A Multiclass SVM Classification Approach for Intrusion Detection

Santosh Kumar Sahu(✉) and Sanjay Kumar Jena

National Institute of Technology, Rourkela, Odisha, India
santoshsahu@hotmail.co.in, skjena@nitrkl.ac.in

Abstract. As the number of threats to the computer network and network-based applications is increasing, there is a need for a robust intrusion detection system that can ensure security against threats. To detect and defend against a specific attack, the pattern of the attack should be known a priori. Classification of attacks is a useful way to identify the unique patterns of different type of attack. As a result, KDDCup99, NSLKDD and GureKDD datasets are used in this experiment to improve the learning process and study different attack patterns thoroughly. This paper proposed a multi-class Support Vector Machine classifier(MSVM), using one versus all method, to identify one attack uniquely, which in turn helps to defend against the known as well as unknown attacks. Experimentally, the proposed scheme provides better detection accuracy, fewer false positives, and lesser training and generalization error in comparison to the existing approach.

Keywords: MSVM · Threats · KDD corrected · NSL KDD · Gure KDD

1 Introduction

The highly integrated electronic world is an effect of technological development over decades. The number of malicious activities and attacks are also growing besides the advances in security against threats. To mitigate the situations, various attempts are made to control the attack activities. There is a need to improve and innovate different techniques for the detection of intrusion against the enormous amount of malicious attempts on networks [1]. To detect and countermeasures such attacks, multi-class problem should be adapted. Most of the learning methods are biased in multiclass problems. As a result proper combination approaches should be used to improve the detection rate, overcome the bias and over fitting situation. In this work, Support Vector Machine (SVM) learning approach is used as a base learner to solve the multi-class problem.

1.1 Support Vector Machine

The classification is used to achieve high accuracy for classifying the maximum number of instances with the small number of training samples. It gives better

N. Bjørner et al. (Eds.): ICDCIT 2016, LNCS 9581, pp. 175–181, 2016.
DOI: 10.1007/978-3-319-28034-9_23

result for two class classification problem [4]. It maps input vectors to a high dimensional feature space. Both linear and non-linear data is separated by a hyperplane in two classes. The hyperplane is found with the help of support vector (training tuples) and margin (defined by support vectors) [5]. SVMs are the successful and resilient classification algorithms [4]. The SVM supports only binary classification and deals with maximizing the margin which is the minimum distance from nearest example to the separating hyperplane. The concept of SVM can be extended to multiclass classification [6].

1.2 Multiclass Support Vector Machine

The multiclass problem needs to be decomposed into several binary class problems. Each of the binary classifiers is applied to new data point and the frequency of the number of times the point is assigned to the same label is counted and labeled with the highest count. The popular two methods for decomposition of multi-class problem discussed as follows: [7].

One-verses-all. One-verses-all is also called as winner takes all strategy. This is the simplest approach to reduce the problem of classification from k classes into k binary problems. Each problem is different from other $k-1$ problems. This approach requires k binary classes in which we train kth classifier with positive example and belonging to class k and negative examples belonging to other $k-1$ classes. An unknown example is tested and the classifier for which maximum output is produced is considered to be the winner class. That class label is assigned to that example. Although this approach is simple, its performance can be compared with more complicated approaches [8].

One-versus-one. For every pair of different classes, one binary classifier is constructed. In this way, the multi-class problem is broken into a series of a set of binary class problems; so that we can apply SVM model for each pair of classes. Total $k(k-1)/2$ classifiers are needed to classify the unknown data. The binary classifier is trained taking one class as positive and other class as negative. For a new data point x if that classifier classifies x in first class, then a vote is added to that class. If the classifier classifies x in second class the vote is added to the second class. This process is repeated for each of the $k(k-1)/2$ classifiers. Finally, label of the class with maximum number of votes is assigned to the new data point x. In this way the class to which the unknown data point belongs is predicted [8,9].

1.3 Intrusion Dataset

The intrusion dataset takes a vital role in model assessment and learning process. In this experiment the benchmarked intrusion datasets are used. The public datasets namely KDDCup99, NSLKDD, and GureKDD are used in learning and evaluation process. The details about the datasets are discussed in [10].

1.4 Motivation and Objective

As the number of attacks are growing day by day, it becomes utmost essential to classify the specific attack type with maximum accuracy that motivated to implement the MSVM IDS. The objective of this work is to detect the exact type of attacking effort to the network that helps to analyze, countermeasure and implement security policies.

The rest of the paper is organized as follows: The existing work on SVM and multiclass SVM discussed in Sect. 2. The result and discussion is presented in Sect. 3. The comparison of the proposed approach with existing approaches elaborated in Sect. 4, and finally, Sect. 5 conclude the work.

2 Related Work

Mathur et al. [3] has extended the SVM approach to multiclass SVM. He has undertaken a multiclass classification based on a single optimization. Chen et al. [12] uses hierarchical SVM for clustering the classes into binary tree. The clusters are formed by arranging the classes into undirected graph. Each node of the tree is a binary SVM classifier . Hsu et al. [14] has proposed two methods one by considering all data at once and second is a decomposition implementation.

According to latest research, there are a lot of attempts to improve IDS using the data mining and machine learning techniques. In this paper, a multi-class SVM approach is proposed to detect the specific attack types with low false alarm rate. The accuracy is calculated for each of the five classes i.e., Normal, DOS, U2R, R2L, and Probe attack.

3 Result and Discussion

In this paper, one against all approach of MSVM is implemented on Matlab R2015a. To improve the detection accuracy, cross validation and re-sampling

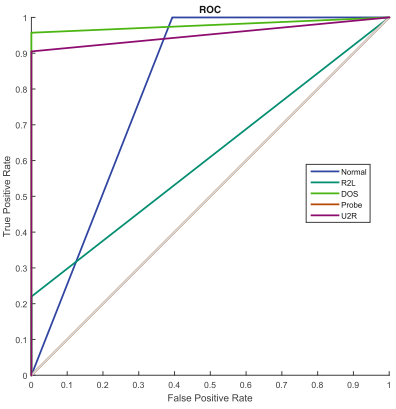
Table 1. The details of datasets

Dataset	No. of instances for training	No. of instances for testing	Number of instances correctly classified	No. of Class	Accuracy
KDD Corrected	77291	311029	284421	38	91.445 %
NSL-KDD	47736	125973	118447	23	94.025 %
Gure-KDD	160904	178810	177283	28	99.146 %

Table 2. The confusion matrix on KDD corrected dataset

790	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
16	98	0	0	0	0	0	0	0	0	984	0	0	0	0	0	0	0	0	0
0	0	12	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	4	0
0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0
2	0	0	3959	1	0	0	1	0	0	401	0	0	3	0	0	0	0	0	0
0	0	1	0	150	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	296	0	0	0	0	5	0	0	0	2	0	3	0	0	0
0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	4421	0	0	0	579	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	6	0	0	0	1039	0	4	0	0	3	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	7	57989	5	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	9	1	46	2	1	0	15	4	5	48539	9	65	2	0	82	2	11707	103	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	13	73	0	0	0	0	1	0	0	0
0	0	0	0	23	0	0	0	0	0	1	0	330	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	0	0	0	0	756	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0	0	17	0	0	0	104	614	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	1624	0	0	0	0
0	0	2	2	0	0	0	0	0	0	0	9	0	0	0	0	0	0	4	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	164090	0	0
0	0	0	0	0	0	0	0	0	0	0	106	0	0	0	0	0	7635	0	0
0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	1	0	2395	0	0
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0	0	0	1505	0
0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	5	0	0	3	0	0	0	0	0	0
0	0	2	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	2	0

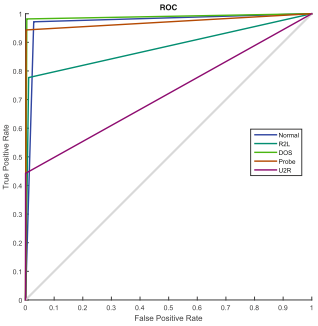
methods are applied on U2R and R2L distributions. The three intrusion datasets namely KDD corrected , NSL-KDD and Gure-KDD used for training and testing purpose. The details about the dataset and detection accuracy is given in the Table 1. The confusion matrices for KDD corrected, Gure-KDD and NSL-KDD dataset are given in Table 2, Fig. 1b and Table 3 respectively.



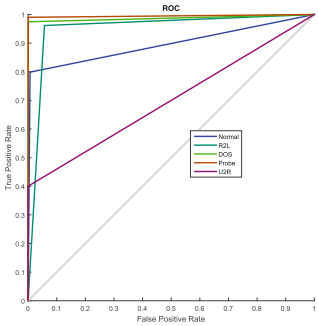
(a) GureKDD

0	0	0	9	0	0	0
864	0	0	15	0	0	0
1	0	0	0	0	0	0
0	1	0	10	0	0	0
0	0	0	1	0	0	0
0	0	0	10	0	0	0
0	0	0	1	0	0	0
0	0	0	6	0	0	0
0	0	0	1	0	0	0
0	0	0	1	0	0	0
0	0	0	8	0	0	0
0	49	0	1	0	0	0
0	0	0	7	0	0	0
0	0	35	0	0	0	0
0	0	0	1	0	0	0
0	0	0	9	0	0	0
3	0	2	174864	0	3	1
0	0	0	1	0	0	0
0	0	0	4	0	0	0
0	0	0	5	0	0	0
1	0	0	28	0	0	0
0	0	0	2	0	0	0
0	0	0	4	0	0	0
0	0	0	9	1076	0	0
0	0	0	1	0	0	0
0	0	0	1369	0	380	0
0	0	0	4	0	0	15
0	0	0	8	0	0	0

(b) The confusion matrix on GureKDD dataset



(c) NSLKDD



(d) KDD-Corrected

Fig. 1. ROC curve for different datasets (a, c and d) and confusion matrix (b) for GKDD dataset

Table 3. The confusion matrix on NSL-KDD dataset

65781	4	116	80	2	0	0	30	151	52	423	1	345	7	0	21	37	63	16	11	203
4	41200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	9	0	0
380	0	492	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0
111	0	0	3449	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	5	0	0	2852	0	0	0	0	0	0	0	0	0	0	0	27	2	4	0	0
7	0	0	0	0	884	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
271	0	0	13	0	0	1209	0	0	0	0	0	0	0	0	0	0	0	0	0	0
287	0	0	0	2	0	0	0	0	3325	0	0	1	0	0	0	13	0	5	0	0
66	0	0	0	0	0	0	0	2580	0	0	0	0	0	0	0	0	0	0	0	0
97	0	0	0	0	0	0	0	0	0	0	0	0	0	104	0	0	0	0	0	0
749	0	2	0	0	0	0	200	0	0	0	0	0	0	0	5	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	1	0
26	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0

4 Comparison

The existing approach by [1] failed to detect the R2L and U2R attack patterns. As a result, the accuracy of that model is 91.67 % and only KDDCup99 dataset is used. In the proposed MSVM approach, three datasets are used and pre-processed properly before the model formation. The detection accuracy of the proposed scheme is 99.146 % on GureKDD, 94.025 % on NSLKDD and 91.445 % on KDDCorrected Dataset.

5 Conclusion

In this paper, an MSVM classifier is used to detect and identify the attacks by type. Evaluation has been done over the three benchmark intrusion datasets. Cross-validation and re-sampling methods are applied to improve the learning process to the datasets. The model can determine a particular known type of attack when the unknown instances need to be classified. This scheme provides a better detection accuracy and reduces the complexity of the model. Further, it can detect the least data distributions i.e. U2R and R2L attacks efficiently.

References

1. Ambwani, T.: Multi class support vector machine implementation to intrusion detection. In: Proceedings of the International Joint Conference on Neural Networks, vol. 3. IEEE (2003)
2. Mukkamala, S., Janoski, G., Sung, A.: Intrusion detection using neural networks and support vector machines. In: Proceedings of the 2002 International Joint Conference on Neural Networks, IJCNN 2002, vol. 2. IEEE (2002)
3. Mathur, A., Foody, G.M.: Multiclass and binary SVM classification: implications for training and classification users. IEEE Geosci. Remote Sens. Lett. **5**(2), 241–245 (2008)
4. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
5. Han, J., Kamber, M., Pei, J.: Data Mining, Southeast Asia Edition: Concepts and Techniques. Morgan kaufmann, Burlington (2006)
6. Lee, Y., Lin, Y., Wahba, G.: Multicategory support vector machines: theory and application to the classification of microarray data and satellite radiance data. J. Am. Stat. Assoc. **99**(465), 67–81 (2004)
7. Allwein, E.L., Schapire, R.E., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. J. Mach. Learn. Res. **1**, 113–141 (2001)
8. Aly, M.: Survey on multiclass classification methods. Neural Netw. 1–9 (2005)
9. Duan, K.-B., Keerthi, S.S.: Which is the best multiclass SVM method? An empirical study. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 278–285. Springer, Heidelberg (2005)
10. Sahu, S.K., Sarangi, S., Jena, S.K.: A detail analysis on intrusion detection datasets. In: 2014 IEEE International Advance Computing Conference (IACC). IEEE (2014)
11. Tavallaee, M., et al.: A detailed analysis of the KDD CUP 99 data set. In: Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications (2009)
12. Chen, Y., Crawford, M.M., Ghosh, J.: Integrating support vector machines in a hierarchical output space decomposition framework. In: 2004 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2004, Proceedings, vol. 2. IEEE (2004)
13. Lee, H., Song, J., Park, D.: Intrusion detection system based on multi-class SVM. In: Ślęzak, D., Yao, J.T., Peters, J.F., Ziarko, W.P., Hu, X. (eds.) RSFDGrC 2005. LNCS (LNAI), vol. 3642, pp. 511–519. Springer, Heidelberg (2005)
14. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. **13**(2), 415–425 (2002)