available at www.sciencedirect.com

ScienceDirect

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

Computers & Security

Building lightweight intrusion detection system using wrapper-based feature selection mechanisms

Yang Li[a,d,*], Jun-Li Wang[b], Zhi-Hong Tian[d], Tian-Bo Lu[c], Chen Young[c]

[a]China Mobile Research Institute, Beijing 100053, China
[b]Peking University Founder Technology College, Beijing 065001, China
[c]National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China
[d]Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 8 January 2008
Received in revised form
11 November 2008
Accepted 7 January 2009

Keywords:
Network security
Intrusion detection system
Feature selection
Modified RMHC
Modified linear SVMs

ABSTRACT

Intrusion Detection System (IDS) is an important and necessary component in ensuring network security and protecting network resources and network infrastructures. How to build a lightweight IDS is a hot topic in network security. Moreover, feature selection is a classic research topic in data mining and it has attracted much interest from researchers in many fields such as network security, pattern recognition and data mining. In this paper, we effectively introduced feature selection methods to intrusion detection domain. We propose a wrapper-based feature selection algorithm aiming at building lightweight intrusion detection system by using modified random mutation hill climbing (RMHC) as search strategy to specify a candidate subset for evaluation, as well as using modified linear Support Vector Machines (SVMs) iterative procedure as wrapper approach to obtain the optimum feature subset. We verify the effectiveness and the feasibility of our feature selection algorithm by several experiments on KDD Cup 1999 intrusion detection dataset. The experimental results strongly show that our approach is not only able to speed up the process of selecting important features but also to yield high detection rates. Furthermore, our experimental results indicate that intrusion detection system with feature selection algorithm has better performance than that without feature selection algorithm both in detection performance and computational cost.

1. Introduction

Intrusion detection system (IDS) plays a vital role in detecting various kinds of attacks and it is a valuable tool for the defense-in-depth of computer networks. Network-based IDS looks for known or potential malicious activities in network traffic and raise an alarm whenever a suspicious activity is detected.

In general, IDS 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 is one of the key topics in IDS. For example, in many pattern classification tasks we are confronted with the problem that we have a very high dimensional feature space. Some of these features may be irrelevant or redundant. Removing these irrelevant or

* *Corresponding author*. China Mobile Research Institute, Unit 2, 28 Xuanwumenxi Ave., Xuanwu District, Beijing 100053, China. Tel./fax: +86 10 66006688.
E-mail address: samsunglinux@163.com (Y. Li).
0167-4048/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2009.01.001

redundant features is very important because they may deteriorate the performance of classifiers. Furthermore, by choosing the effective and important features, we can improve the classification mode and improve the classification performance. Feature selection involves finding a subset of features to improve prediction accuracy or decrease the size of the structure without significantly decreasing prediction accuracy of the classifier built using only the selected features (Koller and Sahami, 1996).

Methods for feature selection have been essentially divided into two categories: filter methods and wrapper methods (Kohavi and John, 1997). Wrapper methods use the actual classifier, and its resultant probability of error, to select the feature subsets. The feature selection algorithm is wrapped inside the classifier. Filter methods analyze features independent of the classifier and use ''goodness'' metric to decide which features should be kept. Because the classification results can be used as a metric, wrapper methods generally perform better than filter methods. Wrapper methods involve some more computational complexity and require more execution time than the filter methods due to retraining a classifier for each new set of features. In order to get a better performance with less computational complexity, some researchers have proposed hybrid feature selection methods which combine wrapper and filter methods. However, the performance of the hybrid methods is far from perfect. In this paper, we adopted several methods to improve the wrapper method to solve the computational complexity.

Therefore, in this paper, we firstly introduce a random search method named random mutation hill climbing (RMHC) which can be enhanced in terms of its speed by adopting methods from simulated annealing. In addition, for evaluation criterion, we secondly propose a novel evaluation criterion based on modified linear SVMs for feature selection. Moreover, we apply them to building a lightweight intrusion detection system and examine the feasibility and effectiveness of our feature selection algorithm by conducting several experiments on KDD (Knowledge Discovery and Data mining) Cup (1999) intrusion detection dataset.

The rest of this paper is organized as follows. We outline the related work about our methods in Section 2. Section 3 discusses the feature selection based lightweight IDS in detail. We report our experimental results in Section 4 and conclude our work in Section 5.

2. Related work

Lightweight IDS can be implemented through an efficient feature selection method, but it is a very difficult problem. There are two important parts in feature selection, one is the search strategy, and the other is the evaluation criterion.

For search strategy, Jain and Zongker (1997) found that the heuristic methods (forward sequential search) perform best in large datasets, while Kudo and Sklansky (2000) observed that the random methods (genetic algorithms) are most effective in solving large-scale problems (Jain and Zongker, 1997; Kudo and Sklansky, 2000). However, for large-scale feature selection problems, these popular methods cause too much heavy computational cost especially for practical training times.

For evaluation criterion, support vector machines (SVM) has become a popular tool in recent years due to its remarkable characteristics such as the absence of local minima, the sparse representation and good generalization ability. Weston (Grandvalet and Canu, 2003) introduced a method of the feature selection for SVMs based upon finding those features which minimize bounds on the leave-one-out error. Grandvalet (Cao et al., 2003) introduced an algorithm for the automatic relevance determination of input variables. Guyon (Guyon et al., 2002) utilized SVMs methods based on Recursive Feature Elimination (RFE) for gene selection. These applications have illustrated new aspects of the applicability of SVMs in the field of feature selection. As a kind of classifier for IDS, SVMs (Vapnik, 1995) are still outperformed by many standard classifiers in terms of its classification-speed. Recent work on SVM classification speed up mainly focused on the reduction of the decision problem: a method called RSVM (Reduced Support Vector Machines) was proposed by Lee in (Lee and Mangasarian, 2001). RSVM preselects a subset of training samples as SVMs and solves a smaller Quadratic Programming problem. Moreover, Kim and Park (Kim et al., 2005; Park et al., 2005) and Makkamala et al. (Ribeiro, 2005) proposed a method to optimize parameters of kernel function in SVM. All these methods yield good improvements, but they are fairly complex and computationally expensive. In this paper, we will use a classification method based on a decision tree (Arreola et al., 2006; Amor et al., 2004) whose nodes consist of linear SVMs. The classification method was endorsed by the work of Bennett et al. (2000) that experimentally proved that inducing a large margin in decision trees with linear decision functions improved the generalization ability.

3. Feature selection based lightweight IDS

3.1. *Overall lightweight IDS architecture*

The overall flow of our approach is depicted in Fig. 1. The approach starts the search from a subset S_0 which is evaluated by modified linear SVMs. The metric of the evaluation is θ_{best}, which represents the best feature subset S_{best}. After initialing the values of S_{best} and θ_{best}, the approach goes into an iterative procedure. In each iteration, generated feature subset S is compared by previous best subset S_{best}. If S is better than S_{best}, it is assigned as S_{best}. In this process, each subset S generated by modified RMHC is evaluated by modified linear SVMs in an iterative way. If θ is higher than θ_{best}, it is assigned as θ_{best} and the approach goes forward. The approach stops if a predefined stopping criterion δ is reached or when maximum number of iterate is reached. S_{best} is returned as the optimal subset of features. In next phase, only the selected feature subset S_{best} is used to build detection system which will be evaluated on testing dataset in terms of testing time, true positive rate and false positive rate. In this paper, we used a decision tree (Arreola et al., 2006; Amor et al., 2004) whose nodes consist of linear SVMs as classifier to build lightweight IDS.

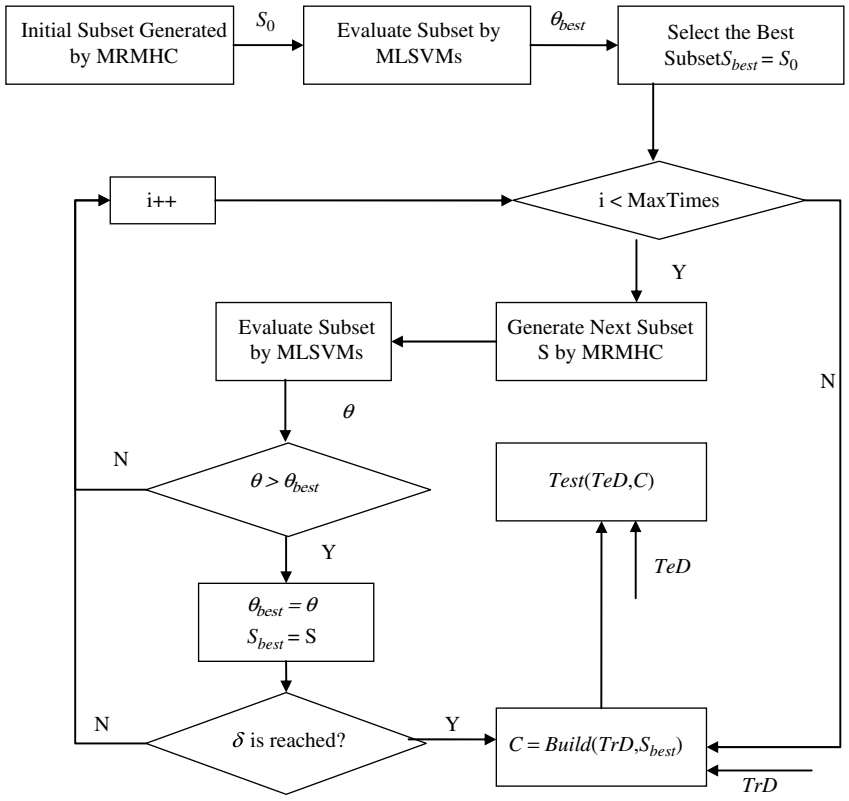

Fig. 1 – Illustration of intrusion detection system based on a wrapper method of feature selection.

3.2. *Modified RMHC (random mutation hill climbing) method*

Random mutation hill climbing is a member of the family of random search optimization tools that include methods such as simulated annealing, random mutation, and genetic algorithms (Skalak, 1994). Random search algorithms derive their power from the ability to search the optimization space in a random manner, which makes them inherently immune to local minima. The difficulty of the random methods is that the randomness must be controlled to ensure the method converges, while allowing it to be free enough to allow 'complete' coverage of the overall search space.

For the random mutation hill climbing algorithm, the complete set of features is represented by a binary string of length *N*, where a bit in the string is set to '1' if it is to be kept, and set to '0' if it is to be discarded, and *N* is the original number of features (Skalak, 1994). The key free parameter to set when using an algorithm such as random mutation is the number of bits, *M*, that are allowed to randomly change at each iteration. The most conservative approach is to only allow a single bit to change per pass (Skalak, 1994). The algorithm operates as follows:

Step (1): Initialize a binary string, *S*, of length *N*, where *M* features are marked as used,'1' and the remaining *N*–*M* are '0'.
Step (2): Test binary string, *S*, for fitness $F(S)$ using the probability of correct classification.
Step (3): Randomly mutate *M* bits in the binary string, *S*.
Step (4): Return to step (2) and continue until either the fitness goal is reached or the maximum number of iterations is reached.

Since this is a wrapper algorithm, the definition of the fitness function for the basic method is simply the classification error:

$$F(S) = P_{error}(S) \tag{1}$$

where *S* is the set of currently utilized features.

Then, we modify the RMHC to enhance its speed and improve its dimensionality reduction ability. We adopt a method that is loosely motivated by simulating annealing where a system is cooled over time (Kirkpatrick et al., 1983). We implement this concept of cooling by reducing the number of features that can be mutated at each iteration. This allows us to get the benefit of rapidly changing the mix of the features in the early iterations, and then more slowly changing the set of features as the system converges to a solution. The number of features to mutate at any iteration is:

$$M = M_{\max} * \min\left[\frac{(I_{\max} - i_{current})}{I_{\max}}, P_{error}(S)\right] \tag{2}$$

where M_{max} is the maximum allowed value for the number of features to mutate, I_{max} is the maximum number of iterations, $i_{current}$ is the current iteration, and $P_{error}(S)$ is the current error rate.

The definition of the fitness function is also required for random mutation hill climbing. Since this is a wrapper algorithm, the fitness function must be a function of the

classification accuracy. It is worth noting that the fitness function should also be a function of the number of features remaining or else there will be no explicit incentive to reduce the number of features. A natural fitness function is then:

$$F(S) = \alpha \cdot P_{error}(S) + (1-\alpha)\cdot\frac{|S|}{N} \tag{3}$$

where N is the original number of features, S is the current set of features, $|S|$ is the cardinality of S, and $0 \le \alpha \le 1$ is the relative weighting factor between dimensionality reduction and error rate. Thus, this fitness function is the weighted average of the classification error and the fraction of the features used. The goal in the random search is to drive both of these values to zero simultaneously, but at some point there is clearly a trade-off between dimensionality reduction and error rate. The parameter α is set based on how aggressively the algorithm reduces the number of features. A larger α encourages the final solution to be based more on the resultant classification error, and a smaller α encourages the final solution to use fewer features at the expense of classification accuracy.

3.3. *Modified linear SVM (support vector machines)*

Linear SVMs introduced by Vapnik (1998) has been shown to have good performance over real applications. Given the training data $\{(x_i, y_i)\}_1^l$ with input data $x_i \in R^n$ and the corresponding binary class label $y_i \in \{-1,1\}$. The goal of SVM is to find an optimal hyperplane that separates two classes such that the hyperplane is the farthest away from the closed training vectors of each one class. Often, the hyperplane can be obtained by solving the following equation:

$$\begin{aligned} &\min \tfrac{1}{2}\langle \omega, \omega\rangle \\ &y_i(\langle \omega, x_i\rangle + b) \ge 1, i = 1, \ldots, l. \end{aligned} \tag{4}$$

where $(\omega, b) \in R^n \times R$ and $\langle .,. \rangle$ is the inner product of two vectors. When data cannot be perfectly separable, a penalty term $C\sum_{i=1}^{l}\xi_i$ is added to the objective function in Equation (4), where C is a positive number. Accordingly, the linear SVMs is to solve the following problem:

$$\begin{aligned} &\min \tfrac{1}{2}\langle \omega, \omega\rangle + C\sum_{i=1}^{l}\xi_i \\ &y_i(\langle \omega, x_i\rangle + b) \ge 1 - \xi_i, i = 1, \ldots, l,\ \xi_i \ge 0, i = 1, \ldots, l \end{aligned} \tag{5}$$

Equation (5) can be solved in the dual space of Lagrange multiplies $\alpha_i \ge 0, i = 1, \ldots, l$. In such a case, Equation (5) can be translated into

$$\begin{aligned} &\max W(\alpha) = \sum_{i=1}^{l}\alpha_i - \tfrac{1}{2}\sum_{i=1}^{l} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle \\ &\sum_{i=1}^{l} y_i \alpha_i = 0, \xi_i \ge 0, i = 1, \ldots, l \end{aligned} \tag{6}$$

After α_i, b are obtained, the following decision function is defined as follows:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l}\alpha_i y_i \langle x_i, x\rangle + b\right) \tag{7}$$

Note that although there are l training samples in Equation (7), only the samples with $\alpha_i > 0$ play a role in the decision function. The samples with $\alpha_i > 0$ is called support vectors.

Motivated from that the kernel matrix can be learned from data via semidefinite programming (SDP) techniques, we apply the generalized performance measure introduced by Lanckriet for feature extraction in this section. In (Lanckriet et al., 2004), Lanckriet et al. developed the following generalized performance measure for choosing kernel parameters.

$$\begin{aligned} &\min_{k \in K}\ \max_{\alpha}\ 2\alpha^T e - \alpha^T diag(Y) k\, diag(Y)\alpha \\ &\alpha^T y = 0, trace(k) = c, C \ge \alpha \ge 0 \end{aligned} \tag{8}$$

where k is a linear combination of different kernel matrices such as Gaussian kernel, polynomial kernel, $diag(Y) = diag(y_1, y_2, \ldots, y_l)$, $y = (y_1, y_2, \ldots, y_l)$, $e = (1 \cdots 1)$. Note that the Gram matrix k in linear support vector machines can be denoted as a linear combination of dimensions of features. In other words, there exists

$$k = \begin{bmatrix} \langle x_1, x_1\rangle & \cdots & \langle x_1, x_l\rangle \\ \vdots & \vdots & \vdots \\ \langle x_l, x_1\rangle & \cdots & \langle x_l, x_l\rangle \end{bmatrix} \tag{9}$$

Furthermore, inspired by the idea of feature selection in Lanckriet et al. (2004), we introduce the weights $\beta_i (i = 1, \ldots, l)$ in the Gram matrix k. In such a case, Equation (9) can be transformed into

$$\begin{aligned} k1 &= \beta_1 (x_{11}, \ldots, x_{l1})(x_{11}, \ldots, x_{l1})^T + \cdots \beta_n (x_{1n}, \ldots, x_{ln})(x_{1n}, \ldots, x_{ln})^T \\ &= \sum_{i=1}^{n} \beta_n v_i v_i^T \end{aligned}$$

where $v_i = (x_{1i}, \ldots, x_{1n})^T$.

It is obvious that the ith feature is removed if $\beta_i = 0$. The bigger β_i, the more important the corresponding ith feature. Note that the condition $\beta_i \ge 0$ is imposed such that the matrix $k1$ is positive semidefinite. In such a case, we rewrite Equation (8) as follows:

$$\begin{aligned} &\min_{\beta_i}\ \max_{\alpha}\ 2\alpha^T e - \alpha^T diag(Y) \textstyle\sum_{i=1}^{n} \beta_i v_i v_i^T diag(Y)\alpha \\ &\alpha^T y = 0, C \ge \alpha \ge 0.\ trace\left(\textstyle\sum_{i=1}^{n} \beta_i v_i v_i^T\right) = c, \beta_i \ge 0 (i = 1, \ldots n). \end{aligned} \tag{10}$$

Applying the similar idea in Lanckriet et al. (2004), we can recast Equation (10) as a semidefinite program. A general-purpose program such as SeDuMi (Wolf, in press) or SDPT3 (Amor et al., 2004) can be used to solve these problems. Furthermore, note that $k1$ is a linear combination of rank-one matrices $\beta_i v_i v_i^T$. Equation (10) can be transformed into the following form:

$$\begin{aligned} &\max_{\alpha}\ 2\alpha^T e - ct \\ &\alpha^T y = 0, C \ge \alpha \ge 0.\ trace\left(\textstyle\sum_{i=1}^{n} \beta_i v_i v_i^T\right) = c, \\ &t \ge (diag(Y) v_i)^2, i = 1, \ldots n.\ \beta_i \ge 0 (i = 1, \ldots n) \end{aligned} \tag{11}$$

It is obvious that Equation (11) belongs to the quadratically constrained quadratic programming (QCQP), which is a special form of SDP. The QCQP can be solved by MOSEK optimization software. After Equation (11) is solved, the dual variables $\beta_i (i = 1, \ldots n)$ can be easily obtained. Note that if strong duality holds, the optimal values for $\beta_i (i = 1, \ldots n)$, $\alpha_j (j = 1, \ldots l)$ are simultaneously obtained. However, when the optimal values $\alpha_j (j = 1, \ldots l)$ are obtained, we can see that Equation (11) is a linear programming problem with respect to $\beta_i (i = 1, \ldots n)$ under the assumption that $\alpha_j (j = 1, \ldots l)$ are fixed. It is shown that the optimal values $\beta_i (i = 1, \ldots n)$ are extreme points in linear programming. Therefore, in such a case, only a feature is adopted, which is often not reasonable for feature extraction. It also

shows from a theoretical viewpoint the reason that only a kernel matrix is adopted when a linear combination of different kernels in 1-norm support vector machines is applied. The detailed experimental results can be found in Lanckriet et al. (2004).

Since only choosing a feature is not reasonable in general cases, we modify Equation (10) in terms of Wolf idea in Wolf (in press) and change the constraint $trace(\sum_{i=1}^{n}\beta_i v_i v_i^T)$ to $\sum_{i=1}^{c}\beta_i\beta_i = 1$. The constraint $\sum_{i=1}^{c}\beta_i\beta_i = 1$ has been successfully applied in Wolf (in press) and has been shown the existence of sparsity in feature selection for unsupervised inference. Accordingly, we have the following equation.

$$\begin{aligned}&\min_{\beta_i}\max_{\alpha} 2\alpha^T e - \alpha^T diag(Y)\textstyle\sum_{i=1}^{n}\beta_i v_i v_i^T diag(Y)\alpha\\ &\alpha^T y = 0, C \geq \alpha \geq 0\\ &\textstyle\sum_{i=1}^{n}\beta_i\beta_i = 1, \beta_i \geq 0 (i = 1,\ldots n).\end{aligned} \tag{12}$$

In such a case, Equation (12) is a minmax problem. Often the interior-point methods can be used to handle this class of problems. However, note that when the dimensions of features are high and the number of samples is large, the interior-point methods for solving this minmax problem are still computationally expensive. To this end, we resolve to the iterative algorithm for handling Equation (12). For fixed $\beta_i (i = 1,\ldots n)$, classical SVM algorithm can be used for solving Equation (12). For fixed $\alpha_j (j = 1,\ldots l)$, Equation (12) belongs to quadratically constrained quadratic programming, which can be solved by SEMOK optimization software. But, in the following, we will provide an efficient approach for obtaining $\beta_i (i = 1,\ldots n)$. Assume that $\alpha_j (j = 1,\ldots l)$ are given. We can obtain the following Lagrange equation from Equation (10).

$$L(\beta_i) = 2\alpha^T e - \alpha^T diag(Y)\sum_{i=1}^{n}\beta_i v_i v_i^T diag(Y)\alpha + \lambda\left(\sum_{i=1}^{n}\beta_i\beta_i - 1\right) \tag{13}$$

Note that we do not add the constrains $\beta_i \geq 0 (i = 1,\ldots n)$ in Equation (13). Setting the derivate of Equation (13) with respect to β_i to zero, we can obtain

$$2\lambda\beta_i = \alpha^T diag(Y) v_i v_i^T diag(Y)\alpha \tag{14}$$

Applying Equation (14) and $\sum_{i=1}^{c}\beta_i\beta_i = 1$, we have

$$\beta_i = \frac{\alpha^T diag(Y) v_i v_i^T diag(Y)\alpha}{norm(B)} (i = 1,\ldots n) \tag{15}$$

Table 1 – Selected feature subsets for ALL attacks, DOS, PROBE, R2L and U2R.

Attack type	Selected features
ALL	Service, src_bytes, count, dst_host_count
DOS	Protocol_type, src_bytes, count,dst_host_same_srv_rate
PROBE	Duration, service, src_bytes, dst_bytes, count, dst_host_diff_srv_rate
R2L	Duration, service, src_bytes
U2R	Duration, service, src_bytes, root_shell, dst_host_count

Table 2 – Time of selecting processes for different feature selection algorithms.

Feature selection algorithm		ALL (h)	DOS (h)	PROBE (h)	R2L (h)	U2R (h)
Search strategy	Evaluation criterion					
RHMC	Modified Linear SVMS	1.3	0.5	4	1.5	1.5
Modified RHMC	Modified Linear SVMS	0.8	0.3	3.2	1.1	1.0

where $B = (\alpha^T diag(Y) v_1 v_1^T diag(Y)\alpha, \ldots, \alpha^T diag(Y) v_n v_n^T diag(Y)\alpha)$ and $norm(B)$ denotes 2-norm of vector B. Note that β_i is required to not be smaller than zero. Fortunately, we can see that the matrices $diag(Y) v_i v_i^T diag(Y) (i = 1,\ldots n)$ are positive semi-definite. Therefore, $\alpha^T diag(Y) v_i v_i^T diag(Y)\alpha \geq 0 (i = 1,\ldots,n)$ hold. As a result, the condition $\beta_i \geq 0$ are satisfied. As a summary of the above discussion, we state the proposed method as follows.

Step (1): Initialize $\beta_i (i = 1,\ldots n)$ to some values;
Step (2): using a standard SVM algorithm to obtain $\alpha_j (j = 1,\ldots l)$;
Step (3): update the parameters $\beta_i (i = 1,\ldots n)$ by Equation (15);
Step (4): Go to step 2 or stop when the optimal values are obtained or the maximal number of iterations is reached.
Theoretically speaking, the above iterative algorithm is local optimal.

4. Experimental results

As our experimental dataset, the KDD Cup (1999) dataset was collected at the MIT Lincoln Lab to evaluate the proposed intrusion detection systems using feature selection algorithm, and the dataset contains a wide variety of intrusions simulated in a military network environment. It contains NORMAL data and 24 different types of attacks that are broadly categorized in four groups such as PROBE, DOS (Denial of Service), U2R (User to Root) and R2L (Remote to Local).

In the experiments, we firstly utilized our feature selection algorithm to select important features for each type of the previous discussed attacks, and then built lightweight intrusion detection systems using these selected features. For each type of attacks, we compared the intrusion detection systems using selected features with those using all 41 features and evaluated their performance in detecting known attacks and new attacks. All experiments were performed in a Windows platform having configurations Intel(R) Pentium(R) processor 1.73 GHz, 512 Mb RAM.

Table 3 – Average time of building and testing processes with all features and selected features for ALL attacks.

—		ALL	DOS	PROBE	R2L	U2R
Building time(s)	All features	78	136	245	317	193
	Selected features	36	41	123	35	85
Testing time(s)	All features	18	22	49	55	50
	Selected features	8	9	29	8	18

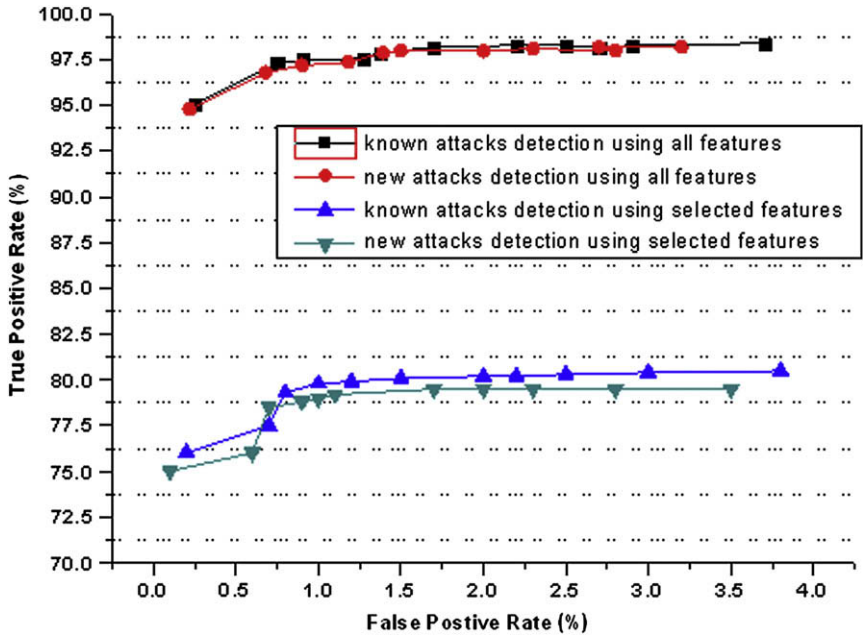

Fig. 2 – ROC curves for all attacks detection.

4.1. *KDD1999 intrusion detection dataset preprocess*

We have preprocessed the KDD Cup (1999) labeled training dataset to form five separate dataset—NORMAL, DOS, PROBE, R2L and U2R. The dataset totally contains 494021 instances, among them, 97278 (19.69%) instances are normal and the other 396743 (80.31%) instances belong to the attacks. It contains 22 different types of attacks that are broadly categorized in four groups (PROBE, DOS, U2R and R2L). To perform our experiments, we constructed five independent training datasets sampling from KDD Cup (1999) intrusion dataset. We combined 97278 normal instances with 391458 DOS instances, 4107 PROBE instances, 1126 R2L instances and 52 U2R instances respectively, and then we sampled four datasets which have 11701 instances, from the above four combined datasets by uniform random distribution so that the distribution of the datasets should remain unchanged. In addition, we also sampled one dataset which have the same 11701 instances, from the total training dataset in KDD1999 by uniform random distribution. Each of the five sampled dataset consists of 41 features.

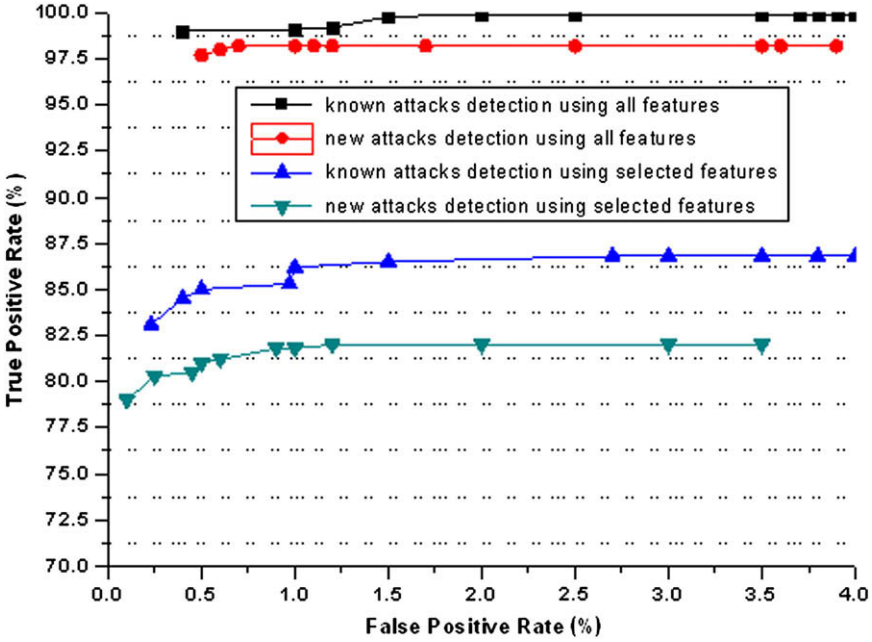

Fig. 3 – ROC curves for DOS attack detection.

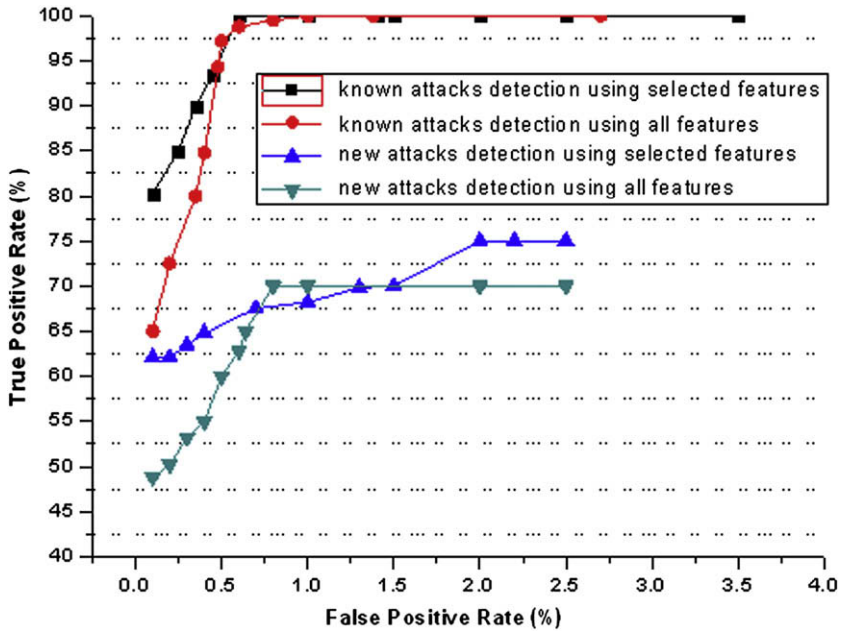

Fig. 4 – ROC curves for PROBE attack detection.

Moreover, in order to evaluate the performance of our approach in detecting known attacks and new attacks, we further preprocessed the KDD Cup (1999) labeled test dataset to form two datasets—known attacks and new attacks for each type of attacks. The test dataset contains total 311029 instances, there are 229853(73.9%) instances belonging to DOS attacks and nearly 2.9% instances of them have new attacks, which do not appear in our training dataset. For each type of attacks, we sampled two different datasets from KDD Cup (1999) test dataset. One is for known attacks, and the other is for new attacks. These sampled test datasets were used for evaluating the proposed lightweight intrusion detection systems.

The overall structure of our approach is depicted in Fig. 1. We designed several experiments based on Fig. 1 to examine the effectiveness of our approach. Our approach consists of two components: our feature selection algorithm which includes modified RMHC and modified linear SVMs, and the classifier algorithm that was used for building intrusion detection systems. In this paper, we used a classification algorithm (Arreola et al., 2006) based on a decision tree whose

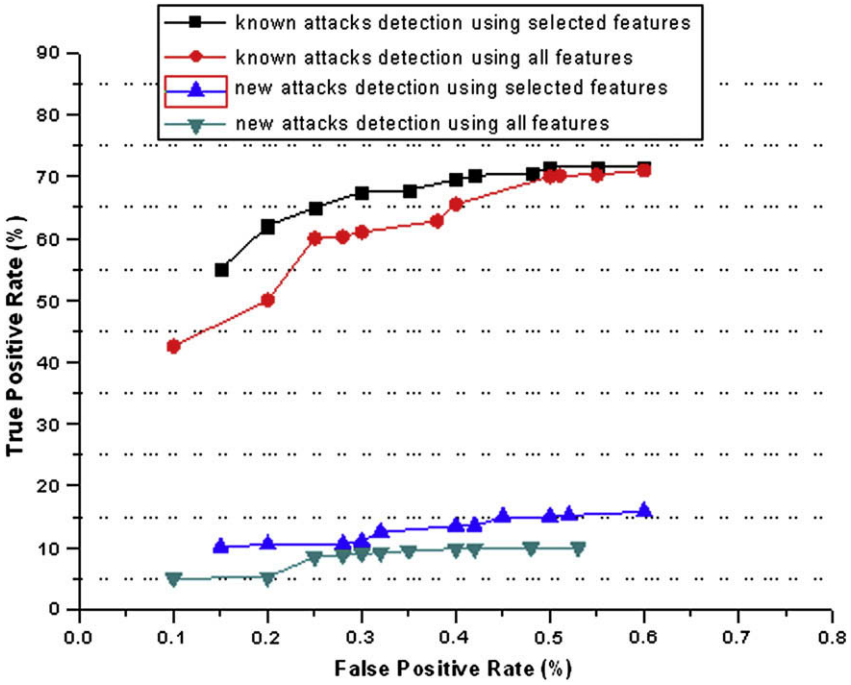

Fig. 5 – ROC curves for R2L attack detection.

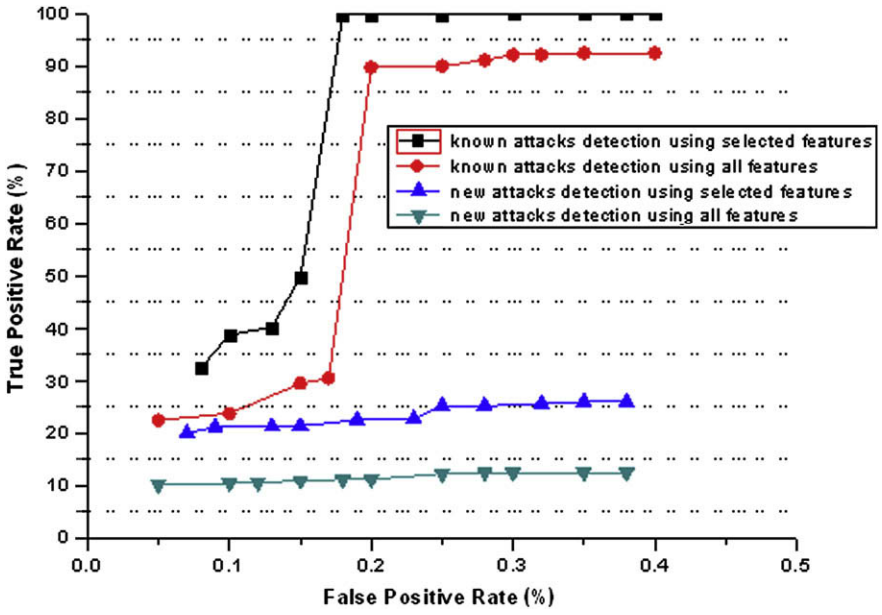

Fig. 6 – ROC curves for U2R attack detection.

nodes consist of linear SVMs. Firstly, we selected the best feature subsets by using our own feature selection algorithm through the above five sampled training datasets. Secondly, for each sampled training dataset, we built intrusion detection systems using all 41 features and selected features respectively. The detailed experimental results will be discussed in next section.

4.2. *Experimental results and analyses*

We used our feature selection algorithm to select the best feature subsets for all attacks, DOS, PROBE, R2L and U2R, and the selected feature subsets were depicted in Table 1. The detailed descriptions of these selected features given in the second column in Table 1 are introduced in KDD Cup (1999) dataset. It is clear that after our feature selection methods, the useful feature subsets for each type of the attacks are greatly reduced.

As was mentioned earlier, feature selection algorithm has two main components: search strategy and evaluation criterion. Therefore, in order to test our search strategy modified RHMC, we conducted several experiments to compare the time of selecting processes between modified RHMC and RHMC. Table 3 shows the consuming time of feature selecting processes using two different feature selection algorithms for five types of attacks. It demonstrates that for search strategy, modified RHMC has the fastest process speed. For U2R attacks, the processing time of RHMC is 1.5 h, and that of modified RHMC is only 1 h, nearly 50% faster than RHMC.

After selected five best feature subsets by using our own feature selection algorithm, we then built several intrusion detection models on the sampled training datasets using the above five feature subsets and all 41 features. For each sampled training dataset, we built intrusion detection models and then compared the models using only the selected feature subset with those using all 41 features in several aspects: average building time, average testing time, as well as ROC (Receiver Operating Characteristic) curves of detecting known attacks and new attacks. The average building time and testing time of the models were showed in Table 3. Through Table 2, we can see that in each type of attacks, the model with selected features has the smaller building time and testing time than that with all 41 features. Especially for R2L attacks, the average building time for all features is 317 s, and that for selected features is only 35 s, nearly saving 90% computational cost. The obviously smaller building time and testing time for these models with selected features strongly demonstrate that feature selection algorithm can help build lightweight intrusion detection system. In the following, we will further introduce the detection rates of models with selected features and all 41 features in terms of detecting known attacks and new attacks.

Many researchers have focused on improving detection rates of intrusion detection systems through proposing efficient classifiers, and it is a very difficult problem. Few people care about feature selection algorithm in intrusion detection systems. In this paper, we put forward a new feature selection algorithm aiming at building lightweight intrusion detection system. In order to prove an intrusion detection system combined with our feature selection algorithm has higher detection rates than that without feature selection algorithm in detecting known attacks and new attacks, we also performed several experiments to compare the two different intrusion detection systems. The comparisons were depicted in Figs. 2–6.

In Fig. 2, we considered all attacks as a whole, and built two types of intrusion detection system, one type was built using all 41 features, and the other was built using selected features. For each system, we adopted two test datasets: one dataset contains known attacks which have the same attacks as that in the sampled training dataset, and the other dataset contains new attacks which do not appear in the sampled

training dataset. Through Fig. 2, we can see that intrusion detection systems with selected features have higher ROC scores than those with all 41 features in terms of detecting known attacks and new attacks. Moreover, for detecting new attacks, systems with selected features have much higher ROC score than those with all features.

In Fig. 3, we only evaluating the effectiveness of our proposed methods in detecting DOS attacks, and compared the systems using selected features with those using all features in detecting known attacks and new attacks. We can clearly see that when detecting new attacks, systems with selected features have higher true positive rates than those with all features.

From Fig. 4 to Fig. 6, we continued to conduct the same comparisons as those with DOS attacks. Fig. 4 presents the ROC curves for detecting PROBE attacks, Fig. 5 for R2L attacks and Fig. 6 for U2R attacks. In Fig. 4, although systems with selected features have lower true positive rates in detecting new attacks than those with all features in some parts, as a whole, they have higher ROC score. Figs. 5 and 6 also show that systems with feature selection algorithm have higher ROC score than those with no feature selection algorithm in detecting known attacks and new attacks.

5. Conclusions

Existing studies to build lightweight IDS have proposed two main approaches: parameters optimization of classification algorithms and feature selection of audit data. In this paper, we proposed a novel wrapper-based feature selection algorithm to build lightweight IDS. Our feature selection algorithm consists of search strategy—modified RMHC and evaluation criterion—modified linear SVMs. We adopted modified RMHC to speed up the wrapper method to solve the computational complexity. In order to select a best subset of features for classifier, we adopted modified linear SVMs to evaluate the selected subset. We developed a series of experiments on KDD Cup (1999) intrusion detection dataset to examine the effectiveness of our feature selection and its efficiency in building lightweight IDS. The experiment results show that our approach is not only able to speed up the process of selecting important features but also to yield high detection rates for IDS.

In our future work, we will further improve our feature selection algorithm on search strategy and evaluation criterion to help build efficient and lightweight intrusion detection system.

REFERENCES

Amor NB, Benferhat S, Elouedi Z. Naive bayes vs. decision trees in intrusion detection systems. In: Proc. of the 19th ACM Symposium on Applied Computing (SAC); 2004.

Arreola Karina Zapien, Fehr Janis, Burkhardt Hans. Fast support vector machine classification using linear SVMs. In: The 18th international conference on pattern recognition 0-7695-2521-0/06; 2006.

Bennett KP, Cristianini N, Shawe-Taylor J, Wu D. Enlarging the margins in perceptron decision trees. Machine Learning 2000; 41(3):295–313.

Cao LJ, Chua KS, Chong WK, Lee HP, Gu QM. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. Neurocomputing 2003;55:321–36.

Grandvalet Y, Canu S. Adaptive scaling for feature selection in SVMs. Advances in Neural Information Processing Systems 2003;15:553–60.

Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. Machine Learning 2002;46(1–3):389–422.

Jain AK, Zongker D. Feature selection: evaluation, application, and small sample performance. IEEE Trans. on Pattern Analysis and Machine Intelligence 1997;19(2):153–8.

KDD Cup. Data available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html; 1999.

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, editors. Advances in Neural Networks. Lecture Notes in Computer Science, vol. 3498. Berlin Heidelberg New York: Springer-Verlag; 2005. p. 415–20.

Kirkpatrick K, Gelatt CD, Vecchi MP. Optimization by Simulated Annealing. Science 1983;220(4598).

Kohavi R, John GH. Wrappers for feature subset selection. Artificial Intelligence 1997;97:273–324.

Koller D, Sahami M. Toward optimal feature selection. In: Proceedings of international conference on machine learning, Bari, Italy; 1996. p. 284–92.

Kudo M, Sklansky J. ‘‘Comparison of algorithms that select features for pattern classifiers. Pattern Recognition 2000;33(1):25–41.

Lanckriet GRG, Critianini N, Barteltt P, Ghaoui, Jordan MI. Learning the kernel matrix with semidefinite programming. Journal of Machine Learning Research 2004;5:27–72.

Lee Y, Mangasarian O, editors. RSVM: Reduced Support Vector Machines. Chicago, Philadelphia: 2001 SIAM International Conference; 2001.

Park J, Shazzad, Sazzad KM, Kim D. Toward modeling lightweight intrusion detection system through correlation-based hybrid feature selection. In: Feng D, Lin D, Yung M, editors. Information Security and Cryptology. Lecture Notes in Computer Science, vol. 3822. Berlin Heidelberg New York: Springer-Verlag; 2005. p. 279–89.

Ribeiro BM. Model selection for kernel based intrusion detection systems. In: Proc. of Int. Conf. on Adaptive and Natural Computing Algorithms. Springer-Verlag; 2005. p. 458–61.

Skalak DB. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In: Proc. of the eleventh international conference on machine learning; 1994. p. 293–301.

Vapnik V. The Nature of Statistical Learning Theory. New York: Springer Verlag; 1995.

Vapnik V. Statistical Leaning Theory. New York: Wiley-Interscience Publication; 1998.

Wolf Lior, Shashua Amnon. Feature selection for unsupervised and supervised inference: the emergence of sparsity in a weighted-based approach. Journal of Machine Learning Research, in press.

Yang Li received his Ph.D. degree in Institute of Computing Technology, Chinese Academy of Sciences in 2008. He is now serving as a researcher and project manager in China Mobile Research Institute (CMRI). His current research interests include network security, intrusion detection, anomaly detection, P2P security, distributed

system security, security in next generation network, etc. He has published more than 30 high quality research papers in many distinguished international conferences and journals, such as SIGCOMM 2008, Computers & Security, Computers Communications, etc.

Jun-Li Wang received her Bachelor's degree in Hunan Normal University in 2005. She is an educational researcher in Peking University Founder Technology College.

Zhi-Hong Tian received his Ph.D. degree in Harbin Institute of Technology in 2006. He is pursuing Postdoc in Institute of Computing Technology, Chinese Academy of Sciences.

Tian-Bo Lu received his Ph.D. degree in Institute of Computing Technology, Chinese Academy of Sciences in 2006. He is now serving as a researcher and project manager in National Computer network Emergency Response Technical Team/Coordination Center of China.

Chen Young is now pursuing his Ph.D. degree in Chinese Academy of Sciences. He specializes in traffic measurement and network security.