Network Intrusion Detection using Semi Supervised Support Vector Machine

Jyoti Haweliya
Department of Computer Engineering
Institute of Engineering & Technology, Devi Ahilya University
Indore, India

Bhawna Nigam
Department of Information Technology
Institute of Engineering & Technology, Devi Ahilya University
Indore, India

ABSTRACT
The use of Internet is growing bit by bit and therefore huge amount of security threats faced in front of computer network system. Due to these threats secrecy of the information which is available in the network system is highly affected. To protect our network system from these threats, it becomes very important to build up a system that acts as a barrier between the network systems and the unessential security attacks. For the monitoring and detecting the intrusion (unwanted access), an Intrusion Detection Systems (IDS) were developed. But the expected performance and accuracy are not achieved by these systems. In this paper we propose a Semi Supervised Support Vector Machine (S3VM) to overcome these two concerns. The semi supervised SVM also overcomes the shortcoming of supervised SVM that require only labeled data for training the classifier. Semi Supervised Support Vector Machine is based on Self Training algorithm for semi supervised learning. The dataset used for training and testing purpose is NSL-KDD dataset. This model provides classification accuracy up to 90%.

Keywords
Semi Supervised Learning, Support Vector Machine, Intrusion Detection, Self Training, kernel functions

1. INTRODUCTION
The network based computer systems plays a crucial role in the growth of any business and country. Some unwanted attacks are often made on these systems, to access the important data. An intrusion can be defined [1] as "a group of events that are harmful for the secure working of the system". Many different systems for intrusion detection are developed such as firewall, intrusion detection systems (IDS), IT policy, encryption etc. But, none of them fulfill the required performance expectation. This is due to the [2] lack of accuracy, lack of flexibility, large resource requirements and limitation of use of the intrusion detection systems (IDS).

There are mainly two policies for intrusion detection over the computer network – misuse detection and anomaly detection [3]. In misuse detection, network intrusion is detected by matching the system behaviour with the known pattern. On the other hand, in anomaly detection network intrusion is detected by analysing the deviation of behavior of the system from the normal behavior. Despite of many systems available for intrusion detection, they cannot be used much productively. To overcome this problem, a new intrusion detection system is proposed based on semi supervised SVM [4].

The problem with traditional machine learning method that is supervised learning is that it requires large amount of labeled data for training the classifier. Since labeled data is difficult, expensive and time consuming to obtain. So a new machine learning method named semi supervised learning [5][6] is widely used nowadays. Semi supervised learning uses small amount of labeled data together with large number of unlabeled data to train the classifier. Thus the semi supervised support vector machine provides good generalization performance using small amount of labeled data unlike supervised support vector machine which uses huge quantity of labeled data to train. The self training algorithm [7] is used for semi supervised learning, which is an iterative algorithm.

This paper proposes an intrusion detection system using semi supervised support vector machine [15] using different kernel functions. The semi supervised SVM is trained with the combination of the labeled and unlabeled data. The dataset used is KDD-Cup99 [8] dataset. Then it is used to detect the presence of intrusion in the packet by classifying the packets based on the hyperplane. The semi supervised SVM provides good generalization performance like supervised SVM but unlike supervised SVM it uses small amount of labeled data.

2. ARCHITECTURE OF INTRUSION DETECTION SYSTEM
This intrusion detection system can be used to detect any kind of network intrusion. We have applied KDD-CUP99 dataset to examine our system. The architecture and parts of the semi supervised SVM intrusion detection system is shown in Fig. 1. The dataset used is first pre-processed using Z-score and Principal component analysis (PCA) [9][10]. The processed dataset is split into two parts training data and testing data. Training data is used to train the model and testing data is used to determine accuracy of the model.

2.1 Pre Processing
The procedure of pre-processing a dataset consists of normalization and dimension reduction of dataset.

2.1.1 Normalization
For efficient use of dataset normalization has to be performed before it can be used. Normalization is applied to make dataset more structured. There are many methods available for normalization are used some of them are z-score, min-max normalization and decimal scaling. We have used z-score normalization on the dataset, it is given as:

z'= z-mean/standard_deviation

2.1.2 Dimension reduction using Principal Component Analysis (PCA)
It is difficult to handle data with large attributes. Thus, it becomes necessary to reduce the dimensionality of data to use it in a practical application. Principal component analysis [12] follows a mathematical procedure that transforms a cluster of possibly correlated variables into a cluster of uncorrelated variables, which is called principal components. It reduces the dimension of data without affecting the results. In PCA following calculations are to be done

The data samples can be represented as

{xi}Ni=1 = {x1. . . xn} where n=41 (1)

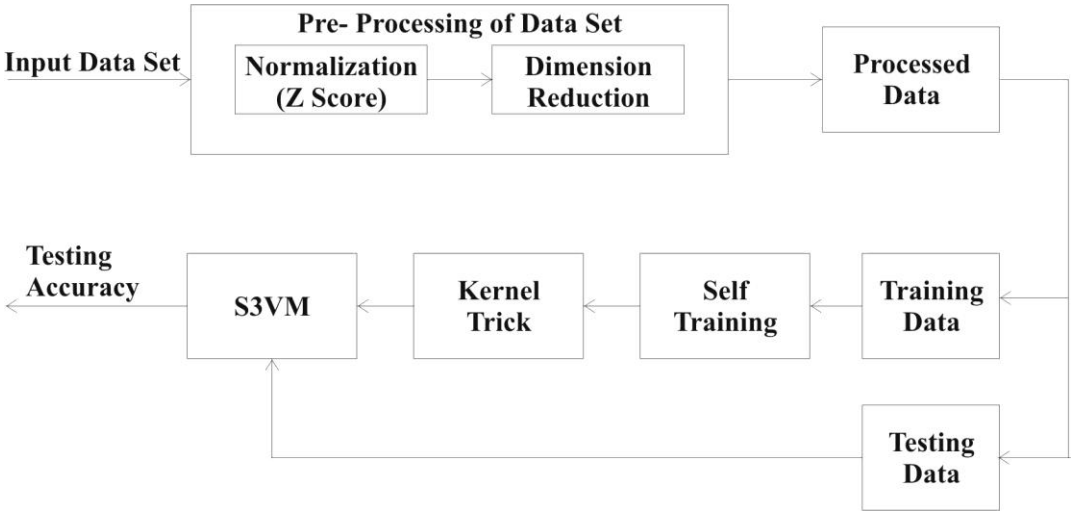

Fig 1: Architecture of semi supervised SVM for Intrusion Detection

Suppose xm = ΦT x

$$\mu = 1/N \sum_{i=1}^{N} xi \quad (2)$$

The covariance matrix is given by

$$C = 1/N \sum_{i=1}^{N} (xi - \mu)(xi - \mu)T \quad (3)$$

By solving the eigen value problem of C we can get the principal components

$$Czk = \lambda kzk \quad (4)$$

Where λk (k=1,2,3…n) represents eigenvalues and zk represents eigenvectors.

For reducing the dimensions of data we have to calculate the m eigenvectors which corresponds to m eigenvalues which are largest. Suppose we have

$$\Phi = [z1, z2\ldots.zm] \text{ and} \quad (5)$$

$$\Delta = diag[\lambda 1,\lambda 2\ldots\lambda m] \quad (6)$$

Then

$$C\Phi = \Phi\Delta \quad (7)$$

A parameter v can be also used for approximation of m largest eigenvectors such that

$$\sum_{i-1}^{m} \lambda i \,/\, \sum_{i-1}^{n} \lambda i >= z \quad (8)$$

We can reduce the dimensions of data x as

$$Xm = \Phi T\, x\, . \quad (9)$$

2.2 Semi Supervised Learning

Semi supervised learning technique is developed to overcome the disadvantage of traditional supervised learning. It is a special form of learning that unlike the traditional learning strategy uses large amount of unlabeled data and small amount of labeled data for training the classifier. Semi supervised learning becomes popular day by day as labeled data is difficult to obtain, while the unlabeled is easily available and it is also inexpensive. Semi supervised learning requires less human effort and also provide better accuracy.

2.2.1 Semi Supervised Learning Algorithm - Self Training

There are many algorithms for semi supervised learning have been developed. The commonly used algorithms [6] are Self Training, Co-Training, Tri-Training, Generative Mixture Model, Transductive Support Vector Machine, and Graph Based Method. We have applied Self Training which is simple and effective semi supervised learning algorithm.

In Self Training algorithm [7], the classifier is first trained with the small amount of labeled data. Then it is used to classify the unlabeled data and these newly obtained labeled data is added to the previous labeled data. Now the classifier is trained with the increased labeled data. This procedure is repeated until the classifier is sufficiently trained. This algorithm is also called Self-Teaching.

The algorithm works as follows :

Let we are having a training dataset dt containing n samples { Xi, i=1...n } with given label { Yi, i=1...n } and a test dataset ds containing m samples { xj, j=1...m } but its labels are known.

1. In the first step, Train the classifier with the training dataset dt and then classify the test samples.

2. Perform step 2.1 – 2.2 k times (k=2...)

2.1 Create a new training set Dnew = dt + ds. ds is the new labeled data obtained in the first step.

2.2 Train the classifier with the dataset Dnew and again classify the unlabeled data in ds.

3. After kth iteration the classifier is sufficiently trained for performing the classification. We have done 11 iterations for training the classifier.

2.3 Kernel Trick

All the parameters that are needed in the learning task of the semi supervised support vector machine require lengthy and complex procedure for calculating their values. It requires in transformed space $\phi(xi).\phi(xj)$, calculations of dot product between pairs of vectors. These calculations are very time consuming and complex and may suffer from the dimensionality problem. To reduce these calculation kernel functions are used. These are also called kernel trick.

The kernel functions are a kind of mathematical trick by which we can classify the data in three dimensional spaces, while the data was originally defined in two dimensional spaces. The kernel functions originally maps the data from the lower dimensional space in which the data is not linearly separable to the higher dimensional space where the data is linearly separable.

A kernel function K can be represented as

$K(u,v,) = \Phi(u) . \Phi(v)$

Where u and v are n-dimensional vectors.

The following are the kernel function used:

Log Kernel Function –

$$k(x,y) = -log(||x-y||^d + 1)$$

Multiquadric Kernel Function –

$$k(x,y) = \sqrt{||x-y||^2 + c^2}$$

Radial Basis Function (RBF) –

$$k(x,y) = \exp\left(-\gamma||x-y||^2\right)$$

Inverse Multiquadric Kernel Function –

$$k(x,y) = \frac{1}{\sqrt{||x-y||^2 + c^2}}$$

Power Kernel Function – $k(x,y) = -||x-y||^d$

2.4 Semi Supervised Support Vector Machine

Semi supervised support vector machine is one of the newest approach to the problem of classification. The semi supervised support vector machine is a classifier that is used to separate the data into different classes based upon their attributes .The semi supervised support vector machine when trained with the training data containing some labeled data together with unlabeled data, it finds the maximum margin hyperplane that separates the two classes in the original space or higher dimensional space through the use of kernel function. Classification of data in the higher dimensional space using kernel functions is often known as kernel trick. Different types of kernel functions can be used based upon needs. Semi supervised support vector machine overcomes the disadvantage of supervised support vector machine.

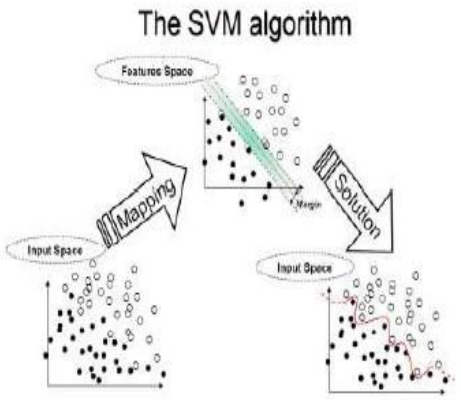

Fig 2: Semi Supervised Support Vector Machine classifier

The semi supervised support vector machine which uses small quantity of labeled data provides better accuracy than the supervised support vector machine. The semi supervised support vector machine uses the concept of maximal margin hyperplane. There are many hyperplanes available but we have to find the hyperplane that have maximum margin. The maximum margin hyperplane have less generalization errors. The semi supervised support vector machine finds the maximal margin hyperplane and thus the semi supervised support vector machines are often called the maximal margin classifier.

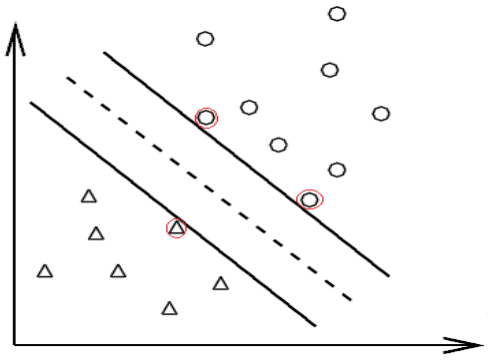

Fig 3: Maximum Margin Hyperplane

The learning task of the semi supervised support vector machine is given as :

$yi(w.\ \phi(\ xi) + b) >= 1$, where i = 1,2,3…n (1)

where yi represents the output labels of the training data sets and xi represents the training samples.

The dual lagrangian for the constrained optimization problem is given as :

$\alpha i = \sum i=1n\ \alpha i - 1/2 \sum i=1n \sum j=1n\ \alpha i \alpha j\ yi\ yj\ K(xi,xj)$

Where K(xi,xj) denotes the Kernel Functions whose value is equivalent to $K(xi,xj) = \phi(xi).\phi(xj)$

Once the value of αi's are found using the quadratic programming techniques, the parameters w and b are calculated with the help of the following equations:

$W = \sum i\ \alpha i\ yi\ \phi(xi)$ (2)

$\alpha i\ \{\ yi\ (\sum j\ \alpha i\ yj\ \phi(xi).\phi(xj) + b\ ) -1\ \} = 0$ (3)

Finally a test sample or instance s is classified using the following decision function equation:

$f(s) = \text{sign}(w.\ \phi(s) + b) = \text{sign}(\sum_{i=1}^{n} \alpha_i\ y_i\ K(x_i,x_j) + b)$

The value of the decision function f(s) will decide the label or class of the test samples. If the value of the decision function f(s) = 1, then the test sample will be classified as a positive class and if the value of the decision function is 0 i.e. f(s) = 0 then the test samples will be classified as the negative class or given negative label.

Thus, the sign of the decision function f(s) is checked for the labelling of the unlabeled samples or test samples.

3. EXPERIMENT

3.1 Dataset Used in Training and Testing

The dataset used for the training and testing purpose is NSL-KDD [14]. The reason for using NSL-KDD dataset is that it is advanced version of KDD99 dataset. NSL-KDD dataset does not include redundant records. Thus, performance of machine is evaluated in a better way.

The training dataset consist of total 1012 records is consist of two parts unlabeled data and labeled data. The training dataset consist of 739 unlabeled records and only 273 labeled records. Thus, training dataset consist of large portion of unlabeled data and small portion of labeled data. Testing dataset consist of 1386 records for testing purpose. Testing dataset consist of records that are not part of the training dataset. Thus, the machine is not biased towards some particular type of records.

Testing accuracy of the model is calculated as number of records correctly predicted to the total number of records present in the testing dataset. Testing accuracy tells how the model judge the labels of records which are previously not seen that is which are new. So, good testing accuracy of the model is desirable.

3.2 Experiment Result

The semi supervised support machine is first trained with the collection of labeled and unlabeled data. Five kernels are used for the purpose of training and testing. The testing accuracy of the model is calculated along with the training and testing time of the model. Testing accuracy, training and testing time of the model is depends on the type of the kernel function used along with the kernel parameters. These kernel parameters are gamma used in radial basis kernel, degree of polynomial used in log, multiquadric, power, inverse, multiquadric kernel functions. The value of the various kernel parameters are taken as g=1,d=1,2 , c=1

Training dataset = 739(unlabeled data) + 3(labeled data)

= 1012

Testing dataset = 1386 (distinct records)

TABLE 1: Testing Accuracy of S3VM model on NSL-KDD dataset

Kernel Function	Testing Accuracy
Radial Basis Function(RBF)	85.5699%
Log	90.7647%
Mutiquadric	91.7749%
Power	91.9191%
Inverse Mutiquadric	85.28139%

TABLE 2: Training and Testing time for various Kernel Functions

Kernel Function	Training Time (in sec)	Testing Time (in sec)
Radial Basis Function(RBF)	111.431	4.604
Log	147.915	4.024
Multiquadric	85.322	3.671
Power	77.126	3.495
Inverse Multiquadric	94.483	3.609

4. CONCLUSION

This paper proposes a solution for achieving performance and accuracy in the network system by semi supervised support vector machine. All the kernel functions give high accuracy even using small amount of labeled data and large amount of unlabeled data. Also testing time of all the kernel functions is quite less. We would like to extend my work in the following two directions. First, we would like to investigate the effectiveness of our algorithm in some other datasets suchas DEFCON and UCSB, as well as noisy and incomplete datasets. Second, we would like to address the intrusion detection problem in the domain of mobile wireless networks.

5. REFERENCES

[1] W. Lee and S. J. Stolfo, 1998 "Data mining approaches for intrusion detection," at USENIX Security Symposium.

[2] Damiano Bolzoni and Sandro Etalle, "Approaches in Anomaly-based Network Intrusion Detection Systems"

[3] D. S. Kim and J. S. Park, 2003, "Network-Based Intrusion Detection with Support Vector Machines," Information Networking.

[4] Jimin Li, Wei Zhang and Kunlun Li, April 2010 , "A Novel Semi-supervised SVM Based on Tritraining for Intrusition Detection", Journal of computers, Vol. 5, No.4.

[5] Intrusion Detection http://www.wikipedia.org/Intrusion_Detection.

[6] Xiaojin Zhu, "Semi-Supervised Learning Literature Survey", Computer Sciences TR 1530 University of Wisconsin – Madison.

[7] Yuanqing Li *, Cuntai Guan, Huiqi Li, Zhengyang Chin, "A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system", Pattern Recognition Letters 29 (2008) 1285–1294.

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

[9] X. Xu and X. Wang, 2005, "An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines," Advanced data Mining and Applications, First International Conference, ADMA.

[10] http://en.wikipedia.org/wiki/Principal_component_analysis.

[11] I. Lindsay Smith, 2002, "A Tutorial on Principal Components Analysis," Cornell University, USA.

[12] S. Mukkamala & A. H. Sung, 2003, "Feature Selection for Intrusion Detection using Neural Networks and Support

Vector Machines," 82nd Annual Meeting of the Transportation.

[13] Data Mining: Methods and Techniques by ABM Shaukat Ali, Sleh A.Wasimi.

[14] http://www.iscx.ca/NSL-KDD/.

[15] Glenn Fung and O. L. Mangasarian, "Semi-Supervised Support Vector Machines for Unlabeled Data Classification", Computer Sciences Department University of Wisconsin 1210 West Dayton Street Madison, WI 53706.