A Novel Semi-supervised SVM Based on Tri-training for Intrusition Detection

Jimin Li [1,2]
[1] College of Computer Science and Technology,Tianjin University, Tianjin, 300072 China
[2]College of Mathematics and Computer, Hebei University, Baoing, 071002 China
Email: ljm@hbu.edu.cn

Wei Zhang
College of Electronic and Information Engineering, Hebei University, Baoding, 071002 China
Email: wzhanghbu@gmail.com

KunLun Li
College of Electronic and Information Engineering, Hebei University, Baoding, 071002 China
Email: likunlun@hbu.edu.cn

***Abstract*—One of the main difficulties in machine learning is how to solve large-scale problems effectively, and the labeled data are limited and fairly expensive to obtain. In this paper a new semi-supervised SVM algorithm is proposed. It applies tri-training to improve SVM. The semi-supervised SVM makes use of the large number of unlabeled data to modify the classifiers iteratively. Although tri-training doesn't put any constraints on the classifier, the proposed method uses three different SVMs as the classification algorithm. Experiments on UCI datasets and application to the intrusion anomaly detection show that tri-training can improve the classification accuracy of SVM and its improved algorithms. We also find the accuracy of final classifier will be higher by increasing the difference of classifiers. Theoretical analysis and experiments show that the proposed method has excellent accuracy and classification speed.**

***Index Terms*—semi-supervised learning, co-training, tri-training, support vector machine, intrusion detection**

I. INTRODUCTION

There are two traditional strategies in machine learning (that is supervised learning and unsupervised learning). Recently, as a new machine learning strategy, semi-supervised learning was proposed. It has attracted many scholars' attentions and become one of the machine learning hotspots in recent decade. Semi-supervised learning is different from two traditional machine learning strategies. Traditional supervised learning needs a set of enough and labeled data as training set to train the classifier. Unsupervised learning doesn't need labeled data. In order to train a classifier, it tries to find the implied structure of unlabeled data [1,2].

Since obtaining and storing data is more cheap and easy, there has been massive data in many practical applications. But most of the data is unlabeled. The limited labeled data can't train a supervised classifier with fine generalization performance. Large numbers of unlabeled data could not be applied. Such as spam classification, there are more than 100 million mails in the daily network. Most of the mails will be unlabeled data. They are useless for supervised learning. In this case, traditional supervised learning can't train a classifier with low generalization error. So many people research more and more semi-supervised learning. As the third learning strategy, it exploits unlabeled data in addition to labeled ones. Semi-supervised learning has good prospect in application [2].

In this paper, a semi-supervised SVM based on tri-training is proposed. The experiments on UCI data sets indicate its good performance.

II. SUPPORT VECTOR MACHINE (SVM) AND THREE IMPROVED ALGORITHMS OF SVM

Support vector machine (SVM) is a new general and efficient machine learning algorithm based on Statistical Learning Theory (SLT). The goal of SVM is to separate two classes by a separating hyperplane. We hope that the optimal hyperplane we found has the maximal margin, so as to the good generalization performance [3,4].

Compared with conventional machine learning methods, it has many advantages [3,5]:

(1) Good generalization performance. (2) Global optimal solution. (3) Kernel trick. (4) Good robustness

Because of the above advantages, SVM has been recently used in many applications and has many improved algorithms, such as Least square SVM (LS-

Our research is supported by:

National Natural Science Foundation of China No. 60773062, Program for Science and Technology Development of Hebei Province No. 072135188 and Research Foundation of Education Bureau of Hebei Province No. 2008312.

doi:10.4304/jcp.5.4.638-645

SVM) [6], Proximal SVM (PSVM) [7] and One-Class SVM[8].

As the improved algorithm of classical SVM, LS-SVM was proposed by Suykens and Vandewalle in 1998. Its important improvement is that it solves a set of linear equations, instead of quadratic programming for classical SVM's.

PSVM is another improved algorithm of classical SVM. It changes the inequality constraints into equality constraints in the optimisation problem. And instead of a standard support vector machine that classifies points by assigning them to one of two disjoint half-spaces, PSVM classifies points by assigning them to the closest of two parallel planes. The most advantage of PSVM is its speed.

In 1999, Schölkopf et al. suggested One-Class SVM. The traditional SVM is used to two-class problem. It needs two-class examples, i.e. negative and positive examples. One-Class SVM adapts the SVM to the one-class classification problem. It uses a kernel function to map the data into a feature space. One-Class SVM treats the origin in feature space as the only example of the second class. And it tries to separate the most of examples from the origin with maximum margin. It believes the most of examples are the normal data and the rest are the outlier [8,9,10].

Ⅲ. SEMI-SUPERVISED LEARNING AND CO-TRAINING

A Semi-supervised learning

Now there are some learning strategies in machine learning. Semi-supervised learning is one of them. It makes use of massive unlabeled data and reduces the difficulty and cost of obtaining the labeled data.

In 1960s, researchers had proposed semi-supervised learning idea in classification. It is self-training, which is also known as self-learning. With the in-depth study, researchers have a lot of research results and the corresponding algorithm on semi-supervised learning. Some often-used methods include: Generative mixture models, Self-training, Co-training and Graph-based methods [1,2]. And semi-supervised learning is also applied to regression, clustering and so on.

B Co-training and Tri-training

Co-training is a kind of semi-supervised learning paradigm that was proposed by Blum and Mitchell in 1998 [11]. It assumes that attributes can be split into two sufficient and redundant views. Each view is sufficient to train a good classifier. Initially two separate classifiers are trained with the labeled data, on the two views respectively. Each classifier then classifies the unlabeled data, and teaches the other classifier with some unlabeled examples (and their predicted labels) if the predicted labels are most confident. Each classifier will be retrained with the additional training examples given by the other classifier, and the process is repeated for higher accuracy [2, 11, 12].

In 2000, Goldman and Zhou proposed an improved co-training algorithm [12]. It employs time-consuming cross validation technique to determine how to label the unlabeled examples and how to produce the final hypothesis.

Co-training's main disadvantages are the time-consuming cross validation and the strict condition for the classification algorithm and data.

Zhou and Li proposed tri-training algorithm for solving the problem of co-training in 2005[13]. It doesn't require the instance space be described with sufficient and redundant views. And it uses three learners. This approach thus avoids explicitly measuring label confidence of each learner. So it is fast and easy to extend to the common data.

Tri-training is described in detail as follows:

Let L denote the labeled example set, h_1, h_2 and h_3 denote initial learners and U denote the unlabeled example set. x is an example in U. Three classifiers are initially trained from labeled examples. Any two of three classifiers are used to label the unlabeled examples x, if two of them agree on the label; the example will be used to teach the third classifier. It repeats this work until none of $h_i\,(i=1,2,3)$ changes. The final prediction is made with a variant of a majority vote among all the learners.

Ⅳ. SEMI-SUPERVISED SVM

Though co-training is an effective algorithm, most of data can't be described with sufficient and redundant views in practical applications. And the improved co-training algorithm, which is proposed by Goldman and Zhou, consumes much time when it improves the classifiers. These problems lead co-training to use in practical application hard. So we select its improved algorithm-tri-training to improve SVM. Tri-training has more learners than standard co-training algorithm. But according to ensemble theory [14], the more learners and the better effect. And tri-Training thus avoids time-consuming cross validation. So it is faster than co-training.

A Learning from noisy examples

According to the research of Angluin and Laird [15], we let $\sigma=\{(x_1,y_1),\cdots,(x_m,y_m)\}$ denote a sequence of m samples, which is drawn. x_i is one example and y_i is the label of x_i. L_i is any possible hypothesis; $F(L_i,\sigma)$ denote the number for which L_i disagrees with σ.

Theorem:

If we draw a sequence σ of

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

samples and find any hypothesis L_i that minimizes $F(L_i,\sigma)$, then

$$\Pr[d(L_i,L_*)\geq\varepsilon]\leq\delta \tag{2}$$

this means the hypothesis L_i will have the PAC property.

Where ε is error tolerance, it is the hypothesis worst-case classification error rate, δ is the confidence, η is an upper bound on the classification noise rate of training set, $0 \le \eta \le 0.5$, N is the number of hypothesis, $d(L_i, L_*)$ is the sum on probability of symmetric difference between the hypothesis L_i and the real hypothesis L_*.

When μ make Eq. (1) hold equality, $C = 2\mu \ln(\frac{2N}{\delta})$ and $m = \frac{C}{\varepsilon^2(1-2\eta)^2}$, let

$$u = \frac{C}{\varepsilon^2} = m(1-2\eta)^2 \tag{3}$$

B Semi-supervised SVM based on Tri-training

Tri-training needs three learners. It has no special demand for these learners. According to ensemble theory, we know that the more difference between learners, the higher accuracy. For increasing the independence of learners, we complete the experiment in two ways. First, we select three different SVM classifiers: classical SVM, LS-SVM and PSVM. The different SVM classifiers can avoid the Semi-supervised SVM become the ensemble of three self-training classifiers. Second we also select three SVM or LS-SVM classifiers. But the difference between the three classifiers is the kernel function. In practice, most of users know less about the types of data. So they don't know how to select the kernel function. Besides increasing the independence of learners to avoid the Semi-supervised SVM classifier with three different kernel functions becomes the ensemble of three self-training classifiers, it can suit more types of data, Although the three SVM classifiers are different, they have the same output form. It is easy to ensemble.

Let L denote the initial labeled example set and size is $|L|$, U denote the unlabeled example set and size is $|U|$. First, the three classifiers are trained by data set that is bootstrap sampling from L. After the initial training, one of the three classifiers will be as the Training target and the others are the auxiliary classifiers. The auxiliary classifiers are used to classify the examples in U. If they have an agreement about the label of an unlabeled example, the example with the label will be gathered together as L' and the training target classifier is retrained by $|L \bigcup L'|$. It should be noted that in the next round L' is not as the labeled data set and will be reused as unlabeled data. If the label of one example in L' is a correct prediction, it means the training set will have an additional correct example for the training target classifier. Otherwise it means that the classifier will get a noisy example. The noise will decrease the classifier's accuracy. How to avoid the influence of noise? According to the description 4.1, let L'' denote the training set for the target classifier in new round of tri-training. Let η_L denote the classification noise rate of L, $e^{t'}$ and η' denote the upper bound of classification error rate and the classification noise rate of the target classifier in previous round. Then

$$\eta' = \frac{\eta_L |L| + e^{t'} |L'|}{|L \bigcup L'|} \tag{4}$$

due to Eq. (3), if $u'' > u'$, namely

$$|L \bigcup L''|(1 - 2\frac{\eta_L |L| + e^{t''} |L''|}{|L \bigcup L''|}) > |L \bigcup L'|(1 - 2\frac{\eta_L |L| + e^{t'} |L'|}{|L \bigcup L'|}) \tag{5}$$

then $\varepsilon'' < \varepsilon'$. With joining the unlabeled data, the performance of classifier will be improved. So we can remove the influence of noise by increasing the examples. This is the reason why we let L' be as the unlabeled data in the next round [13, 15].

In this paper, we select classical SVM, LS-SVM and PSVM as the three different SVM classifiers. And for observing effect of classifier independence, we choose two ways to do the experiments. One is Semi-supervised strategy apply on the three different SVM classifiers. The other is Semi-supervised strategy apply on the three same PSVM classifiers.

Ⅴ. EXPERIMENTS

A Experiments Result

In experiments, we select 5 UCI data sets [16]: Australian, German, Ionosphere, Pima, Wdbc to prove the validity of semi-supervised SVM algorithm. German, Ionosphere, Pima, Wdbc is used in Experiment Ⅰ. Australian, German, Ionosphere, Wdbc is used in Experiment Ⅱ.

All our experiments were performed on a computer, which utilizes a 2GHz Pentium E2180 CPU and a 2 Gigabytes memory. The computer runs on Windows XP, with Matlab 7.1 installed.

Experiment Ⅰ

OSU SVM3.0, LS-SVM and PSVM Matlab toolbox are used in the experiments Ⅰ. And the selection of SVM kernel function is linear function: $K(x, x_i) = (x \cdot x_i)$. In classical SVM the penalty parameter c is set to 10. And the regularization parameter gam is set to 1 in LS-SVM. The same tri-training strategy will be applied in the experiments.

We keep 25% data as the test data for every dataset. And the rest are divided into two parts under different unlabel rates, one is L, the other is U. The unlabel rate is respectively 70%, 50%, 30%. Every parts of the dataset have the similar pos/neg ratio with the whole dataset. We complete the experiments on different unlabeled rate. Under each unlabeled rate, three independent classifiers

runs will be performed. The three results are averaged and summarized in Table Ⅰ to Ⅲ.

TABLE Ⅰ.
CLASSIFICATION ERROR RATES AND ALGORITHM RUNNING TIME, UNDER 70% UNLABELED RATE

Data set		German	Ionosphere	Pima	Wdbc
SSDC-SVM	Initial (%)	27.1	16.3	24.5	9.6
	Final (%)	25.7	15.6	23.2	7.1
	Time (s)	78.6	15.7	76.4	67.7
SSP-SVM	Final (%)	26.1	16.7	25.9	7.7
	Time (s)	1.2	0.4	0.6	0.5

TABLE Ⅱ.
CLASSIFICATION ERROR RATES AND ALGORITHM RUNNING TIME, UNDER 50% UNLABELED RATE

Data set		German	Ionosphere	Pima	Wdbc
SSDC-SVM	Initial (%)	24.8	14.8	22.4	6.0
	Final (%)	23.3	13.6	21.4	4.6
	Time (s)	155.8	16.9	318.8	77.4
SSP-SVM	Final (%)	24.6	14.7	21.9	5.6
	Time (s)	1.8	0.7	1.0	1.2

TABLE Ⅲ.
CLASSIFICATION ERROR RATES AND ALGORITHM RUNNING TIME, UNDER 30% UNLABELED RATE

Data set		German	Ionosphere	Pima	Wdbc
SSDC-SVM	Initial (%)	22.5	12.5	20.8	4.2
	Final (%)	20.8	11.4	20.3	3.5
	Time (s)	189.6	18.3	388.9	97.1
SSP-SVM	Final (%)	22	14.8	21.3	4.9
	Time (s)	2.9	1.4	1.1	1.7

In the table, SSDC-SVM denotes the semi-supervised SVM algorithm that uses three different SVM classifiers，SSP-SVM denotes the semi-supervised SVM algorithm with three same PSVM classifiers. All results have been rounded to one decimal place.

For the SSDC-SVM, we select the initial error rate, final error rate and running time as the evaluation of the algorithm performance (the initial error rate is the ensemble result comprising three classifiers which are trained by Bootstrap sampling from initial labeled set L). For SSP-SVM algorithm, the final error rate and running time are selected as the experiment result.

Experiment Ⅱ

OSU SVM3.0 and LS-SVM Matlab toolbox are used in the experiments. And the selections of SVM kernel function are: linear function $K(x, x_i) = (x \cdot x_i)$, polynomial function $K(x, x_i) = (x \cdot x_i + c)^d$, and RBF function $K(x, x_i) = \exp(-\| x - x_i \|^2 / \sigma^2)$. The same tri-training strategy will be applied on the classic SVM and LS-SVM.

We also complete the experiment as Experiment Ⅰ. But the unlabeled rates become 80%, 60%, 40% and 20%. The averaged results are summarized in Table Ⅳ to Ⅷ. In the table, SSDK-SVM1 denotes Semi-supervised LS-SVM algorithm with three different kernel functions.

TABLE Ⅳ.
CLASSIFICATION ERROR RATES, UNDER 80% UNLABELED RATE

Data set	SSDK-SVM1		SSL-SVM	SSDK-SVM2
	Initial(%)	Final(%)	Final(%)	Final(%)
Australian	15.12	14.92	13.95	-----
German	28.67	27.2	26.8	26.8
Ionosphere	21.96	18.94	14.394	10.984
Wdbc	15.96	11.74	7.40	9.62

TABLE Ⅴ.
CLASSIFICATION ERROR RATES, UNDER 60% UNLABELED RATE

Data set	SSDK-SVM1		SSL-SVM	SSDK-SVM2
	Initial(%)	Final(%)	Final(%)	Final(%)
Australian	14.54	13.76	13.37	-----
German	28.13	26.27	25.6	24.8
Ionosphere	21.21	16.67	12.5	8.33
Wdbc	11.03	8.69	5.4	9.39

TABLE Ⅵ.
CLASSIFICATION ERROR RATES, UNDER 40% UNLABELED RATE

Data set	SSDK-SVM1		SSL-SVM	SSDK-SVM2
	Initial(%)	Final(%)	Final(%)	Final(%)
Australian	13.18	12.98	12.79	-----
German	26.13	25.73	25.2	25.2
Ionosphere	17.80	10.23	10.61	6.82
Wdbc	9.39	6.81	5.16	7.04

TABLE Ⅶ.
CLASSIFICATION ERROR RATES, UNDER 20% UNLABELED RATE

Data set	SSDK-SVM1		SSL-SVM	SSDK-SVM2
	Initial(%)	Final(%)	Final(%)	Final(%)
Australian	12.43	12.40	12.21	-----
German	26.93	25.07	24.4	-----
Ionosphere	15.15	10.23	9.50	5.30
Wdbc	7.04	6.10	4.93	-----

SSDK-SVM2 denotes the Semi-supervised SVM algorithm with three different kernel functions. And SSL-SVM is the Semi-supervised SVM algorithm which three SVM kernel functions are all linear functions.

For SSDK-SVM1, we select the initial error rate, final error rate and running time as the evaluation of the algorithm performance (the definition of initial error rate is the same as it in Experiment Ⅰ). For SSL-SVM and SSDK-SVM2, the final error rate and running time are selected as the experiment result. If the program runs

more than 14 hours, we will give no result on this data set.

TABLE Ⅷ.
ALGORITHM RUNNING TIME

Unlabeled rate	Australian			German			Wdbc		
	SSDK-SVM1	SSL-SVM	SSDK-SVM2	SSDK-SVM1	SSL-SVM	SSDK-SVM2	SSDK-SVM1	SSL-SVM	SSDK-SVM2
80%	62.68	496.83	---	128.64	46.44	46.88	95.92	5.48	53.05
60%	65.49	10040.10	---	142.7	174.88	1187.34	97.64	111.82	178.88
40%	68.35	18476.63	---	145.93	293.33	10463.89	98.71	232.62	596.13
20%	72.71	24933.93	---	161.15	463.34	---	105.04	329.44	---

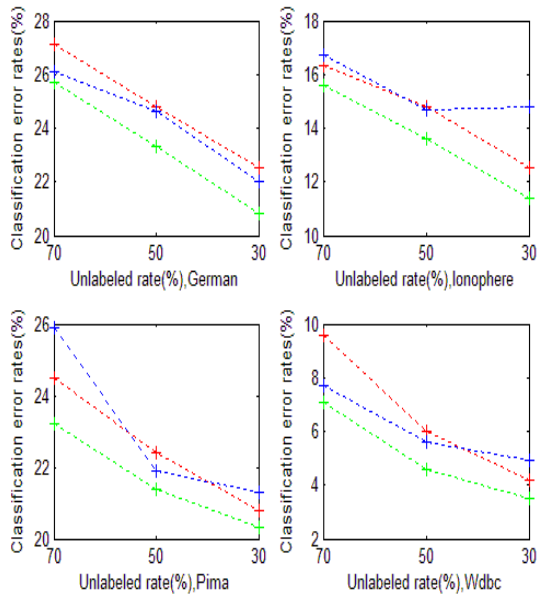

Figure 1. Experiment Ⅰ results on 4 data sets

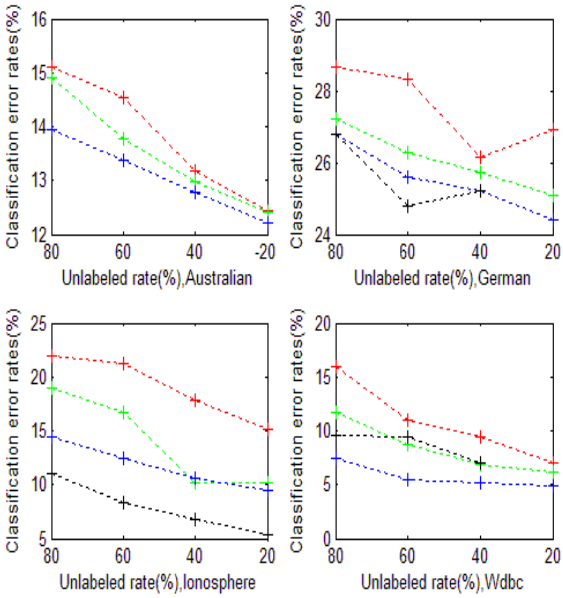

Figure 2. Experiment Ⅱ results on 4 data sets

B Experiments analysis

Experiment Ⅰ

Table Ⅰ to Ⅲ reveal that the semi-supervised learning strategy can improve the accuracy remarkably. The error rates of Experiment Ⅰ compared algorithms are depicted in Fig 1. For the two algorithms, they have same training data and testing data. The red and green curves indicate the initial and final classification error rate of SSDC-SVM respectively. The blue curve indicates the final classification error rate of SSP-SVM. Fig 1 obviously shows that the final accuracy generated by Semi-supervised SVM is better than the initial accuracy. And with decrease of the unlabeled rate, the error rates are also decreasing. In classification accuracy, SSDC-SVM is better than SSP-SVM. It shows that the different classifier can increase the independence and improve the last classification accuracy. And PSVM only focuses attention on the SVs. Although the optimal separating hyperplane is decided by the SVs, the other points maybe influence the place of the hyperplane. So the SSP-SVM accuracy is lower than SSDC-SVM. But the gap between the two algorithms is little. Wdbc is a linearly separable data set, so the accuracy on this dataset is better than the accuracy on the others. This shows that the selection of kernel function in SVM may also affect the final accuracy a lot for some data.

Table Ⅰ to Ⅲ reveal the algorithm's running time. For SSP-SVM or SSDC-SVM, with the unlabeled rate decreasing, the initial training set is increasing. The running time of all two semi-supervised SVM algorithm become longer. And SSP-SVM is faster than SSDC-SVM. When the data set is large, the speed advantage of SSP-SVM is more obvious. This is because PSVM is fast than any other two SVM classifiers.

Experiment Ⅱ

The error rates of Experiment Ⅱ compared algorithms are depicted in Fig 2. For the two algorithms, they have same training data and testing data. The red and green curves indicate the initial and final classification error rate of SSDK-SVM1 respectively. The blue and black curves indicate the final classification error rate of SSL-SVM and SSDK-SVM2 respectively.

Ⅵ. Application to Intrusion Anomaly Detection

We will use the intrusion detection data to test the performance of the Semi-supervised SVM. The data is derived from the 1999 Knowledge Discovery and Data mining Cup publicly available dataset (KDD'99)[16]. We choose this data set for two reasons. First, it has been used popularly as a standard for comparing the performance of intrusion-detection systems. Second, since the data is labeled, we can verify the accuracy of our detection scheme. The labeled 10% training data (kddcup.data_10_percent) is used. It has about 500,000 data and consists of 22 attack types, which can be arranged into 4 namely Probe, DOS, U2R and R2L. Each date comprises of a comma delimited set of 41 features and a label that indicates whether the record is normal or attack. We want to do the intrusion anomaly detection. So all attack types will be treat as outlier. It means there are only two categories-- normal and anomaly. In the real network environment, a very small number of network examples are labeled. And labeled ones are fairly difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile unlabeled data may be relatively easy to collect examples by many network capture tools, such as winpcap, libpcap and so on. We want to exploit our Semi-supervised SVM to use the mass unlabeled records and improve the anomaly detection performance [17,18,19,20].

TABLE Ⅸ.
THE PERFORMANCE OF SEMI-SUPERVISED SVM ON INTRUSION DETECTION

Data1							
Method	Initial			Final			Detection time (s)
	Precision (%)	False positive (%)	Detection rate (%)	Precision (%)	False positive (%)	Detection rate (%)	
SSR-SVM (C=50,sig2=0.003)	98.567	0	91.383	98.9	0.04	93.587	17.156
SSDK-SVM1 (d=3,gam=0.08,sig2=0.5)	98.5	0	90.982	99.067	0.12	94.99	353.484
SSDC-SVM (C=100,gam=10,sig2=0.001)	98.9667	0	93.788	99	0.04	94.188	158.906
SSOC-SVM ($sig2=2*10^{-6}$,nu=0.0969)	87.267	14.195	98.397	90.467	11.116	98.397	2.281
Data2							
Method	Initial			Final			Detection time (s)
	Precision (%)	False positive (%)	Detection rate (%)	Precision (%)	False positive (%)	Detection rate (%)	
SSR-SVM (C=50,sig2=0.01)	99.23	0	95.38	99.257	0	95.54	1834.125
SSOC-SVM ($sig2=1*10^{-6}$,nu=0.07)	91.907	9.184	97.36	92.347	8.652	97.34	98.141

To simulate the real network environment (few labeled training data, much unlabeled training data and mass testing data), We randomly extract instances for labeled training set, unlabeled training set and testing set from kddcup.data_10_percent respectively. And the labeled training set consists of 4000 examples, which has 3000 normal examples (75%) and 1000 attack ones (25%). The unlabeled training set comprises of 10000 examples—7000 normal examples (70%) and 3000 attack ones (30%). The testing set consists of 30000 examples—25000 normal examples and 5000 attack ones. The three data sets are together called Data2 in tableⅨ. Data1 is the subsample of Data2. It is randomly extracted from Data2 and the labeled, unlabeled training set and testing set are all 10% of Data2's three data sets. They have same normal/anomaly ratio. There are some non-numeric features in KDD'99 data set. So we have changed them into number in data preprocessing.

On Data1, we test 4 kinds of Semi-supervised SVM methods, which are proposed in this paper. There are the optimal results for every method by adjusting parameters. In the table Ⅸ, SSR-SVM denotes the Semi-supervised SVM algorithm which three SVM kernel functions are all RBF functions. SSDC-SVM denotes the Semi-supervised SVM algorithm with three different SVM classifiers, and the kernel functions of OSU SVM and LS-SVM are RBF kernels, the kernel function of PSVM is linear function. SSOC-SVM is the Semi-supervised One-Class SVM with RBF kernel function. The precision (Eq 6), false positive (Eq 7), detection rate (Eq 8) and detection time are shown in table Ⅸ, where initial denotes the performance achieved using only the labeled training data, and final denotes the performance obtained after adding unlabeled examples.

$$Precision = \frac{Number\ of\ correctly\ classified\ examples}{Number\ of\ total\ examples} * 100\% \quad (6)$$

$$false\ positive = \frac{Number\ of\ misclassified\ normal}{Number\ of\ normal} * 100\% \quad (7)$$

$$Detection\ rate = \frac{Number\ of\ detected\ attacks}{Number\ of\ attacks} * 100\% \quad (8)$$

Table Ⅸ shows that Semi-supervised learning can effectively use unlabeled examples to improve the performance of SVM classifier on the intrusion anomaly detection. All of the precision and the most of the detection rate are enhanced obviously on data1. For the SSOC-SVM, it has 3.7% improvements in precision. For the SSDK-SVM1, 4.4% improvement has achieved in detection rate. The data is randomly selected. Some attacks in the testing set may not exist in the training set. So the false positive increase lightly. Only SSOC-SVM has 21.7% improvement in false position.

Actually, through observing Table Ⅸ it can be found that when Semi-supervised learning is used, the improvement brought by different kernel function is bigger than that brought by different classifiers. And the improvement of SSDK-SVM1 and SSDC-SVM are bigger than the improvement of SSR-SVM. This indicates that increasing the independence of learners can improve classification performance more effectively.

For SSOC-SVM, its false positive is higher than any other Semi-supervised method, but its detection rate is much better than their detection rate. That is because One-Class SVM is good at anomaly detection. It can find more attack. So it suit for anomaly detection, but not misuse detection.

Table Ⅸ also shows that the detection time of SSOC-SVM is faster than any other three methods. It's the advantage of One-Class SVM [10]. And this advantage is more obvious on Data2. Comparing the result on Data1 with the result on Data2, with the enlargement of data the precision, false positive, detection rate of SSR-SVM are significantly improved. Despite the detection rate of One-Class SVM is slight decline on Data2, the improvement of the other two indicators are also significant. There is a distinct speed advantage in One-Class SVM on large datasets.

Ⅶ. Conclusion

In this paper, a new semi-supervised SVM based on tri-training is proposed. Theoretical analysis and experiments show that the proposed has excellent accuracy. It can improve the accuracy obviously. This suggests semi-supervised SVM has research and practical value.

Through the experiment, it is known that SSDC-SVM and SSDK-SVM have accuracy advantage and SSP-SVM has speed advantage. So the first two methods based on tri-training is suitable for the no real-time applications, such as Text Categorization, Speech Recognition and so on. It can receive better accuracy. Although the accuracy of SSP-SVM is lower than SSDC-SVM's，the running time is its advantage and the performance is particularly evident in large-scale data. So it's suitable for the real-time applications, such as real-time misuse intrusion detection. A high detection rate and the speed are the advantage of SSOC-SCM, so it's suitable for the real-time anomaly intrusion detection.

By observing and analyzing the experiments result, semi-supervised learning strategy can improve the classification accuracy of SVM algorithm. But there is much repetitive work on training classifier in tri-training and most of data are used repeatedly. This wastes a lot of time. So how to avoid the iterative work on the used data is problem that needs to be desiderated. Incremental learning can be used as one of solution in our future work for this semi-supervised SVM. And how to select the kernel function or create a new kernel, which suit the most data, is the other question we need to solve.

References

[1] O. Chapelle, B. Schölkopf, and A. Zien, "Semi-Supervised Learning," *The MIT Press*, 2006.

[2] X. J. Zhu, "Semi-supervised learning literature survey," *Technical Report*, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, December, 2007.

[3] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods," *Cambridge University Press*, 2000.

[4] R. G. Steve, "Support vector machines classification and regression," *Technical Report*, Department of Electronics and Computer Science, University of Southampton, 1998.

[5] V. Vapnik, "The Nature of Statistical Learning Theory ," New York: Springer-Verlag, 2000.

[6] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, pp. 293-300, 1999.

[7] G. Fung and O. L Mangasarian, "Proximal Support Vector Machine Classifiers," *Knowledge Discovery and Data Mining*, pp. 26-29, August 2001, San Francisco, CA, New York, Association for Computing Machinery, pp.77-86, 2001.

[8] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Technical report*, Microsoft Research, MSR-TR-99-87, 1999.

[9] M. Larry Manevitz and Y. Malik, "One-class SVMs for document classification," *Journal of Machine Learning Research*, ,vol. 2, pp. 139-154, 2001.

[10] K. L. Li, H. K. Huang and S. F.Tian, "Improving one-class SVM for anomaly detection," *International Conference on Machine Learning and Cybernetics*, Vol. 5, pp. 3077-3081, 2003.

[11] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," *In: Proceedings of the 11th Annual Conference on Computational Learning Theory*, Madison, WI, pp.92–100, 1998.

[12] S. Goldman and Y. Zhou, "Enhancing supervised learning with unlabeled data ," *In: Proceedings of the 17th ICML*, San Francisco, CA, Morgan Kaufmann, pp. 327-334, 2000.

[13] Z. H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 1529-1541, 2005.

[14] O. Richard Duda and E. Peter, "Pattern Classification (2nd Edition)" , New York: Wiley, 2001.

[15] D. Angluin and P. Laird, "Learning from noisy examples", *Machine Learning*, vol.2, no.4, pp. 343–370, 1988.

[16] UCI repository of machine learning databases, http://www.ics.uci.edu/mlearn/MLRepository.html
[17] K. L. Li, "Unsupervised SVM Based on p-kernels for Anomaly Detection", *International Conference on Innovative Computing, Information and Contro*l, Aug. 30-Sept.1, 2006, Beijing, China.
[18] S. Mukkamala and H. Sung Andrew, "A comparative study of techniques for intrusion detection," *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, Sacramento, CA, pp. 570-577, 2003.
[19] Z. H. Zhang and H. Shen, "Online Training of SVMs for Real-time Intrusion Detection[," *Proc. of the 18th International Conference on Advanced Information Networking and Application*, Fukuoka, Japan, 2004.
[20] W. H. Chen, H. H. Sheng and H. P. Shen, "Application of SVM and ANN for intrusion detection," *Computers & Operations Research*, vol. 32, pp. 2617-2634, 2005.
[21] K. L. Li and H. K. Huang, "Fuzzy multi-class support vector machine and application in intrusion detection," *Chinese Journal of Computers*, vol. 28, pp. 274-280,2005.
[22] K. L. Li and H. K. Huang, and S. F.Tian, "A novel multi-class SVM classifier based on DDAG", *IEEE 2002 International Conference on Machine Learning and Cybernetics*, Beijing, China, vol. 3, pp. 1203-1207, 2002.

Jimin Li, born in 1969, received M.S. degree in Computer Application from Hebei University of China in 2001. He is studying for PhD degree in College of Computer Science and Technology of Tianjin University, China. He is an Associate Professor at College of Mathematics and Computer of Hebei University. His main research interests include network security, machine learning and data mining. In these areas, he has published over 10 technical papers in refereed international journals or conference proceedings.

Wei Zhang, born in 1982, received the MSc degree in Electronic and Information Engineering from Hebei University, China, in 2005. Currently he is a BSc candidate at the Electronic and Information Engineering of Hebei University. His research interests are in machine learning and data mining and intelligent network security especially in learning with unlabeled examples.

Kun-Lun Li, born in 1962, received the PhD degrees in Signal & Information Processing from Beijing Jiaotong University, China, in 2004 and join College of Electronic and Information Engineering of Hebei University as the associate professor at present. His main research interests include machine learning, data mining, intelligent network security and biology information technology. In these areas, he has published over 20 technical papers in refereed international journals or conference proceedings.