World of Computer Science and Information Technology (WCSIT)
ISSN: 2221-0741
Vol. 2, No. 7, 215-219, 2012

A Comparison of Support Vector Machine and Multi-Level Support Vector Machine on Intrusion Detection

Milad Aghamohammadi
Department of Computer Engineering
Iran University of Science and Technology
Tehran, Iran

Morteza Analoui
Department of Computer Engineering
Iran University of Science and Technology
Tehran, Iran

Abstract—Accessibility and openness of the Internet cause increase information security risk. Information security means protecting information from unallowed access, use, disruption, change and etc. This paper is about Intrusion Detection. The main goal of IDS (Intrusion Detection System) is to protect the system by analyzing users behaviors and habits when they are working with system, detect behaviors that don't match with previously learned normal behaviors patterns and raise a warning. Support Vector Machine (SVM) is a classification method that used for IDS in many researches. We compare performance of SVM and Multi-Level Support Vector Machine (MLSVM) as a new edition of SVM on a challenging intrusion detection data set based on KDD'99 with name NSL-KDD. Our experiments indicate that MLSVM is more suitable for this data set rather than SVM.

Keywords- Intrusion Detection System; Support Vector Machine; Multi-Level Support Vector Machine; Pattern Recognition; Classification.

I. INTRODUCTION

Over the years, to design intrusion detection systems, many techniques have been used by researchers and designers. Anderson was the first one who worked on intrusion detection in 1980[1]. Since then, various discrimination techniques were proposed, ranging from support vector machines. Many complete systems have been designed and used on live computer system. However, despite of over 25 years of research, because of the rapid development of information processing system and the consequent discovery of new vulnerabilities, also due to fundamental difficulties in achieving an accurate declaration of an intrusion, the topic is still popular. Intrusion systems are known for high false rate and more research effort is still concentrated on finding effective intrusion and non-intrusion discriminates [2-5]. However present intrusion detection systems have many problems [6].

In[6], strategies of data mining and expert system are combined to design an intrusion detection system(IDS). This strategy appeared to be promising but there are some problem in structural and system performance. However, combining multiple techniques in designing the IDS is new topic and needs more research and improvement. Valdes [7] proposed a new approach by using sensor correlation, in which alarms from different components in the detection system are analyzes and correlated at different levels. Multi-sensor data fusion is named as another method to correlate and draw conclusion from data which can be gathered from many distributed sources.

Lack of exactness and in consistency in the network traffic patterns, have caused a number of approaches toward intrusion detection system based on "Soft computing" [8] to be proposed [9]. In this work, more exact solution to the computationally hard task of detecting abnormal patterns corresponding to intrusion is proposed. In [10] a fuzzy rule based system is used to present a soft computing approach toward intrusion detection.

[11] Suggest a machine learning based approach for intrusion detection. [12] Applies a combination of protocol analysis and pattern matching approach for intrusion detection.

In [13] an approach toward intrusion detection by analyzing the system activity for similarity with the normal flow of system activities using classification trees is proposed.

This paper is organized as followed:

Section II recalls the SVM and MLSVM methods. Experimental results are presented in Section III and conclusion is in Section IV.

II. PROPOSED METHOD

A. Support Vector Machines

One of the useful method for classification in high dimensional problems is SVM. In this method, a hyperplane with has maximal margin from patterns of each class separate two class of data. SVM uses the kernel trick [14] if two class of data can not separate in linearly. Kernel maps input data to a higher dimensional space where may be find a hyperplane that can separate two class of data linearly. Using the kernel function allow to SVM avoids the costly computation in new high dimensional space.

Objective function of none linear SVM model is

$$\min_{W,\xi,b} \quad L_P = \frac{1}{2}\|W\|^2 + C(\sum_{i=1}^{n}\xi_i)^k \qquad (1)$$

$$s.t. \quad y_i(W^T.\Phi(x_i)+b) \geq 1-\xi_i,$$
$$\xi_i \geq 0, i = 1,...,n$$

Where x_i is *i*th pattern of training set, $y_i \in \{+1,-1\}$ is class label of x_i, $\Phi : R^d \mapsto H$ is a mapping function, ξ_i is slack variable of x_i that let to x_i for misclassification, $W \in H$ is normal vector of the separating hyperplane, $b \in \Re$ is distance of hyperplane from the origin and C is a parameter that define by user and control the trade-off between width of margin and misclassification risk.. With Lagrangian multiplier method we can insert constraint of (1) to objective function. For all positive value of k, (1) will be a convex problem and can solve dual of it. For converting L_P to its dual form, must be vanish of gradient of L_P with respect to W and b and ξ. This type of dual in [15] is called the Wolf dual. For k=1, dual of (1) has been changed to

$$\max_{\alpha} \quad L_D = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum \alpha_i\alpha_j y_i y_j K(x_i, x_j) \qquad (2)$$

$$s.t. \quad 0 \leq \alpha_i \leq C,$$
$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

Where, α_i is a positive Lagrangian multiplier for *i*th inequality constraint in (1) and $K(x_i, x_j)$ is kernel function used for computing $\Phi(x_i).\Phi(x_j)$. After the training SVM and finding values of α_i, class label of unknown pattern x' has been defined as follow:

$$SVM(x') = sign(\sum_{i=1}^{n}\alpha_i y_i \Phi(x_i).\Phi(x') + b)$$
$$= sign(\sum_{i=1}^{n}\alpha_i y_i K(x_i, x') + b) \qquad (3)$$

Note that the class label of unknown pattern depends only on training points that have nonzero α_i. Such training points are called "Support vectors".

With solving (2), normal vector (W) of hyperplane will be

$$W = \sum_{i=1}^{n}\alpha_i y_i x_i \qquad (3)$$

B. Multi-Level Support Vector Machine (MLSVM)

We in [16] proposed a method that its name is MLSVM. As it have been mentioned before, in finding the margins in SVM, the only samples that matches our pattern are the one's that are close to the hyperplane and other samples have been found irrelevant to our pattern elsewhere.

"Fig. 1" shows a synthetic dataset which consists of two classes of +1 and -1. Samples that selected with SVM as support vectors are marked with a circle around them. As seen, only the places that are marked are specified to match favored results the other conclusions in this case have no value in defining vectors. But with, carefully observing it's noticeable that one of SVM results that are in +1 are the one which are encircled by do bold lining and this shows the existence of data in an area which had given similar results in other cases causes deduction in the width of margin (in the mentioned area), thus, the most favored outcomes and so it ruins the chances of separating hyperplanes.

Meanwhile, the rest of the data are located in different places and the majority of data are pointing elsewhere. It is obvious that this discussion with the existence of the kernel brought to this conclusion that when this result is sought a different environment it could have separable linear outcomes.

The omittance of outlier brings up this crucial point that one of the pre-processes happens before classifications. Although, the outcomes that follow a different pattern of repetition is omitted, but the possibility of cases such as "Fig. 1" is always possible. As it have been pointed out in "Fig. 1" the only results that are close to favored margin are considered relevant –regardless of the data and the places of their positioning in confirming our favored results.

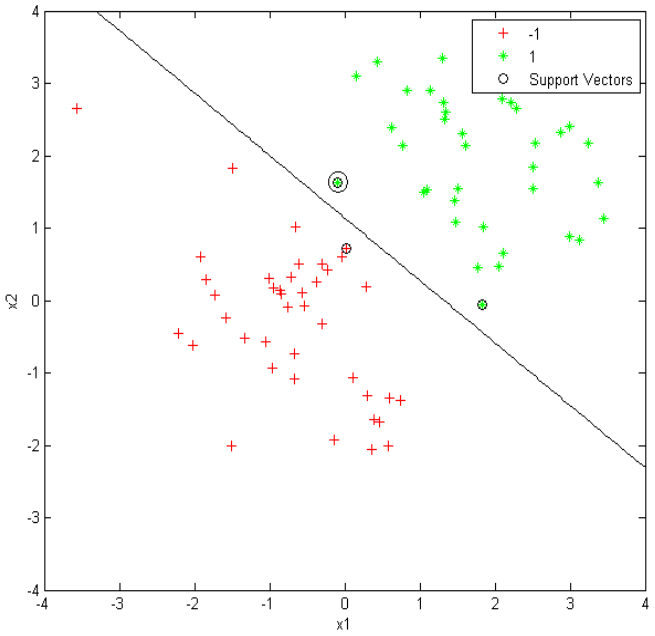

Figure 1. Artificial dataset and final decision of the SVM [16]

The outcome although vary but they contain useful details that helps in classification of so called data. To get resolve hidden information embedded in the data we use MLSVM.

In the follow method and our first step the where about of possible answer is found by support vector indicates the primary resolution which we refer to it as step 1. In the current step some data are selected as support vector.

The new data is derived from old ones with the difference that certain support vectors have been removed from the equation. In second step, by using SVM on the new data in order to find new separating hyperplane. In future steps same as this step the support vectors are removed and by using a SVM on these data, we could ascertain new separating hyperplanes.

The results of 3timings of this method on the data in "Fig. 1", had been brought up on "Fig. 2". In this figure the hyperplane that we have in step 2 and 3 are different then the one presented in step 1 but their differences compare to each other is tolerable and by comparison their hyperplanes are in accordance to step 1 and it shows more general to the main pattern.

III. EXPERIMENTAL RESULTS

A. Dataset Description

Public domain dataset named KDD Cup 1999 dataset [17] that is based on 1998 DARPA Lincoln Lab network connection. KDD'99 is a very famous dataset in the intrusion detection domain, and it has been used widely for the evaluation of various intrusion detection techniques but has an important disadvantage. Number of redundant patterns is a lot, which causes classifiers to be biased toward the frequent patterns, and also infrequent patterns cannot be learned correctly by usual methods (Table I), while infrequent patterns are usually more harmful to networks. In addition, redundant patterns in test set (Table II) can change the results of experiments and disguise the reality of performance of classifiers in the Intrusion detection.

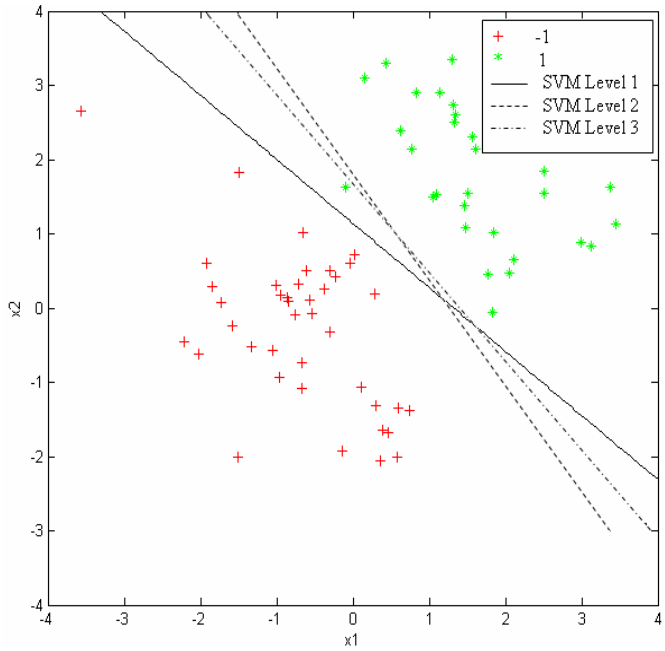

Figure 2. The 3 descending levels of MLSVM that have processed on the data of "Fig. 1" [16]

Tavallaee et al. [18] proposed a subset of KDD'99 data set to solve some of the inherent problems on that (NSL-KDD). NSL-KDD has reasonable number of patterns in train and test set, so it doesn't need to randomly select a subset of patterns for train and test set and can evaluate methods on its whole dataset. NSL-KDD has some improvements rather than KDD'99 data set. Training set contains only distinct patterns and exactly same patterns had been deleted. Also redundant records had been removed from test set.

TABLE I. STATISTICS OF REDUNDANT PATTERNS IN THE KDD TRAIN SET [18]

	Original Records	**Distinct Records**	**Reduction Rate**
Attacks	3,925,650	262,178	93.32%
Normal	972,781	812,814	16.44%
Total	4,898,431	1,074,992	78.05%

TABLE II. STATISTICS OF REDUNDANT PATTERNS IN THE KDD TEST SET [18]

	Original Records	Distinct Records	Reduction Rate
Attacks	250,436	29,378	88.26%
Normal	60,591	47,911	20.92%
Total	311,027	77,289	75.15%

They also create a more challenging subset of the KDD data set with names $KDDTrain^{+}$ and $KDDTest^{+}$ that includes 125,973 and 22,544 records, respectively. By using 21 learned machines (7 learners, each trained 3 times) difficulty levels for each pattern had been defined and created the $KDDTest^{-21}$ test set which dose not include records with difficulty level 21 out of 21.

We used "KDDTrain+_20Percent" and "KDDTest-21" for train and test set, can be downloaded from [19]. Tables III and IV show the details of used data sets.

TABLE III. NUMBER OF ATTACK AND NORMAL PATTERNS IN KDDTRAIN+_20PERCENT AND KDDTEST-21 DATA SETS

	KDDTrain+_20Percent Data set	KDDTest-21 Data set
Attacks	11,742	9,698
Normal	13,450	2,152
Total	25,192	11,850

TABLE IV. NUMBER OF PATTERNS IN EACH DIFFICULTY LEVEL GROUP IN KDDTRAIN+_20PERCENT AND KDDTEST-21 DATA SETS

	KDDTrain+_20Percent Data set	KDDTest-21 Data set
0-5	81	585
6-10	173	838
11-15	1,336	3,378
16-20	11,107	7,049
21	12,495	0
Total	25,192	11,850

B. Parameter Selection

We use 2 levels of MLSVM and all of our SVM with Gaussian kernel. In order to determine the σ parameter, different amounts $\{2^{-10},2^{-9},...,2^{3},2^{4},2^{5}\}$ have all been tested and the best amount with the finest accuracy is selected. Also, the amount of C in SVM and 2 levels of MLSVM is a constant and it had been set on 1. The data have been converted to normal standard distribution.

C. Experimental Results

Table V contains classification accuracy and number of misclassified patterns for both SVM and MLSVM methods. Clearly be seen that MLSVM has better accuracy than SVM in this data set.

TABLE V. THE CLASSIFICATION ACCURACY AND NUMBER OF MISSCLASSIFIED PATTERNS IN SVM AND MLSVM ON KDDTEST-21 DATASET

Methods	Accuracy (Percent)	Number of misclassified patterns
SVM	87.05	1,535
MLSVM	**87.15**	**1,523**

In a comparison that train and test set select randomly from a dataset for each iterations, 0.10% of difference can not be statistically significant difference, but in this comparison we had used predefined train and test set, also huge number of patterns in test set make this amount of difference a significant difference.

IV. CONCLUSION

In this research, we compare SVM and MLSVM in intrusion detection field. This comparison was performed on data set NSL-KDD. We used challenging data set $KDDTest^{-21}$ as test set.

Results of comparison showed that MLSVM has better accuracy rather than SVM in this data set.

[1] S Mukkamala and A.H. Sung, "A comparative study of techniques for intrusion detection," in Tools with Artificial Intelligence, Proceedings of the 15th IEEE International, New Mexico, 2003, pp. 570-577.

[2] W.H. Allen, G.A. Marin, and L.A. Rivera, "Automated detection of malicious reconnaissance to enhance network security," in SoutheastCon, Proceedings. IEEE, Florida, 2005, pp. 450-454.

[3] S. Zaman and F. Karray, "Feature Selection for Intrusion Detection System Based on Support Vector Machine," in 6th Annual IEEE Consumer Communications & Networking Conference IEEE CCNC, 2009.

[4] Z. Yuan and X. Guan, "Accurate classification of the internet traffic based on the SVM method," in Proceedings of the 42th IEEE International Conference on Communications (ICC), 2007, pp. 1373-1378.

[5] S. Srinoy, "Intrusion Detection Model Based On Particle Swarm Optimization and Support Vector Machine," in The IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA), Bangkok, 2007, pp. 186-192.

[6] A. Sodiya, H. Longe, and A. Akinwale, "A new two-tiered strategy to intrusion detection, , Vol. 12 No. 1, pp. (2004).," Information Management & Computer Security, vol. 12, no. 1, pp. 27-44, 2004.

[7] A. Valdes and K. Skinner, "Probabilistic alert correlation," Recent Advances in Intrusion Detection (RAID), vol. 2212, pp. 54-68, 2001.

[8] Ethem Alpaydın, Introduction to Machine Learning, 2nd ed., Thomas Dietterich, Ed. London, England: The MIT Press, 2010.

[9] Chet Langin and Shahram Rahimi, "Soft computing in intrusion detection: the state of the art," Journal of 8 Ambient Intelligence and Humanized Computing, vol. 1, no. 2, pp. 133-145, 2010.

[10] Ajith Abraham, Ravi Jain, Sugata Sanyal, and S.Y. Han, "SCIDS: A Soft Computing Intrusion Detection System, , A. Sen et al. (Eds.) Springer Verlag, Germany, Lecture Notes in Computer Science," in 6th International Workshop on Distributed Computing (IWDC 2004), 2004.

[11] Vegard Engen, "Machine Learning for Network Based Intrusion Detection: An Investigation into Discrepancies in Findings with the KDD Cup '99 Data Set and Multi-Objective Evolution of Neural Network Classier Ensembles for Imbalanced Data," PhD thesis, School of Design, Engineering and Computing, Bournemouth University, 2010.

[12] T. Abbes, A. Bouhoula, and M. Rusinowitch, "Protocol analysis in intrusion detection using decision tree, in Proc. Int. Conf. Inf. Technol.: Coding Comput.," in Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on, Nancy, 2004, pp. 404–408.

[13] Evgeniya Nikolova and Veselina Jecheva, "Some similarity coefficients and application of data mining techniques to the anomaly-based IDS," Telecommunication Systems, pp. 1-9, December 2010.

[14] M. Aizerman, E. Braverman, and L. Rozonoer, "Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning," Automation and Remote Control, vol. 25, pp. 821-837, 1964.

[15] R. Fletcher, Practical Methods of Optimization, 2nd ed.: John Wiley and Sons, Inc., 1987.

[16] Milad Aghamohammadi and Morteza Analoui, "Multi-Level Support Vector Machine," World of Computer Science and Information Technology Journal (WCSIT), vol. 2, no. 5, pp. 174-178, June 2012.

[17] KDD Cup 1999 Data, University of California, Irvine, [online] 1999, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (Accessed: 10 May 2012).

[18] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[19] The NSL-KDD Data Set, Information Security Center of eXcellence, http://www.iscx.ca/NSL-KDD (Accessed: 25 June 2012).