Toward Building Lightweight Intrusion Detection System Through Modified RMHC and SVM

You Chen[1,2], Wen-Fa Li[1,2], Xue-Qi Cheng[1]
[1]Institute of Computing Technology, Chinese Academy of Sciences
[2] Graduate University, The Chinese Academy of Sciences
{chenyou,liwenfa,cxq}@software.ict.ac.cn

***Abstract*- Feature selection attracted much interest from researchers in many fields such as network security, pattern recognition and data mining. In this paper, we present a wrapper-based feature selection algorithm aiming at modeling lightweight intrusion detection system (IDS) by (1) using modified random mutation hill climbing (MRMHC) as search strategy to specify a candidate subset for evaluation; (2) using support vector machines (SVMs) as wrapper approach to obtain the optimum feature subset. We have examined the feasibility of our feature selection algorithm by conducting several experiments on KDD 1999 intrusion detection dataset which was categorized as DOS, PROBE, R2L and U2R. The experimental results show that our approach is able not only to speed up the process of selecting important features but also to guarantee high detection rates. Furthermore, our experiments indicate that intrusion detection system with a combination of our proposed approach has smaller computational resources than that with GA-SVM which is a popular feature selection algorithm in the field.**

I. INTRODUCTION

Intrusion detection system (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 or reducing these irrelevant or redundant features is very important because they may deteriorate the performance of classifiers. Furthermore speaking, by choosing effective and important features, we can improve the comprehensibility of 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 [1]. Methods for feature selection have been essentially divided into two categories: filter methods and wrapper methods [2].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 independently of the classifier and use a 'goodness' metric to decide which features should be kept. Because of their use of the classification results 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 [3] which combine wrapper and filter methods. However, in [3] the number of selected features is large and the performances of intrusion detection system which based on hybrid feature selection are not perfect.

In [2][4], the differences for above three different feature selection algorithm were depicted, and the wrapper feature selection algorithm performs better than other two methods. In [5], it proposed a wrapper method (GA-SVM) to optimize SVM based IDS through operation of GA. It simultaneously enabled one not only to select "optimal features" but also to figure out "optimal parameters" for SVM classifier. But in [5], it cost much time to select features, and then results a slow selecting process. Furthermore, it used detection rate as the criterion of evaluating IDS. As we know, detection rate is not a sufficient evaluation criterion for IDS, and for evaluation criterion of IDS, ROC curve is far superior to detection rate. In this paper, we adopted several methods to speed-up the wrapper method to solve the computational complexity. Experiment results show that our approach is able not only to speed up the selecting process, but also to have high ROC scores on detecting known attacks and new attacks.

Much of the existing research focuses on the achievable accuracy of different machine learning algorithms. The studies have shown that a number of different algorithms are able to achieve high classification accuracy [6]. There have been no comparisons of the relative speed of classification for different algorithms when classifying abnormal flows. However, within a practical IDS system, a consideration as to computational performance is vitally important.

In this paper we attempt to provide some insight into these aspects. Our key findings are:

(1)Feature reduction greatly reduces the number of features needed to identify abnormal attacks and hence greatly improves computational performance

(2)While feature reduction greatly improves performance it does not severely reduce the classification accuracy.

(3)IDSs with combination of feature selection algorithm have bigger capability of detecting attacks, especially for detecting new attacks.

II. RELATED WORK

There are two important parts in feature selection, one is search strategy, and the other is evaluation criterion. For search strategy, Jain and Zongker [7] found that the heuristic methods (forward sequential search) performs, best in large

datasets, while Kudo and Sklansky [8] found the random methods (genetic algorithms, hill climbing) are superior for large-scale problems. For massive-scale feature selection problems these popular methods can be too computationally demanding for practical training times. In this paper, we introduced another random search method named random mutation hill climbing (RMHC) which can be enhanced in terms of its speed by adopting methods from simulated annealing, while its ability to dramatically reduce the feature count can be easily improved. For evaluation criterion, support vector machines (SVMs) have become a tool in recent years due to their remarkable characteristic such as the absence of local minima, the sparse representation and good generalization ability. SVM based IDSs need to not only increase detection rates but also guarantee the stability for detection rates. In order to provide these properties, one has to optimize the parameters for kernels in SVM. This paper proposes a fusion method to optimize SVM based IDS through operation of MRMHC.

III. PROPOSED APPROACH

A. Feature Selection Algorithm based on MRMHC and SVMs

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 [9]. 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 [9]. 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 [9].The algorithm operates as follows:

(1) Initialize a binary string, S, of length N, where M features are marked as used,'1' and the remaining N-M are '0'.

(2) Test binary string, S, for fitness F(S) using the probability of classification error.

(3) Randomly mutate M bits in the binary string, S.

(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, and in this paper, the classification error is obtained by SVMs :

$$F(S) = P_{error}(S) = \frac{1}{C}\sum_{i=1}^{C}\gamma_i \qquad (1)$$

where S is the set of currently utilized features, $P_{error}(S)$ is the current average error rate, C is number of class, and γ_i is the error rate for ith class.

We modified 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 [10]. 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:

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

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 average 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 error. Note 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:

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

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.

B. Lightweight IDS based on Feature Selection Algorithm

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 SVMs. The metric of the evaluation is $P_{error}(S_{best})$ which represents the best feature subset S_{best} .After initialing the values of S_{best} and $P_{error}(S_{best})$, the approach goes into an iterative procedure. In each iterature, 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 MRMHC is evaluated by SVMs in an iterative way. If $P_{error}(S)$ is lower than $P_{error}(S_{best})$, it is assigned as $P_{error}(S_{best})$ and the approach goes forward. The approach stops if a predefined stopping criterion δ is reached or when

maximum number of iterate I_{max} 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 IDS which will be evaluated on test dataset in terms of building time, testing time, true positive rate and false positive rate.

In our method, MRMHC is adopted to obtain the set of features as well as the optimal parameters for a kernel function. MRMHC creates improved detection models containing a set of features and parameters by the iterative process of hill climbing, evaluation, and selection process. At the end of learning step, this method builds the optimal detection model consists of a set of features and parameters for a kernel function. The optimal detection model is used to classify new pattern samples in classification step [11]. To evaluate the feasibility of our approach, several experiments are carried out using KDD 1999 intrusion detection dataset [12]. The following section presents the results of experiments and their analysis.

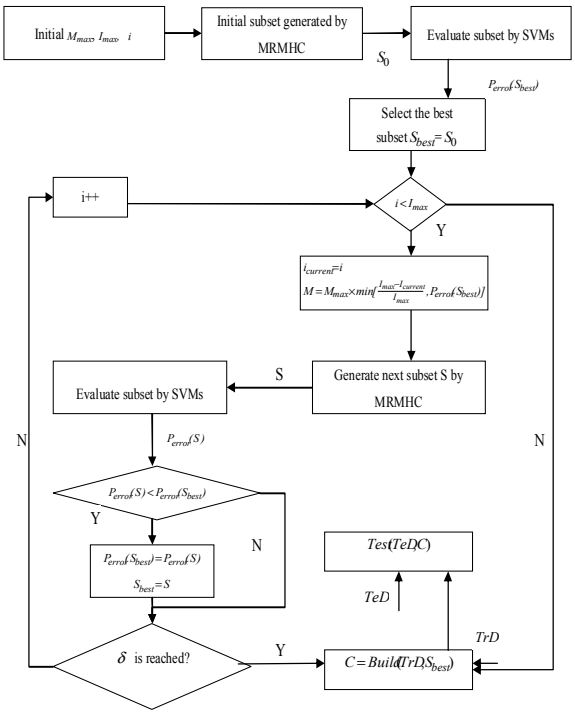

Figure 1. Flow chart of a wrapper-based feature selection algorithm toward building lightweight IDS

IV. EXPERIMENTS AND RESULTS

Our experiments were performed on KDD 1999 intrusion detection dataset [12], and it contains 41 features [12] of each instance. For KDD 1999 intrusion detection training dataset, we categorized it as DOS type of attacks, PROBR, R2L (Remote to Local) and U2R (User to Root). For KDD 1999 test dataset, we classified it as known attacks which are exist in training dataset, and new attacks which are not exist in training dataset. We conducted a statistical computation for training dataset and test dataset in terms of DOS, PROBE, R2L, U2R and NORMAL. NORMAL means no attack, and the number of NORMAL instances in training dataset and test dataset is 97278 and 60593. For each type of attacks, there are two components in test dataset, one is known attacks, and the other is new attacks. For each new attack, the number of instances in training dataset is zero.

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

A. Experiment schema on KDD1999 intrusion detection dataset

We have preprocessed the KDD 1999 labeled training dataset to make it five class dataset—NORMAL, DOS, PROBE, R2L and U2R. The dataset contains total 494,019 instances, among these 97,278 (19.69%) instances are normal and 396,741 (80.31%) belongs to 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 established five training datasets from KDD 1999 intrusion dataset. We combined 97278 normal instances with 391,458 DOS instances, 4107 PROBE instances, 1126 R2L instances and 52 U2R instances respectively, and then we sampled four datasets each has 11701 instances, from the above four combined datasets by uniform random distribution so that the distribution of the datasets should remain unchanged. Meanwhile, we also sampled one dataset named "ALL" which have the same 11701 instances, from the total training dataset in KDD1999 by uniform random distribution. Each instance of above five sampled dataset consists of 41 features. In order to evaluate the performances of our approach in terms of detecting known attacks and new attacks, we have preprocessed the KDD 1999 labeled test dataset to make it two class dataset—known attacks and new attacks for each type of attacks. The test dataset contains total 311029 instances, among these 229853(73.9%) instances are DOS and nearly 2.9% instances of DOS have new attacks, which are not exist in forward training dataset. For each type of attacks, we sampled two different datasets from KDD 1999 test dataset. One is named as known attacks, and the other is named as new attacks. These sampled test datasets were used for evaluating intrusion detection systems.

The overall structure of our approach is depicted in Fig1. We developed several experiments based on Fig1 to exam the flexibility of our approach. 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 two different types of intrusion detection systems, one used all 41 features and the other used selected feature subset. The detail of experimental results will be introduced in next section.

B. Experimental Results and Analysis

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 Table1. In last column of Table1, we can see the sequence numbers

and names of each selected feature subset. Before the colon are the sequence numbers of selected features in KDD1999 whose largest sequence number is 41, and after the colon are the corresponding names of forward sequence numbers. Detail explanations of these names are introduced in KDD1999 [12].

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

Attack type	Selected features
ALL	3,5,23,33,34 : service, src_bytes, count, dst_host_srv_count, dst_host_same_srv_rate
DOS	5,12,23,34 : src_bytes, logged_in, count,dst_host_same_srv_rate
PROBE	1,3,5,23,37 : duration, service, src_bytes, count, dst_host_srv_diff_host_rate
R2L	1, 5,6 : duration, src_bytes, dst_bytes
U2R	1,3,6,14,33 : duration, service, dst_bytes, root_shell, dst_host_srv_count

Feature selection algorithm has two main components, one is search strategy and the other is evaluation criterion. In order to test the ability of our search strategy--MRMHC, we conducted several experiments to compare the time of selecting processes between MRMHC and GA. Table2 shows the time of feature selecting processes using two different feature selection algorithms for five types of attacks. It demonstrates that for search strategy, MRMHC has a fast process speed. For U2R attacks, the selecting time of GA is 1.5h, and that of modified RHMC is only 0.6h, nearly 40% of GA.

Table2 Time of selecting processes for different feature selection algorithms

Algorithm	ALL (h)	DOS (h)	PROBE (h)	R2L (h)	U2R (h)
GA-SVMs	1.3	0.5	4	1.5	1.5
MRMHC-SVMs	0.4	0.2	2.2	0.8	0.6

Table3 Average time of building and testing processes with all features and selected features for ALL attacks, DOS, PROBE, R2L and U2R

—		ALL	DOS	PROBE	R2L	U2R
Build (s)	all	78	136	245	317	193
	select	30	31	96	24	78
Test (s)	all	18	22	49	55	50
	select	6	5	17	7	15

As the five best feature subsets were selected, we then built intrusion detection models on sampled training datasets using the above five feature subsets and all 41 features.

For each sampled training dataset, we built intrusion detection models using different thresholds, and then we compared the models using selected feature subset with those using all 41 features in terms of average building time, average testing time and ROC scores of detecting known attacks and new attacks. The average building time and testing time of the models were showed in Table3. Through Table3, we can see that for each type of attacks, the model with selected features has the smaller building time and testing time than that with all 41 features. For example, as to "ALL" type of attacks, the average building time and testing time for all features is 78s and 18s respectively, and that for selected features is only 30s and 6s, nearly 38.5% and 33.3% of all features. From [5] and [13], as to "ALL" type of attacks, the average building time and testing time for all features is 77s and 15s respectively, and that for feature subset selected by GA-SVMs is 60s and 10s, nearly 78% and 46.7% of all features. In [6], author introduced that computation consumption is a critical factor for classifier, and it is also adapt to IDS classifier. As a result of comparison in terms of building time and testing time, we can see that our proposed method consumed smaller computational resources than GA-SVMs. The smaller building time and testing time of our models with selected features demonstrate that feature selection algorithm can help to build lightweight intrusion detection system. In the following, we will introduce the detection rates of models with selected features and with 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 intrusion detection system with a combination of our feature selection algorithm has higher detection rates than that with no feature selection algorithm in terms of detecting known attacks and new attacks, we developed several experiments to compare the two different intrusion detection systems. The comparisons were depicted in Fig2, Fig3, Fig4, Fig5 and Fig6.

In Fig2, we considered all attacks as one type, 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 type system, we used two test datasets for testing, the one is known attacks which are exist in forward sampled training dataset, and the other is new attacks which are not exist in forward sampled training dataset. Through Fig2, 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. Especially, for detecting new attacks, systems with selected features have much higher ROC score than those with all features.

In Fig3, we only considered DOS type of attacks, and compared the systems using selected features with those using all features in terms of detecting known attacks and new attacks. From Fig3, we can see that for detecting new attacks, systems with selected features have higher true positive rates than those with all features, nearly 5% increase. It demonstrates that system using selected features has higher true positive rates of detecting new attacks than that with all features.

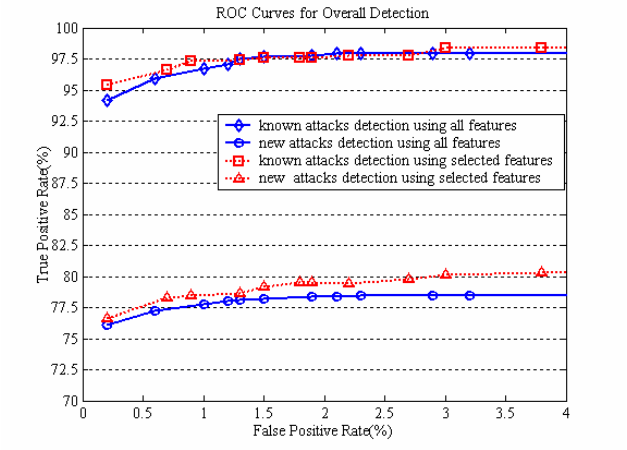

Figure 2. ROC curvers for all attacks detection

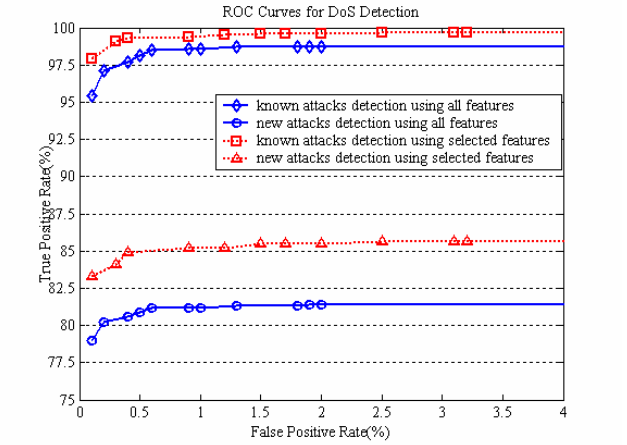

Figure 3. ROC curvers for dos attack detection

From Fig4 to Fig6, we conducted the same comparisons with DOS attacks, and Fig4 is for PROBE attacks, Fig5 for R2L attacks, Fig6 for U2R attacks. In Fig4, although systems with selected features have lower true positive rates of detecting new attacks than those with all features in some parts, as a whole, they have higher ROC score. Fig5 and Fig6 also show that systems with a combination of feature selection algorithm have higher ROC score than those with no feature selection algorithm in terms of detecting known attacks and new attacks.

As to ROC score of detecting known attacks, our proposed method is nearly equal to GA-SVMs in [5] and [13]. As to capability of detecting new attacks, at now, there is no model to compare with. However, our model has higher ROC scores than that with all features.

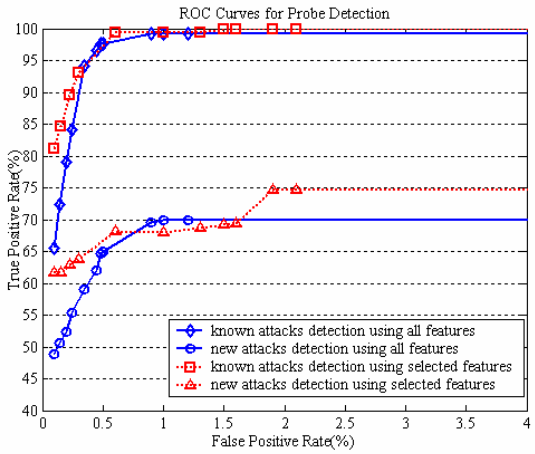

Figure 4. ROC curves for probe attack detectiom

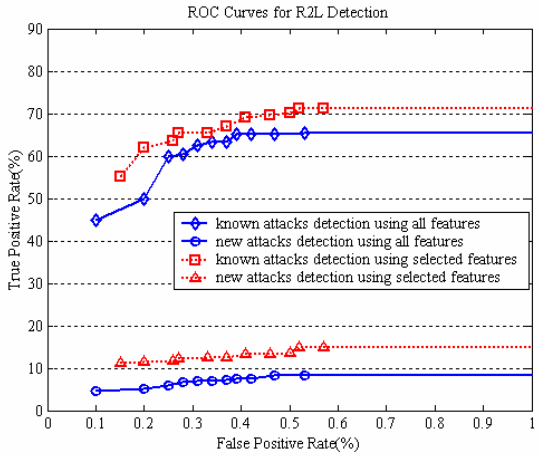

Figure 5. ROC curves for R2L attack detection

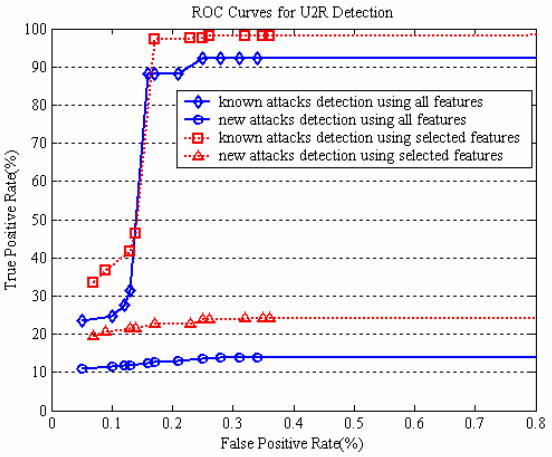

Figure 6. ROC curves for U2R attack detection

V. CONCLUSION AND DISCUSSION

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--MRMHC and evaluation criterion--SVMs. The feature selection of audit data has adopted two main methods: wrapper and filter method. Wrapper method generally performs better than

filter method, but it involves some more computational complexity and requires more execution time than the filter method due to retraining a classifier for each new set of features. In this paper, we adopted MRMHC to speed-up the wrapper method to solve the computational complexity. In our method, MRMHC is adopted to obtain the set of features as well as the optimal parameters for a kernel function. We developed several experiments on KDD1999 intrusion detection dataset to examine the flexibility of feature selection algorithm. The experiment results show that our approach is able not only to speed up the process of selecting important features but also to guarantee high detection rates. Meanwhile, we conducted several comparisons between systems with selected features and those with all features for ALL attacks, DOS, PROBE, R2L and U2R. The results of comparisons demonstrate that IDS with a combination of feature selection algorithm has higher ROC score than that with no feature selection algorithm in terms of detecting known attacks and new attacks. In our future research, we will improve our feature selection algorithm on search strategy and evaluation criterion to help building efficient and real time IDS.

ACKNOWLEDGEMENTS

This study is supported by the 973 National Basic Research Programs of China under Grant No.2004CB318109 & 2007CB311100, and the National High-Tech Research and Development Plan of China under Grant No.2006AA012452.

REFERENCES

[1] Koller, D., Sahami, M.: "Toward optimal feature selection". In: Proceedings of International Conference on Machine Learning, Bari, Italy,.284-292,1996

[2] You Chen, Yang Li, Xue-Qi Cheng and Li Guo. Survey and Taxonomy of Feature Selection Algorithms in Intrusion Detection System. Conference on Information Security and Cryptology. LNCS4318，153-167,2006

[3] J. P. Thomas S. Chebrolu, A. Abraham. Hybrid feature selection for modeling intrusion detection systems. In Proceedings of Lecture Notes in Computer Science, volume 3316, 1020-1025, 2004

[4] You Chen, Lei Dai, Yang Li, Xue-Qi Cheng. Building Efficient Intrusion Detection Model Based on Principal Component Analysis and C4.5 Algorithm. 9th IEEE International Conference on Advanced Communication Technology. 2109-2112,2007

[5] 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 415–420, 2005.

[6] Nigel Williams, Sebastian Zander and Grenville Armitrage, A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification. ACM SIGCOMM Computer Communication Review, Volume 36 5-13,2006

[7] A.K. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, no.2, 153-158, 1997.

[8] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers", Pattern Recognition, vol. 33, no. 1, 25-41, 2000.

[9] D. B. Skalak, "Prototype and feature selection by sampling and random mutation hill climbing algorithms", Proc. of the Eleventh International Conference on Machine Learning, 293-301, 1994.

[10] K. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, Optimization by Simulated Annealing, Science, vol. 220, no. 4598, 1983.

[11] Karina Zapien Arreola, Janis Fehr, Hans Burkhardt: Fast Support Vector Machine Classification using linear SVMs, The 18th International Conference on Pattern Recognition 0-7695-2521-0/06, 2006

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

[13] Chen You. Xue-Qi Cheng, Yang Li, Lei Dai. Lightweight intrusion detection systems based on feature selection. Journal of Software,18(7):1639-1651,2007