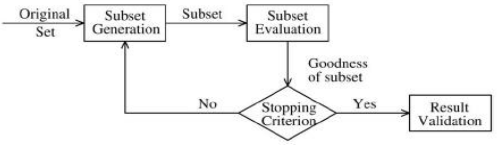

Fig. 1. Four key steps of feature selection

2.1 Subset Generation

Subset generation is essentially a process of heuristic search, with each state in the search space specifying a candidate subset for evaluation. The nature of this process is determined by two basic issues. First, one must decide the search starting point (or points) which in turn influences the search direction. Search may start with an empty set and successively add features (i.e., forward), or start with a full set and successively remove features (i.e., backward), or start with both ends and add and remove features simultaneously (i.e., bidirectional). Search may also start with a randomly selected subset in order to avoid being trapped into local optima [6]. Second, one must decide a search strategy. For a data set with N features, there exist 2N candidate subsets. This search space is exponentially prohibitive for exhaustive search with even a moderate N. Therefore, different strategies have been explored: complete [7], sequential [8], and random [6] search.

2.2 Subset Evaluation

Each newly generated subset needs to be evaluated by an evaluation criterion. An evaluation criterion can be broadly categorized into two groups based on their dependency on learning algorithms that will finally be applied on the selected feature subset. The one is independent criteria, the other is dependent criteria.

Some popular independent criteria are distance measures, information measures, dependency measures, and consistency measures [8], [9], [10], [11]. An independent criterion is used in algorithms of the filter model. A dependent criterion used in the wrapper model requires a predetermined learning algorithm in feature selection and uses the performance of the learning algorithm applied on the selected subset to determine which features are selected.

2.3 Stopping Criteria

A stopping criterion determines when the feature selection process should stop. Some frequently used stopping criteria are as follows:

- The search completes.
- Some given bound is reached, where a bound can be a specified number (minimum number of features or maximum number of iterations).
- Subsequent addition (or deletion) of any feature does not produce a better subset.
- A sufficiently good subset is selected.

2.4 Result Validation

A straightforward way for result validation is to directly measure the result using prior knowledge about the data. In real-world applications, however, we usually do not have such prior knowledge. Hence, we have to rely on some indirect methods by monitoring the change of mining performance with the change of features. For example, if we use classification error rate as a performance indicator for a learning task, for a selected feature subset, we can simply conduct the "before-and-after" experiment to compare the error rate of the classifier learned on the full set of features and that learned on the selected subset [8], [12].

3 Taxonomy of Feature Selection Algorithms

In general, wrapper and filter method have been proposed for feature selection. Wrapper method adopts classification algorithms and performs cross validation to identify important features. Filter method utilizes correlation based approach. Wrapper method demands heavy computational resource for training and cross validation while filter method lacks the capability of minimization of generalization error. In order to improve these problems, several studies have proposed hybrid approaches which combine wrapper and filter approach. In this section, we explain in detail the three key models with some famous feature selection algorithms. In order to compare the differences among these algorithms, we performed all experiments on KDD1999 [24] dataset through open source project WEKA [16]. We experimented in a Windows machine having configurations AMD Opteron 64-bit processor 1.60GHz, 2.00GB RAM, and the operation system platform is Microsoft Windows XP Professional (SP2). We have sampled 10 different datasets, each having 12350 instances, from the corpus by uniform random distribution so that the distribution of the dataset should remain unchanged. Each instance of dataset consists of 41 features. We have carried out 10 experiments on different datasets having full features and selected features and have applied 10 fold cross validation to achieve low generation error and to determine the intrusion detection rate.

3.1 Filter Algorithm

Algorithms within the filter model are illustrated through a generalized filter algorithm [35] (shown in Table 1). For a given data set D, the algorithm starts the search from a given subset S_0. Each generated subset S is evaluated by an independent measure M and compared with the previous best one. The search iterates until a predefined stopping criterion δ is reached. The algorithm outputs the last current best subset S_{best} as the final result. Since the filter model applies independent evaluation criteria without involving any learning algorithm, it does not inherit any bias of a learning algorithm and it is also computationally efficient.

Table 1. A Generalized Filter Algorithm

```
Filter Algorithm
input:   D(F_0, F_1, ..., F_{n-1})   // a training data set with N features
         S_0                         // a subset from which to start the search
         δ                           // a stopping criterion
output:  S_best                      // an optimal subset
01 begin
02    initialize: S_best = S_0;
03    γ_best = eval(S_0, D, M); // evaluate S_0 by an independent measure M
04    do begin
05       S = generate(D); // generate a subset for evaluation
06       γ = eval(S, D, M); // evaluate the current subset S by M
07       if (γ is better than γ_best)
08          γ_best = γ;
09          S_best = S;
10    end until (δ is reached);
11    return S_best;
12 end;
```

Correlation-Based Feature Selection

Correlation-based Feature Selection (CFS) is a filter method. Among given features, it finds out an optimal subset which is best relevant to a class having no redundant feature. It evaluates merit of the feature subset on the basis of hypothesis--"Good feature subsets contain features highly correlated with the class, yet uncorrelated to each other [13]". This hypothesis gives rise to two definitions. One is feature class correlation and another is feature-feature correlation. Feature-class correlation indicates how much a feature is correlated to a specific class while feature-feature correlation is the correlation between two features. Equation 1, also known as Pearson's correlation, gives the merit of a feature subset consisting of k number of features.

$$Merit_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(r-1)\bar{r}_{ff}}} \tag{1}$$

Here, $\bar{r}_{cf}$ is average feature-class correlation, and $\bar{r}_{ff}$ is average feature-feature correlation. For discrete class problem, CFS first dicretizes numeric features using technique Fayyad and Irani [14] and then use symmetrical uncertainty (a modified information gain measure) to estimate the degree of association between discrete features [15].

$$SU = 2.0 \times [\frac{H(X) + H(Y) - H(X,Y)}{H(Y) + H(X)}] \quad (2)$$

In equation 2, H(X) and H(Y) represent entropy of feature X and Y. Symmetrical uncertainty is used because it is a symmetric measure and can therefore be used to measure feature-feature correlation where there is no notion of one attribute being "class" as such[13]. For continuous class data, the correlation between attribute is standard linear correlation. This is straightforward when the two attributes involved are both continuous.

$$\text{Linear Correlation, } r_{xy} = [\frac{\sum xy}{\eta \sigma_x \sigma_y}] \quad (3)$$

In equation 3, X and Y are two continuous feature variables expressed in terms of deviations.

Principal Component Analysis

Principal Component Analysis (PCA) is a probability analyzing method which analyzes the relationships among multivariable, seeks the principal components denoted as a linear combination, and explains the entire changes with several components. The purpose is to make the effective explanations through dimension reduction using linear equations. Although p components are required to reproduce the total system variability, often much of this variability can be accounted for by a small number, k, of the principal components. If so, there is almost as much information in the k components as there is in the original p variables. The k principal components can then replace the initial p variables, and the original data set, consisting of n measurements on p variables, is reduced to one consisting of n measurements on k principal components. [19]. The most common definition of PCA, due to Hotelling (1933) [20], is that, for a set of observed vectors $\{v_i\}$; $i \in \{1,\ldots,N\}$, the q principal axes $\{w_j\}$; $j \in \{1,\ldots,q\}$ are those orthonormal axes onto which the retained variance under projection is maximal. It can be shown that the vectors w_j are given by the q dominent eigenvectors (i.e. those with largest associated eigenvalues) of the covariance matrix $C=\sum_i \frac{(v_i - \bar{v})(v_i - \bar{v})^T}{N}$ such that $Cw_j = \lambda_i w_j$, where $\bar{v}$ is the simple mean. The vector $u_i = w^T (vi - \bar{v})$, where W= $(w_1, w_2, \ldots, w_q)$, is thus a q-dimensional reduced representation of the observed vector v_i.

Experiments and Results

In order to evaluate the effectiveness of CFS and PCA, experiments were performed using ten datasets from the KDD 1999 data [24]. Seven important features were selected by CFS, and then applied to the SVM algorithm. As a feature selection algorithm, PCA extracted eight important features and applied them to the C4.5 [21] algorithm. The performances between these two classifiers are depicted in Fig. 2 and Fig. 3. Fig. 2 shows the true positive rate generated by four classifiers across the folds for each dataset. For all features, it is obvious that the true positive rate of SVM is

much higher than that of C4.5, but for selected features they are nearly equal. In Fig. 3, we can find out that the C4.5 classifier has a lower false positive rate than that of SVM with selected features. Building and testing time of the models are depicted in Table2. Through Table 2, we can see that C4.5 has a fast building speed and testing speed. Compared with the C4.5, the SVM is slower. In Table2 and here after, SVM with all features, SVM with features selected by CFS, C4.5 with all features and C4.5 with features selected by PCA are abbreviated as S, SC, C4.5 and C4.5P respectively.

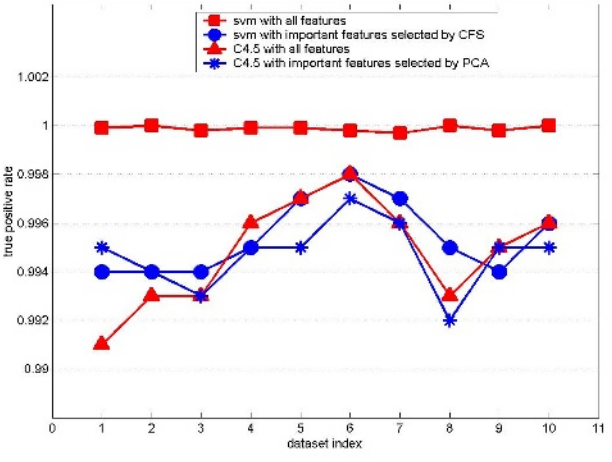

Fig. 2. True positive rate vs. Dataset index

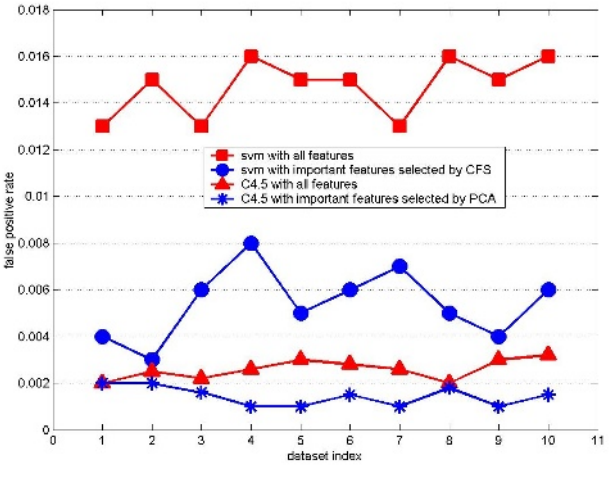

Fig. 3. False positive rate vs. Dataset index

Table 2. Building and Testing time among the four classifiers on the ten datasets

	Classifier	1	2	3	4	5	6	7	8	9	10
Building Time(Sec)	S	119	120	122	125	122	122	123	125	121	124
	SC	52	51	53	52	52	52	52	53	52	51
	C4.5	2.5	3.3	2.5	2.4	3.1	2.3	2.7	2.7	2.4	2.5
	C4.5P	0.9	1.0	1.0	0.9	1.1	0.9	1.2	0.9	0.9	1.0
Testing Time(Sec)	S	53	54	55	54	53	54	53	54	53	53
	SC	24	23	24	24	23	23	24	24	23	23
	C4.5	0.06	0.06	0.05	0.06	0.06	0.05	0.05	0.06	0.06	0.05
	C4.5P	0.03	0.03	0.04	0.04	0.04	0.03	0.03	0.03	0.04	0.04

3.2 Wrapper Algorithm

A generalized wrapper algorithm [35](shown in Table 3) is very similar to the generalized filter algorithm except that it utilizes a predefined mining algorithm A instead of an independent measure M for subset evaluation. Since mining algorithms are used to control the selection of feature subsets, the wrapper model tends to give superior performance as feature subsets found are better suited to the predetermined mining algorithm. Consequently, it is also more computationally expensive than the filter model.

Table 3. A Generalized Wrapper Algorithm

Wrapper Algorithm
input: $D(F_0, F_1, ..., F_{n-1})$ // a training data set with N features
S_0 // a subset from which to start the search
δ // a stopping criterion
output: S_{best} // an optimal subset
01 **begin**
02 **initialize:** $S_{best} = S_0$;
03 $\gamma_{best} = eval(S_0, D, A)$; // evaluate S_0 by a mining algorithm A
04 **do begin**
05 $S = generate(D)$; // generate a subset for evaluation
06 $\gamma = eval(S, D, A)$; // evaluate the current subset S by A
07 if (γ is better than γ_{best})
08 $\gamma_{best} = \gamma$;
09 $S_{best} = S$;
10 **end until** (δ is reached);
11 **return** S_{best};
12 **end**;

Support Vector Machine

Support vector machines, or SVMs, are learning machines that plot the training vectors in high dimensional feature space, labeling each vector by its class. SVMs classify data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in the feature space [22]. SVMs provide a generic mechanism to fit the surface of the hyper plane to the data through the use of a kernel function. The user may provide a function (e.g., linear, polynomial, or sigmoid) to the SVMs during the training process, which selects support vectors along the surface of this function. The number of free parameters used in the SVMs depends on the margin that separates the data points but not on the number of input features, thus SVMs do not require a reduction in the number of features in order to avoid over fitting--an apparent advantage in applications such as intrusion detection. Another

primary advantage of SVMs is the low expected probability of generalization errors. There are other reasons that SVMs are used for intrusion detection. The first is speed: as real-time performance is of primary importance to intrusion detection systems, any classifier that can potentially run "fast" is worth considering. The second reason is scalability: SVMs are relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of the feature space [23], so they can potentially learn a larger set of patterns and thus be able to scale better than neural networks. Once the data is classified into two classes, a suitable optimizing algorithm can be used if necessary for further feature identification, depending on the application [12].

Fusions of GA and SVM

The overall structure and main components of proposed method are depicted in Fig. 4[32]. GA builds new chromosomes and searches the optimal detection model based on the fitness values obtained from the result of SVM classification. A chromosome is decoded into a set of features and parameters for a kernel function to be used by SVM classifier. The SVM is used to estimate the performance of a detection model represented by a chromosome. In order to prevent over fitting problems, n-way cross-validation is used and the detection rates acquired as the results of n tests are averaged so as to obtain a fitness value.

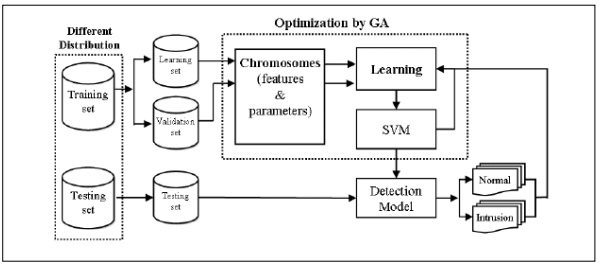

Fig. 4. Overall Structure of Proposed Method

Experiments and Results

Experiments were performed on KDD 1999 dataset [24]. After selecting the important features by using SVM classifier through 5-fold cross validation, we then built the SVM classifier based on those important features. In order to compare the performances between the filter algorithm and the wrapper algorithm, we developed some experiments and summarized the results of them in Fig.5, Fig.6 and Table4. In Fig.5 and Fig.6, we showed the differences of true positive rates and false positive rates among the SVM classifiers which are based on all features or important features selected by CFS or SVM. For features selected by SVM, though the detection rate is lower than that of having features selected by CFS, the decrement is very small, in other words, around 0. 3% in average (see Fig. 5). But the significant performance is achieved in the reduction of false positive rate (see Fig. 6). Table4 shows that for features selected by SVM, the building and testing time of the model are smaller than that of features selected by CFS. In Table4, SVM with features selected by SVM is abbreviated as SS.

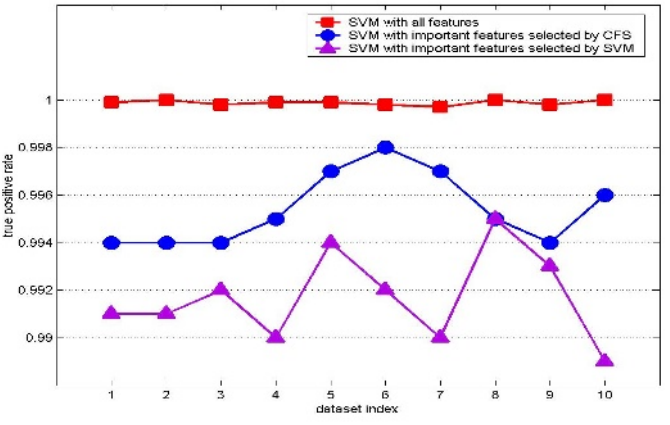

Fig. 5. True positive rate vs. dataset index

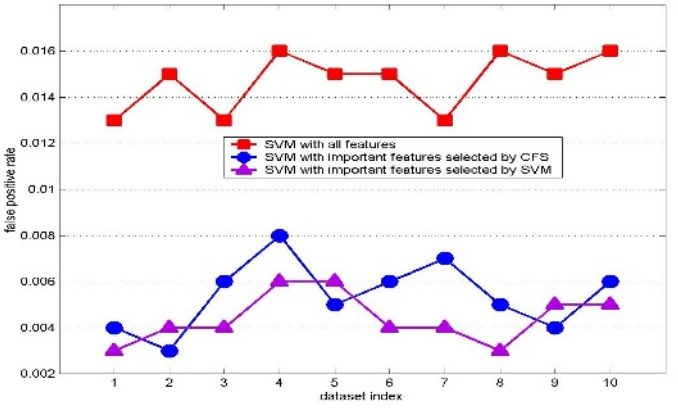

Fig. 6. False positive rate vs. dataset index

Table 4. Building and Testing time among the three classifiers on the ten datasets

	Classifier	1	2	3	4	5	6	7	8	9	10
Building Time(Sec)	**S**	119	120	122	125	122	122	123	125	121	124
	SC	52	51	53	52	52	52	52	53	52	51
	SS	30.1	31.5	31.2	31.3	31.2	30.1	38.1	30.3	30.0	31.8
Testing Time(Sec)	**S**	53	54	55	54	53	54	53	54	53	53
	SC	24	23	24	24	23	23	24	24	23	23
	SS	16	16	16	17	17	17	17	17	17	17

3.3 Hybrid Algorithm

A typical hybrid algorithm [35] (shown in Table 5) makes use of both an independent measure and a learning algorithm to evaluate feature subsets: It uses the independent measure to decide the best subsets for a given cardinality and uses the learning algorithm to select the final best subset among the best subsets across different cardinalities. The quality of results from a learning algorithm provides a natural stopping criterion in the hybrid model.

Table 5. A Generalized Hybrid Algorithm

Hybrid Algorithm
input: $D(F_0, F_1, ..., F_{n-1})$ // a training data set with N features
S_0 // a subset from which to start the search
output: S_{best} // an optimal subset
begin
  initialize: $S_{best} = S_0$;
  $c_0 = card(S_0)$; // calculate the cardinality of S_0
  $\gamma_{best} = eval(S_0, D, M)$; // evaluate S_0 by an independent measure M
  $\theta_{best} = eval(S_0, D, A)$; // evaluate S_0 by a mining algorithm A
  for $c = c_0 + 1$ to N **begin**
    for $i = 0$ to $N - c$ **begin**
      $S = S_{best} \cup \{F_j\}$; // generate a subset with cardinality c for evaluation
      $\gamma = eval(S, D, M)$; // evaluate the current subset S by M
      if (γ is better than γ_{best})
        $\gamma_{best} = \gamma$;
        $S'_{best} = S$;
    end;
    $\theta = eval(S'_{best}, D, A)$; // evaluate S'_{best} by A
    if (θ is better than θ_{best});
      $S_{best} = S'_{best}$;
      $\theta_{best} = \theta$;
    else;
      break and **return** S_{best};
  end;
  return S_{best};
end;

Correlation-Based Hybrid Feature Selection
Correlation-based Hybrid Feature Selection (CBHFS) is a crafted combination of CFS and Support Vector Machines (SVM). It adopt SVM which have been shown a good performance pattern recognition as well as intrusion detection problems [25], [26], [27]. CBHFS is depicted in Fig.7 [36]. As stated earlier, GA is used to generate subsets of features from given feature set. CBHFS takes full feature set as input and returns the optimal subset of feature after being evaluated by CFS and SVM. Each chromosome represents a feature vector. The length of the chromosome is 41 genes where each gene (bit) may have values 1 or 0 which indicates whether corresponding feature is included or not in the feature vector respectively. Like every stochastic algorithm, the initial population of chromosomes is generated randomly. Merit of each chromosome is calculated by CFS. The chromosome having highest Merit, γ_{best} represents the best feature subset, S_{best} in population. This subset is then evaluated by SVM classification algorithm and the value is stored in θ_{best} which represents metric of evaluation. Here we have chosen intrusion detection rates as a metric although a complex criterion such as a combination of detection rate and false positive rate or a rule based criterion like [28] could be used.

Then genetic operations, selection, crossover and mutation, are performed and a new population of chromosomes is generated. In each generation, best chromosome or feature subset is compared by previous best subset, S_{best}. If newer subset is better than previous one, it is assigned as the best subset. This subset is then evaluated by SVM. If new detection rate is higher than previous one, this value is to θ_{best} and algorithm goes forward. Otherwise the S_{best} is returned as the optimal subset of features. The algorithm stops if better subset is not found in next generation or when maximum number of generation is reached.

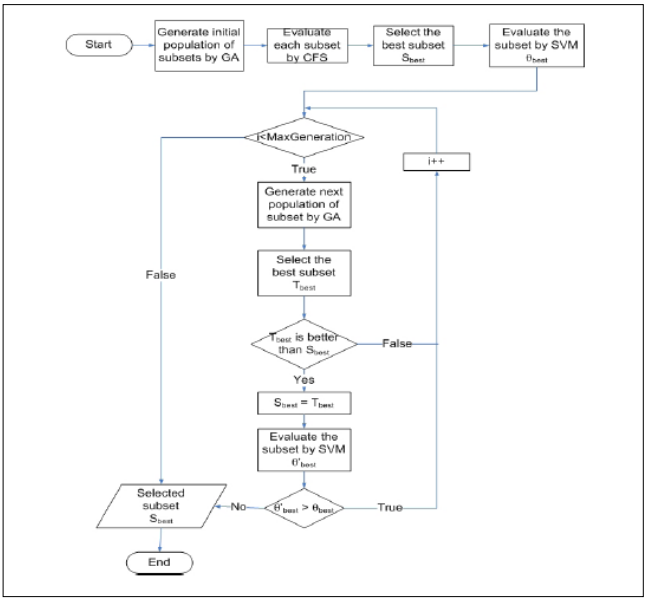

Fig. 7. Flow chart of Correlation-Based Hybrid Feature Selection Algorithm

Random Forest

The overall flow of Random Forest (RF) is depicted in Fig. 8[29]. The network audit data is consisting of training set and testing set. Training set is separated into learning set, validation set. Testing set has additional attacks which are not included in training set. In general, even if RF is robust against over-fitting problem [30], n-fold cross validation method was used to minimize generalization errors [31]. Learning set is used to train classifiers based on RF and figure out importance of each feature of network audit data. These classifiers can be considered as detection models in IDS. Validation set is used to compute classification rates by means of estimating OOB errors in RF, which are detection rates in IDS. Feature importance ranking is performed according to the result of feature importance values in previous step. The irrelevant features are eliminated and only important features are survived. In next phase, only the important features are used to build detection models and evaluated by testing set in terms of detection rates. If the detection rates satisfy design requirement, the overall procedure is over; otherwise, it iterates the procedures.

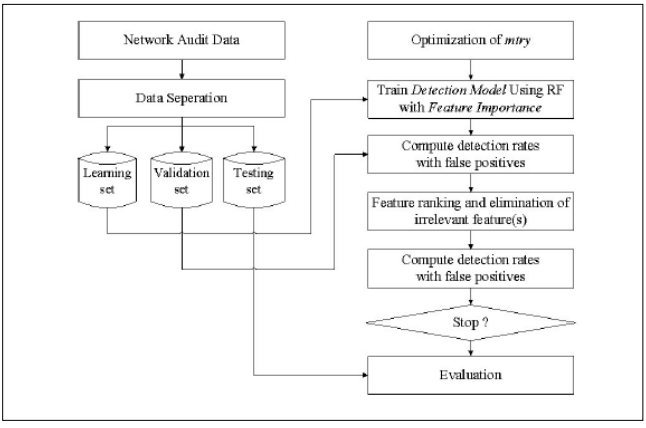

Fig. 8. Overall flow of proposed approach

Experiments and Results

Experiment results are depicted in Fig.9, Fig.10 and Table6. RF has two parameters; the number of variables in the random subset at each node (mtry) and the number of trees in the forest (ntree). As the result of experiments, two optimized parameter values were set; mtry = 6, ntree = 130. For features selected by confusion of CFS and SVM, the true positive rate is nearly equal to that of the features selected by CFS (see Fig.9), but it has a lower false positive rate (see Fig.10). RF has a higher true positive rate and lower false positive rate than SVM, but it requires much more building time (see Table6).

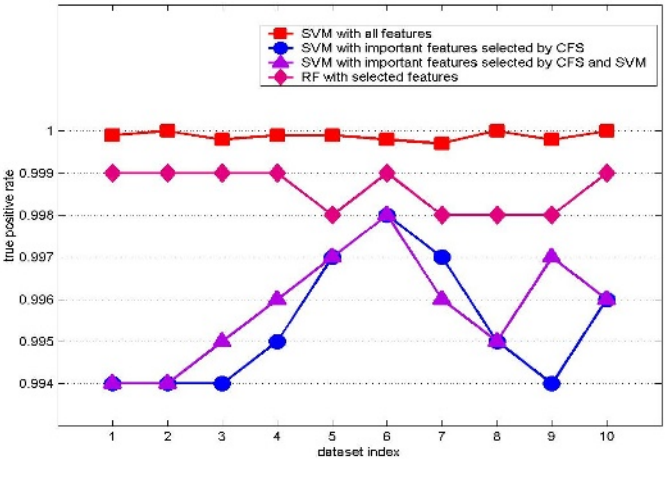

Fig. 9. True positive rate vs. dataset index

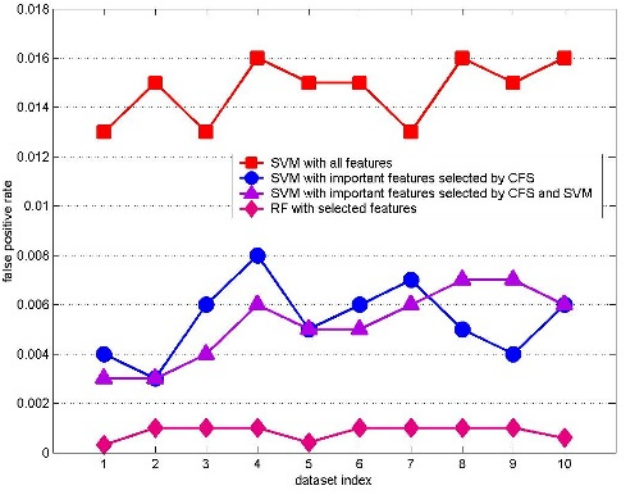

Fig. 10. False positive rate vs. dataset index

Table 6. Building and Testing time among the four classifiers on the ten datasets

	Classifier	1	2	3	4	5	6	7	8	9	10
Building Time(Sec)	S	119	120	122	125	122	122	123	125	121	124
	SC	52	51	53	52	52	52	52	53	52	51
	SCS	57	53	54	53	62	52	64	61	51	56
	RF	170	182	157	148	147	154	153	151	144	154
Testing Time(Sec)	S	53	54	55	54	53	54	53	54	53	53
	SC	24	23	24	24	23	23	24	24	23	23
	SCS	25	24	25	25	25	25	26	25	26	26
	RF	2	2	2	2	2	2	2	3	3	2

4 Concluding Remarks and Future Discussions

This survey provides a comprehensive overview of various algorithms of feature selection. The feature selection of audit data has adopted three main methods; wrapper, filter, and hybrid method. The hybrid approaches have been proposed to improve both filter and wrapper method. However, in some recent applications of feature selection, the dimensionality can be tens or hundreds of thousands. Such high dimensionality causes two major problems for feature selection. One is the so called "curse of dimensionality" [33]. As most existing feature selection algorithms have quadratic or higher time complexity about N, it is difficult to scale up with high dimensionality. Since algorithms in the filter model use evaluation criteria that are less computationally expensive than those of the wrapper model, the filter model is often preferred to the wrapper model in dealing with large dimensionality. Recently,

algorithms of the hybrid model are considered to handle data sets with high dimensionality. These algorithms focus on combining filter and wrapper algorithms to achieve best possible performance with a particular learning algorithm with similar time complexity of filter algorithms. Therefore, more efficient search strategies and evaluation criteria are needed for feature selection with large dimensionality. An efficient correlation-based filter algorithm is introduced in [34] to effectively handle large-dimensional data with class information. Another difficulty faced by feature selection with data of large dimensionality is the relative shortage of instances. Feature selection is a dynamic field closely connected to data mining and other data processing techniques. This paper attempts to survey this fast developing field, show some effective algorithms in intrusion detection systems, and point out interesting trends and challenges. It is hoped that further and speedy development of feature selection can work with other related techniques to help building lightweight IDS with high detection rates and low false positive rates.

References

1. Kruegel, C., Valeur, F.: Stateful Intrusion Detection for High-Speed Networks. In Proc. Of the IEEE Symposium on Research on Security and Privacy (2002) 285–293
2. A.L. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning," Artificial Intelligence, vol. 97, pp. 245-271, 1997.
3. Feature Extraction, Construction and Selection: A Data Mining Perspective, H. Liu and H. Motoda, eds. Boston: Kluwer Academic, 1998, second printing, 2001.
4. Dash M., Liu H., & Motoda H, "Consistency based feature selection", Proc. of the Fourth PAKDD 2000, Kyoto, Japan, 2000, pp. 98–109.
5. H. Almuallim and T.G. Dietterich" "Learning Boolean Concepts in the Presence of Many Irrelevant Features", Artificial Intelligence, vol. 69, nos. 1-2, 1994, pp. 279-305.
6. J. Doak, "An Evaluation of Feature Selection Methods and Their Application to Computer Security," technical report, Univ. of California at Davis, Dept. Computer Science, 1992.
7. P.M. Narendra and K. Fukunaga, "A Branch and Bound Algorithm for Feature Subset Selection," IEEE Trans. Computer, vol. 26, no. 9, pp. 917-922, Sept. 1977
8. H. Liu and H. Motoda, Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academic, 1998.
9. H. Almuallim and T.G. Dietterich, "Learning Boolean Concepts in the Presence of Many Irrelevant Features," Artificial Intelligence, vol. 69, nos. 1-2, pp. 279-305, 1994.
10. M. Ben-Bassat, "Pattern Recognition and Reduction of Dimensionality," Handbook of Statistics-II, P.R. Krishnaiah and L.N. Kanal, eds., pp. 773-791, North Holland, 1982.
11. M.A. Hall, "Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning," Proc. 17th Int'l Conf. Machine Learning, pp. 359-366, 2000.
12. I.H. Witten and E. Frank, Data Mining-Pracitcal Machine Learning Tools and Techniques with JAVA Implementations. Morgan Kaufmann, 2000.
13. Hall, M.A.: Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. In: Proc. of the 17th Int. Conf. on Machine Learning. Morgan Kaufmann Publishers Inc. (2000) 359–366
14. Fayyad, U., Irani, K.: Multi-interval discretization of continuos attributes as preprocessing for classification learning. In: Proc. of the 13th Int. Join Conf. on Artificial Intelligence, Morgan Kaufmann Publishers (1993) 1022–1027

15. Press, W.H., Flannery, B. P., Teukolsky, S. A., Vetterling, W.T.: Numerical recipes in C. Cambridge University Press, Cambridge. (1988)
16. http://www.cs.waikato.ac.nz/ml/weka/index.html
17. Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press (reprinted in 1992 by MIT Press, Cambridge, MA).
18. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, (1975)
19. Johnson, R.A., and Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice Hall (2002) 356-395
20. H. Hotelling. Analysis of a complex statistical variables into principal components. Journal of Educational Psychology, 24:417–441, 1933.
21. J. R. Quinlan. C4.5: Programs for machine learning. Morgan Kaufmann Publishers, 1993.
22. Srinivas Mukkamala, A H. Sung (2002) Comparison of Neural Networks and Support Vector Machines in Intrusion Detection Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection, June 11-13, 2002
23. Sung AH (1998) Ranking Importance of Input Parameters Of Neural Networks. Expert Systems with Applications, pp.405-411.
24. KDD Cup 1999 Data.: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
25. Fugate, M., Gattiker, J.R.: Anomaly Detection Enhanced Classification in Computer Intrusion Detection. Lecture Notes in Computer Science, Vol. 2388. Springer-Verlag, Berlin Heidelberg (2002)
26. Nguyen, B.V.: An Application of Support Vector Machines to Anomaly Detection. (2002) available at. http://www.math.ohiou.edu/~vnguyen/papers/IDS_SVM.pdf
27. Kim, D.S., Park, J.S.: Network-based Intrusion Detection with Support Vector Machines, Lecture Notes in Computer Science, Vol. 2662, Springer-Verlag, Berlin Heidelberg (2003) 747–756
28. Sung, A.H., Mukkamala, S.: Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks. In: Proc. of the 2003 Int. Sym. On Applications and the Internet Technology, IEEE Computer Society Press. (2003) 209–216
29. Dong Seong Kim, Sang Min Lee, and Jong Sou Park:Building Lightweight Intrusion Detection System Based on Random Forest.ISNN 2006, LNCS 3973, pp. 224-230, 2006.
30. Breiman, L.: Random forest. Machine Learning 45(1) (2001) 5–32
31. Duda, R. O., Hart, P. E., Stork, D. G.: Pattern Classification. 2nd edn. John Wiley & Sons, Inc. (2001)
32. Kim, D., Nguyen, H.-N., Ohn, S.-Y., Park, J.: Fusions of GA and SVM for Anomaly Detection in Intrusion Detection System. In.: Wang J., Liao, X., Yi, Z. (eds.): Advances in Neural Networks. Lecture Notes in Computer Science, Vol. 3498. Springer-Verlag, Berlin Heidelberg New York (2005) 415–420
33. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning. Springer, 2001. L. Yu and H. Liu, “Feature Selection for High-Dimensional Data:
34. A Fast Correlation-Based Filter Solution,” Proc. 20th Int’l Conf. Machine Learning, pp. 856-863, 2003.
35. H. Liu and L. Yu. Towards integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering, 17(3):1-12, 2005.
36. Jong Sou Park, Khaja Mohammad Shazzad, Dong Seong Kim: Toward Modeling Lightweight Intrusion Detection System Through Correlation-Based Hybrid Feature Selection. CISC 2005: 279-289. 5