International Journal of Security and Its Applications
Vol.8, No.6 (2014), pp.185-196
http://dx.doi.org/10.14257/ijsia.2014.8.6.17

The Construction Research of Security Computer Network System Based on the Distributed Intrusion Detection Technology

Xin Huang and Rongze Wan

Guangxi Agricultural-vocational Technique College, Nanning Guangxi 530007, China

Abstract

This paper aimed at the actual situation of the difficult of getting a lot of the training sample of the security computer network system in the distributed intrusion detection. In this paper, we studied how to increase the intrusion detection accuracy in the case of small samples, so that processing, maintenance and deal with the invasion of the network timely. In this paper, we proposed a new intrusion detection method based on improved SVM Co - training. The specific implementation process of the algorithm is presented. Through the simulation experiments based on the actual data showed that the method is effective. Apply this method to a classified computer network system, effectively realized the detection to outside intruders and internal intruder.

Keywords: *Security computer, Intrusion detection, Distributed, The SVM collaborative training*

1. Introduction

With the developing of increasing demanding of computer security, traditional firewall and single port intrusion detection technology cannot fulfill the demand of defending the intrusions and the distributed intrusion detection technology becomes an important research area. In this paper, we have done some researches on the key technology of agent based high performance peer-to-peer distributed intrusion detection system, including the technology of distributed intrusion detection, theory of agent, high performance still agent for networking packets, high performance still agent for computer audit data, peer-to-peer distributed intrusion detection model, high performance model extension, and et al. Based on the above research contents, we have designed a system, called APDIDS, with the name of Agent based P2P distributed intrusion detection System.

Network security is becoming a very important issue affecting the reliability of the network computer. The security of company intranet will be at risk if the public can access the network host or server via the Internet, An exposed internal network is very easy to be invaded by unauthorized users. In the past, most of the internal network is under closed network protection before they can become part of the network used by isolated people only. Gateway firewall is commonly used in front-end so that it can repel malicious attacks from the outside, whose premise is that the gateway trusts all network nodes without trusting all external hosts.

With the development of technology, hacking, computer attacks from the outside network can easily penetrate the gateway, thus becoming a threat to intranet. A domino effect will occur if a node inside the network is captured, so that all internal network hosts may soon be captured. Here, we propose a method, called Distributed Micro-

ISSN: 1738-9976 IJSIA

Firewall, to solve the network security threats faced by all the network nodes. Workstation or server can be seen as a network node in a Local Area Cluster formed by computer network and gateway firewall is mainly used to protect the Local Area Cluster to avoid being attacked from the outside.

The research work of literature [1] focuses on systems and structures of firewall gateway. The literature [2] proposes a Mobile Agents for computer network security, while holds a discussion on the software tools for the development of firewall Intrusion Detection System. In the literature [3] Bellovin pioneered the concept of Distributed Firewall. Literature [4] noted that the University of Pennsylvania and AT & T Labs is developing a prototype for Computer Security Systems. Literature [5] provides another distributed approach for the Intrusion Detection and Isolation Protocol (IDIP) project. Literature [6] pointed out that the Purdue Coast Project conducting the Agent-based Intrusion Detection tests, while also discussing the security issues of the proxy computer.

Distributed intrusion detection system collects data based on sensors, and send data to still agent (also, local agent) for analysis. Still agents are generally placed on the different inner places of sub-net to monitor them. With the development of network bandwidth, still agent need to work on 1 Gpbs network or even more. It is hard for traditional distributed intrusion detection systems to catch up with the line speed.

To protect the important computer, many audit data are generating from different parts of it for future auditing. Another kind of still agents is used to analyze this kind of audit data. To middle or large scale network, the important computers are always "Web Server", "Mail Server" like computers, which have many concurrent accessing and generate huge audit data. If this kind of still agent runs on the important computer, the computing requirement for analyzing audit data will cause many resources and further influence the services provided by the server.

To solve this problem, the conception of distributed computing is used. This paper proposed a still agent model based on distributed data auditing. With this model, several computers are used to assist the important computer to analyze the audit data. Based on the idea, we present another still agent model based on distributed middleware.

A classified computer network is vulnerable to external and internal attacks in the Internet environment, the attacks' data, however, is difficult to obtain. In such cases, this paper carried out a study of network intrusion detection. In this paper, the intrusion detection algorithm is researched when the data's amount is small in order to improve the detection accuracy, so that the network intrusion can be promptly processed. An intrusion detection method called SVM Co-training based on improvements is proposed; whose practical application does improve the security level of a classified computer network system.

2. Some Definitions on Issues

The issues, *i.e.*, Distributed Intrusion Detection of classified computer network, can be described by the following procedure. In this paper, Semi-Supervised Learning approach is used to find the best Intrusion Detection Technology for classified computer network system in the sample data set D.

For a given sample set of classified computer network system messages, $D = A \cup B$, among which the message sample data set is divided into two kinds: one is labeled, represented as $A = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$; one is unlabeled, be represented

by $B = \{x_1^{'}, x_2^{'}, ..., x_m^{'}\}$. The total number of sample data in the two mentioned samples is respectively represented by $n = |A|$ and $m = |B|$, usually, the number of unlabeled is much larger than the number of the labeled, that is $m \gg n$.

The ultimate goal of intrusion detection problem is to deal with Intrusion Samples, represented as x which is d -dimensional vector of Classified Computer Network. By appropriate Intrusion Detection Algorithm, its predicted label result can be calculated, represented as y, which is the label values of x ,the sample data sets. The specific features of the algorithm is to create a function of sample data sets and labeling outcomes: $F : D \to Y$.

If the relationship between the sample data sets D and labeled data set A is $D = A$, then the question becomes a simple Traditional Supervised Learning problem; conversely, if the relationship between the sample data set D and unlabeled data set B is $D = B$, then the question becomes simple Unsupervised Learning problem. Semi-supervised Learning problem should focus on how to effectively make use of and data sets B. This article focuses on the Semi-supervision Algorithm at the condition that the data entries in data set A are much smaller than the data set B (ie: $n \ll m$).

3. Semi-supervised Learning Algorithm based on Improved SVM

It is difficult to obtain a complete set of intrusion detection training samples from the classified computer network system in reality. Therefore, for this situation: a small sample size, but unlabeled sample data is much larger than the labeled sample data, that is, $m \gg n$, this paper attempts to study the appropriate intrusion detection algorithms to improve the accuracy of intrusion detection. In the process of network intrusion detection, unlabeled data is very easy to obtain, but the labeled data has been difficult to get. In the field of Semi-supervised Learning, Co-training algorithm is a very important branch, which requires the independent data with each other. Moreover, these data can, respectively, represent their corresponding Characterized Data Set, and each set can get to the corresponding classifier. Different classifier can predict the unlabeled data through which the diversity of samples can be achieved, and thus the prediction accuracy can be improved. Currently, Co-training technique has been widely used in many areas, such as the detection of text processing software, web page classification and tracking of targets. In this paper, with the combination of Support Vector Machine (SVM), an improved SVM Co-training algorithm is proposed, and its application to Distributed Intrusion Detection in classified computer network is on plan.

Definition 1: The proposed SVM Classifier, with variation of (α, β), is an improved SVM Co-training Algorithm, which can be used as Co-training Algorithm of two base classifiers.

The core of Co-training method: to utilize the sample data has been marked and unmarked, resulting in two separate classifiers, and to constantly improve the classification accuracy of the classifier through continuous iterative learning and constant training. In the process of classifier's iterative learning, an improved SVM Co-training algorithm proposed in this paper is used to make the dimensions of computing spaces are reduced, and the classification results can be improved. Seeing in Figure 1, it is a schematic diagram of Improved Co-training algorithm based on SVM.

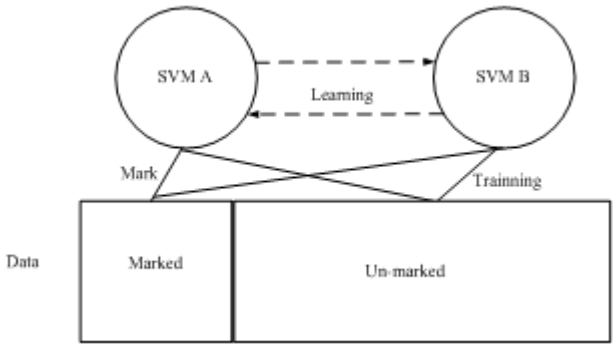


Figure 1. Schematic Diagram of Improved Co-training Algorithm based on SVM

The core of SVM-based Co-training algorithm is: the two classifiers' classification accuracy can be improved respectively though the two classifiers' iterative and continuous training. Specific implementation process of SVM Co-training algorithm is as follows:

Step1: Intrusion Sample Data Set x is divided into two independent feature sets x^1 and x^2;

Step 2: In the labeled sample data set, combining the training classification algorithms of SVM, the two initial classifiers were obtained focused on x^1 and x^2;

Step 3: For the invasion sample set D, to iterate as these following steps until the algorithm termination condition occurs:

Step 4: to use two classifiers, F_1 and F_2,obtained in Step2, in the unlabeled sample data sets, the tagged data sets B^1 and B^2 are obtained by predicting ;

Step 5: to obtain sample data set $(A+B^1)$ and $(A+B^2)$, and combine SVM's respective training, to obtain new classifiers F_1' and F_2'.

Step 6: According to the new classifiers F_1' and F_2, to reclassify the Intrusion Detection Data.

Step 7: After the data classification, to conduct the data preprocessing (removal of unreasonable data, data normalization).

Step 8: Set the algorithm parameters: maximum number of iterations, iteration time and the error rate.

Step 9: To execute the algorithm.

To set the termination condition to the algorithm as: the maximum number of iterations is k, or the Mean Square Error predicted by newly generated classifiers is less than α,the given value[7-8].

Combining the Co-training algorithm's compatibility, we can draw the following two conclusions through the analysis of the algorithm:

Conclusion 1: To randomly select a sample set consisting of m samples from Intrusion data sets, PAC is learnable if it satisfies equation (1).

$$m \geq \frac{2}{\varepsilon^2(1-2\eta)}\ln\left(\frac{2N}{\delta}\right) \tag{1}$$

Conclusion 2: If the minimized empirical error PAC is learnable, feature sets x^1 and x^2 will satisfy independence criteria. Assuming that only weak-useful learning device is given when initialization, Co-training algorithm can also obtain the PAC via continuous iterative learning simply using the unlabeled samples in sample set.

Conclusion 2 is a powerful conclusion, it showed that: as long as the data sets x^1 and x^2 satisfy certain independence conditions, a weak learner, who is obtained by learning from the samples been tagged, can get up to any arbitrary precision through the improved algorithm in this paper by making use of unlabeled sample.

4. The Actual Data Simulation

4.1. The Algorithm Pseudo Code

In order to verify the validity and accuracy of the algorithm proposed in this paper, Java programming language achieved a co-training SVM algorithm proposed in this paper, while the simulation experiments on Distributed Intrusion Detection is conducted towards classified computer network [9-10].

The pseudo-code of algorithm is as follows:

```
public class Co-training SVM
{
  /************
  **** SVM Co-taring algorithm
  ***********/
  public int Co-training SVM (Charactor  Vec Vector)
  {
    CharactorVec Vec1, Vec2:
    splitVector(Vec1, Vec2ector, Vec1, Vec2):
    Classfir fl = svm(Vec1, A);
    Classfir f2 = svm(Vec2, A);
    while(! Reach_k)
    {
      B1 = Fl. Dataforecast (B):
      B2 = F2. Dataforecast(B):
      Fl = svm(Vec1, A+Bl):
      F2 = svm(Vec2, A+B2):
    }
    return 1;
  }
}
```

4.2. Parameter Setting

10,000 training data and 50,000 test data are randomly selected. Among the 50,000 test data set, a normal part of the user's access to 64.5%; among the intrusion data, the invasion data is marked of 18.5%, and the unlabeled data accounts for 17%.

To develop SVM's base functionality using Libsvm's open source as Basic program. Based on the Improved Co-training SVM algorithm above developed by Java language, the corresponding simulation is done.

Select the data classifier C-SV. Where, C=200 is the penalty parameter. Iteration's termination conditions: maximum number of iteration k, or error accuracy is less than a given value. Other parameters in simulation are set as the given values of underlying Libsvm program. The C-SVM algorithm's two variability factor, (α, β), are set to 10% and 45% of the current samples' diameter.

4.3. Simulation Results

The results are shown in Figure 2: X-axis represents the number of data entries of Test Data Set (unit: ten thousand). As can be seen from the Figure, the detection accuracy of the proposed Improved SVM Co-training is approximately 7.68% and 5.1% higher respectively than the selected C-SVM algorithm and basic SVM algorithm. With the increase of Test Data Sets, SVMSVM Co-training's prediction accuracy increases faster than C-SVM algorithm and basic SVM algorithm, showing the effectiveness of SVM Co-training [11-12].

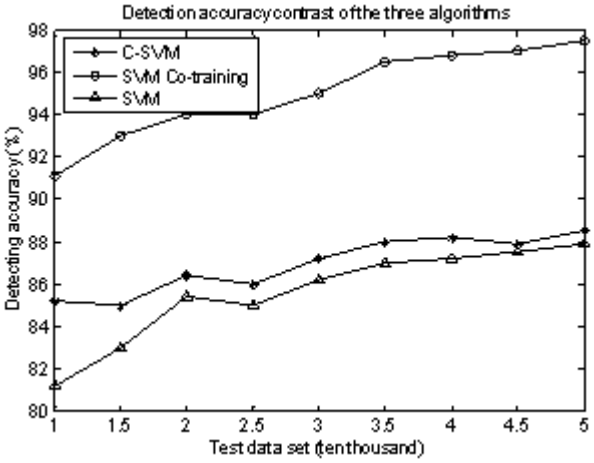


Figure 2. Changes in Detection Accuracy of the Three Algorithms with the Change in the Test Data Set

Figure 3 shows three algorithms' reliance on labeled Training Sample Set. As can be seen from the Figure, the dependence of C-SVM algorithm is higher than improved SVM Co-training algorithm and basic SVM algorithm in this paper. The simulation uses 10,000 training data, which is divided into 10 parts. The x-axis represents corresponding data of the training set (unit: ten thousand), y-axis represents the detection accuracy. As can be seen from Figure 3, C-SVM algorithm and basic SVM algorithm own larger dependence on labeled Training Sample Set, and with the reduce of the sample set the detection accuracy decreased significantly; SVM Co-training algorithm owns smaller dependence, with the reduction of the sample set, the detection accuracy does not decline obviously.

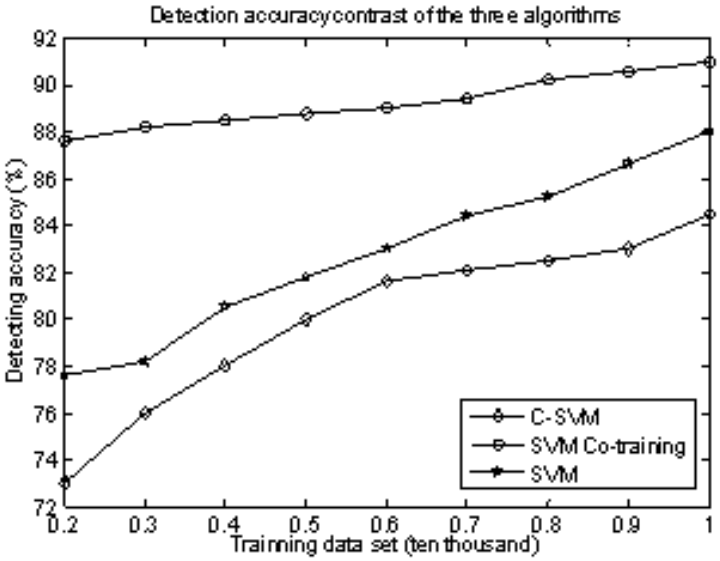


Figure 3. Changes in Detection Accuracy of the Three Algorithms with the Change in the Training Data Set

5. Classified Computer Network System based on Distributed Intrusion Detection Technology

In this section, we present the components of mainly communication and computing models designed efficiently for the Distributed Intrusion Detection System of classified computer network [13].

This model is a combination of mobile agent model and the human immune system model inspired by models based on biotechnology. In the model presented in this paper, the method for distributed Intrusion Detection System is used for monitoring. The main features of IDS model are as follows:

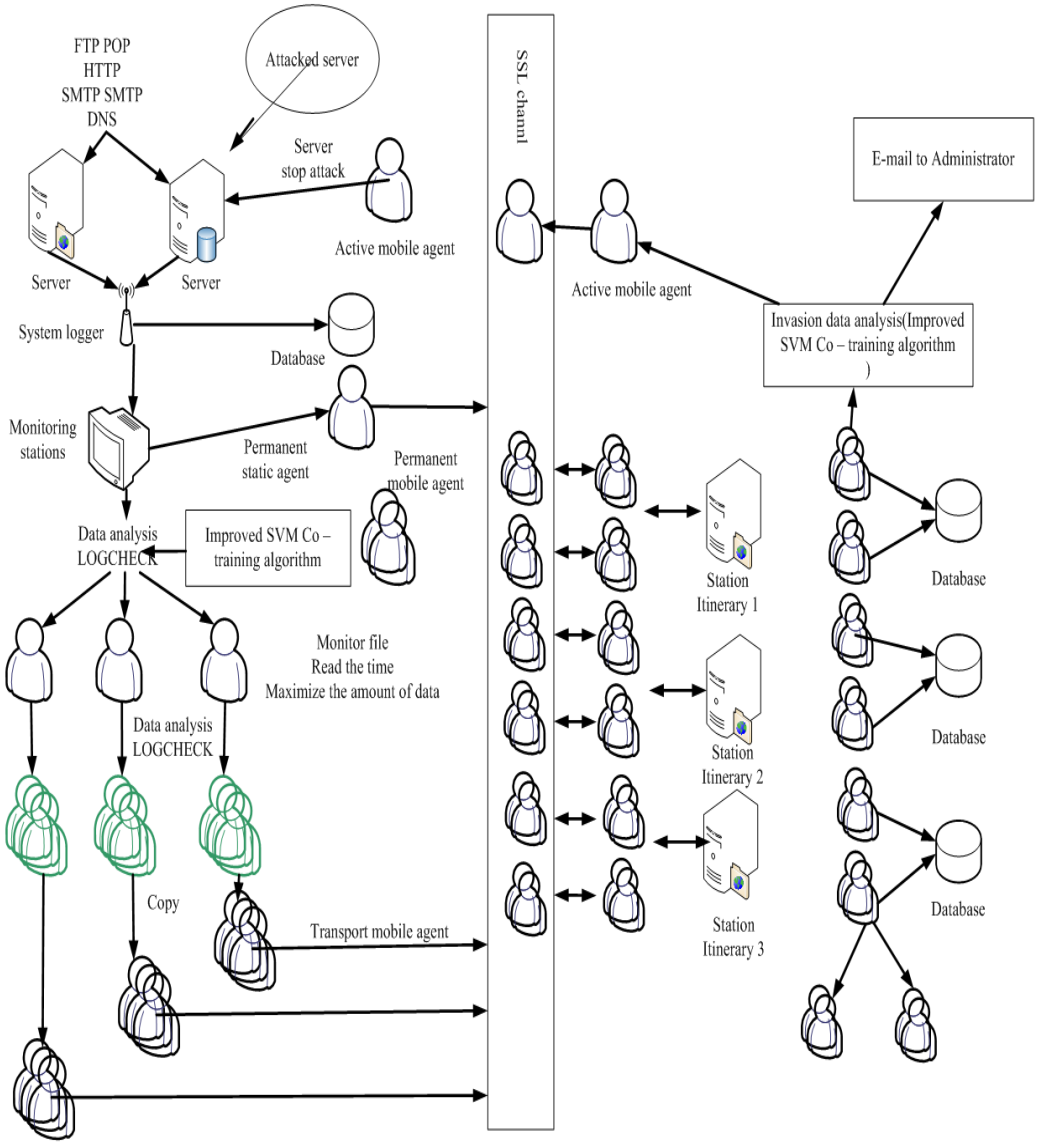
(a) Anomaly Detection Model. The model uses the Log check. It will automatically monitor the system log. Anomaly Detection Model will contribute to the classification of the monitoring service request based on response, dividing these requests into normal events or activities, unusual events or activities.

(b) Agent-based and host-based distributed architecture. Host-based Intrusion Detection System can identify and analyze server's activity logs. Because these services may be distributed in the network intrusion detection's or other relevant log files, so they must be separated. In the application of this article, we registered a log over the network and choose to use syslog-ng tools and mobile agent methods to publish log.

(c) Response Generation Component. When the invasion occurred, the main idea is: to improve the response speed of the IDS model's components. In the design of this paper, the two types of intrusion detection are considered: network attacks based on the view or the monitored services.

Figure 4 shows the major components of IDS model proposed, including not only FTP, DNS, HTTP, POP3, eSMTP services, while also containing a syslog-ng tool on the server side that can detect intrusion information and produce invasion log to get the different servers' registration activities and determine the appropriate Event Generation Function that allows the analysis and recognition of intrusion events. The last component of the model is the "agent", which is mainly responsible for the safety

assurance in operation and the integrity of intrusion logs. To meet the standard CIDF (Common Intrusion Detection Framework), the reaction towards generated event can be taken through the log sequence analysis combining with the experience of the network administrators, and the reaction steps can be divided into: event generation, analysis, storage and feedback [14].



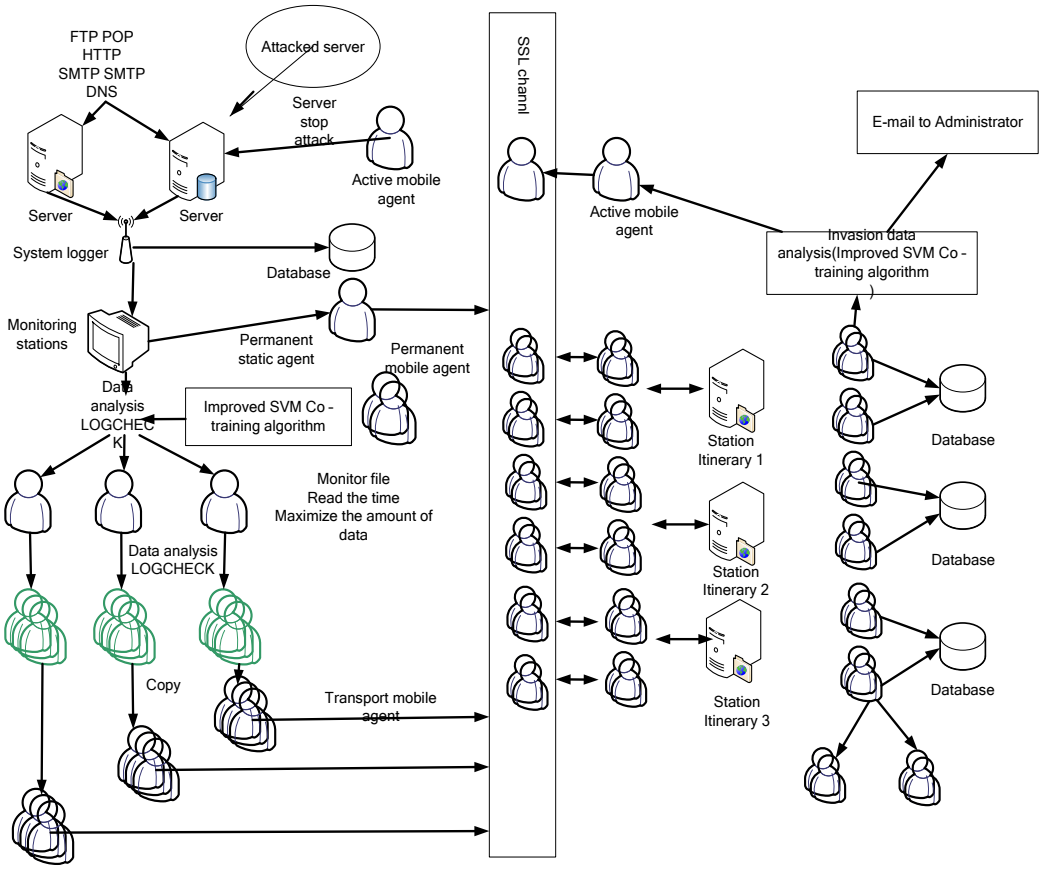


Figure 4. Classified Computer Network System based on Distributed Intrusion Detection

As shown in Table 1, the classified computer network gets Network Intrusion Detection Data after the introduction of SVM Co-training algorithm, after running for some time, been compared with the data generated by using C-SCV algorithm.

Table 1. Comparison

Algorithms	attack type	DOS	Probe	R2L	U2R
C-SVM	SVM's amount	177	178	221	71
	Training Time (s)	0.0150	0.0130	0.0120	0.0078
	Detect Rate (%)	53.3693	83.6957	96.5000	54.8462
	False Rate (%)	5.0000	3.1200	14.7557	0.4563
	Detection	0.0910	0.0557	0.0250	0.0124

	Room (s)				
SVM Co _training	features selected result	{2,3,5,12}	{3,5,6,40}	{1,3,33,36}	{1,3,14,16, 33}
	SVM's amount	1165	107	337	45
	Training time (s)	0.0380	0.0038	0.0150	<0.0001
	Detect Rate (%)	99.8652	89.3478	100.0000	56.1538
	False Rate (%)	2.5316	0.4085	11.0668	0.1704
	Detection Room(s)	0.0940	0.0252	0.0320	<0.0001

By the experimental results shown in Table 1, we can get the following idea:

(1) Compared with C-SVM algorithm, SVM Co-training algorithm proposed takes obvious advantages in the detection rate and false alarm rate. Taking the detection to four kinds of attack types for example, SVM Co-training algorithm's detection rate improves 46.4959%, 5.6521%, 3.5000% and 1.3076% than C-SVM algorithm, and the false alarm rate remains low.

(2) On the detection time, SVM Co-training algorithm takes least time in DoS, Probe and U2R these three types of attacks, reducing 0.097s, 0.0305s, 0.0130s and 0.0123s compared with C-SVM algorithm

Detection for the classified computer can be effectively implemented via Distributed Intrusion Detection Algorithm by improving the SVM Co-training Algorithm, the security of the system being improved.

6. Conclusion

Aiming at the actual situation that very small sample size data can be caught when classified computer network system is under distributed Intrusion Detection. An intrusion detection research was carried out in case of a small amount of data in order to improve detection accuracy, making it easy to timely deal with network intrusion. Intrusion Detection algorithm is proposed based on the improved SVM Co-training, with the algorithm's implementation steps given. Simulation of the actual data shows that the proposed algorithm has a better detection accuracy and stability with respect to the traditional SVM method; and practical application of the algorithm in a classified computer network system does improve the system security.

Acknowledgements

The research work was supported by scientific research fund of guangxi provincial education department (NO.2013lx194).

References

[1] R. N. Smith and S. Bhattacharya, "Firewall Placement in a Large Network Topology", Proceedings of the 6th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDCS '97), (**1997)**.
[2] W. Jansen, P. Mell, T. Karygiannis and D. Marks, "Applying Mobile Agents to Intrusion Detection and Response", NIST Technical Report, (**1999)** September 1-46.
[3] S. M. Bellovin, "Distributed Firewalls", J. Journal of Login, vol. 1, no. 1, (**2001),** pp. 37-39**.**
[4] G. Li, M. Tuo and H. Zeng, "Introduction to support vector machines", electronic industry press, Beijing, (**2004)**.
[5] S. R. Snapp, "DIDS (Distributed Intrusion Detection System) - Motivation, Architecture and an early Prototype", in Proceedings of the 14th National Computer Security Conference, (**1991)** October, pp. 167-176.
[6] L. T. Heberlein, C. V. Gigs, K. N. Levitt, B. Mukherjee, J. Wood and D. Wolber, "A Network Security Monitor", IEEE Computer Society Press, Los Alarnitos, CA, **(1990)** May, pp. 296-304.
[7] N. Tuck and T. Sherwood, "Deterministic Memory-Efficient String Matching Algorithms for Intrusion Detection", IEEE INFOCOM 2004, (**2004)** March 7-11, Hong Kong China.
[8] L. Tan and T. Sherwood, "A High Throughput String Matching Architecture for Intrusion Detection and Prevention", 32nd Annual International Symposium on Computer Architecture, ISCA'2005, (**2005)** June 4-8, Madison, Wisconsin USA.
[9] S. Snapp, J. Brentano, G. Dias, T. Goan, L. Heberlein, C. Ho, K. Levitt, B. Mukherjee, T. Grance, D. Mansur, K. Pon and S. Smaha, "A System for Distributed Intrusion Detection", COMPCON '91, (**1991)** February 25-March 1, San Francisco, CA.
[10] S. Ioannidis, A. D. Keromytis, S. M. Bellovin and J. M. Smith, "Implementing a Distributed Firewall", 7th ACM Conference on Computer and Communication Security, (**2000)** November 1-4, Athens, Greece.
[11] G. Mao and D. Zong, Journal of computer research and development, vol. 46, no. 4, (**2009),** pp. 602-609.
[12] F. Wang, Y. Qian and Z. Wang, Computer engineering, vol. 36, no. 12, pp. 164-166.
[13] C. Hou and L. Jiao, Journal of electronics, vol. 37, no. 10, **(2009),** pp. 2173-2180.
[14] L. Wang, Q. Zhuo and W. Wang, computer engineering, vol. 35, no. 3, **(2009),** pp. 202-204.