Model Selection for Kernel Based Intrusion Detection Systems

Srinvas Mukkamala[1], A. H. Sung[1], B. M. Ribeiro[2]

[1]Department of Computer Science, New Mexico Tech, Socorro NM 87801, U.S.A.
[2]Department of Informatics Engineering, University of Coimbra, Portugal
Email: {srinivas|sung|bmr@cs.nmt.edu}

Abstract

This paper describes results concerning the robustness and generalization capabilities of a supervised machine learning method in detecting intrusions using network audit trails. We also evaluate the impact of kernel type and parameter values on the accuracy with which a support vector machine (SVM) performs intrusion classification. We show that classification accuracy varies with the kernel type and the parameter values; thus, with appropriately chosen parameter values, intrusions can be detected by SVMs with higher accuracy and lower rates of false alarms.

Feature selection is as important for intrusion detection as it is for many other problems. We present support vector decision feature selection method for intrusion detection. It is demonstrated that, with appropriately chosen features, intrusions can be detected in real time or near real time.

1 Introduction

Intrusion detection is a problem of great importance to protecting information systems security, especially in view of the worldwide increasing incidents of cyber attacks. Since the ability of an Intrusion Detection System (IDS) to identify a large variety of intrusions in real time with accuracy is of primary concern, we will in this paper consider performance of SVM-based IDSs with respect to classification accuracy and false alarm rates, and their relation to parameter selection and kernel type.

AI techniques have been used to automate the intrusion detection process; they include neural networks, fuzzy inference systems, evolutionary computation, machine learning, etc. Several research groups recently have used SVMs to build IDSs. However, most groups that studied SVMs for IDS considered only a small set of kernels and parameters [1-5]. Although several groups have extensively considered model selection in SVMs, optimal parameters are usually domain specific. In this paper, we present a methodology to evaluate the impact of model selection (kernel types and parameter values) on the performance of a SVM to detect intrusions.

Data mining techniques have been introduced to identify key features that characterize intrusions [6-8]. We performed experiments to rank the importance of input features using support vector decision function for each of the five classes (normal, probe, denial of service, user to super-user, and remote to local) of network traffic patterns in the DARPA data. It is shown that using only the important features for classification gives better performance.

Intrusion detection data used for experiments is briefly explained in section 2. A brief introduction to model selection using SVMs for intrusion detection is given in section 3. In section 4, we analyze classification accuracies of SVMs using ROC curves. A brief introduction to feature selection and SVM-specific feature identification is given in section 5.

2 Data Used for Analysis

A subset of the DARPA intrusion detection data set is used for offline analysis. In the DARPA intrusion detection evaluation program, an environment was set up to acquire raw TCP/IP dump data for a network by simulating a typical U.S. Air Force LAN. The LAN was operated like a real environment, but being blasted with multiple attacks [9,10]. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted [6] for intrusion analysis. Attacks are classified into the following types.

Attack types fall into four main categories:

1. DOS: denial of service
2. R2L: unauthorized access from a remote machine
3. U2Su: unauthorized access to local super user (root) privileges
4. Probing: surveillance and other probing

3 Model Selection

In any predictive learning task, such as classification, both a model and a parameter estimation method should be selected in order to achieve a high level of performance of the learning machine. Recent approaches allow a wide class of models of varying complexity to be chosen. Then the task of learning amounts to selecting the sought-after model of optimal complexity and estimating parameters from training data [11,12].

Within the SVMs approach, usually parameters to be chosen are (i) the penalty term C which determines

the trade-off between the complexity of the decision function and the number of training examples misclassified; (ii) the mapping function Φ; and (iii) the kernel function such that $K(\mathbf{x}_i,\mathbf{x}_j)=\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$.

In the case of RBF kernel, the width, which implicitly defines the high dimensional feature space, is the other parameter to be selected [13].

We performed a grid search using 10-fold cross validation for each of the five faults in our data set.

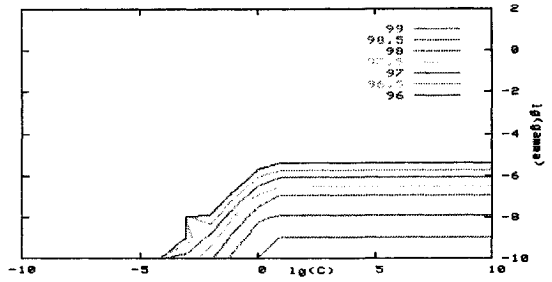

Fig. 1. SVM model for Normal.

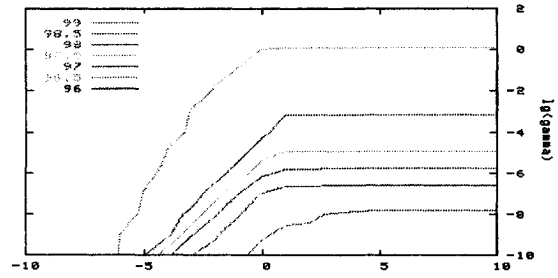

Fig. 2. SVM model for Probe.

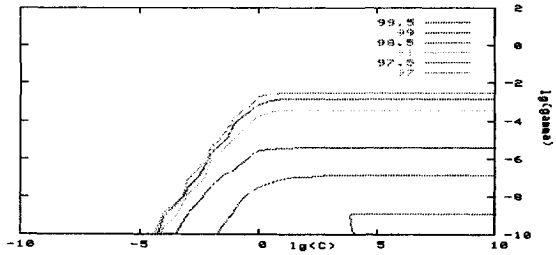

Fig. 3. SVM model for DoS.

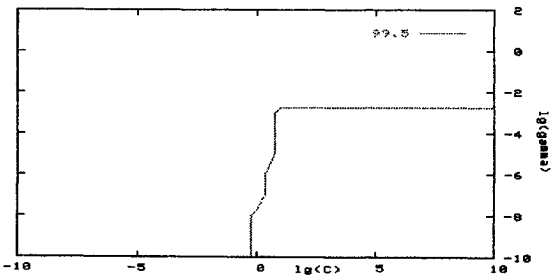

Fig. 4. SVM model for U2Su.

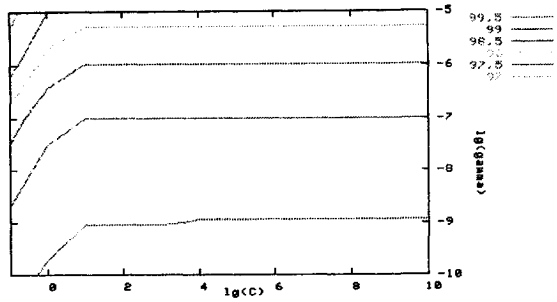

Fig. 5. SVM model for R2L.

First, we achieved the search of parameters C and γ in a coarse scale and then we carried through a fine tunning into the five detection faults proper space. Model selection results obtained through grid search are given in figures 1 to 5 for normal, probe, DoS, U2Su, and R2L, respectively.

4 ROC Curves

The Receiver Operating Characteristic (ROC) curves are generated by considering the rate at which true positives accumulate versus the rate at which false positives accumulate with each one corresponding, respectively, to the vertical axis and the horizontal axis in Figures 6 to 10.

The point (0,1) is the perfect classifier, since it classifies all positive cases and negative cases correctly. Thus an ideal system will initiate by identifying all the positive examples and so the curve will rise to (0,1) immediately, having a zero rate of false positives, and then continue along to (1,1).

Detection rates and false alarms are evaluated for the five-class pattern in the DARPA data set and the obtained results are used to form the ROC curves.

Figures 6 to 10 show the ROC curves of the detection models by attack categories as well as on all intrusions. In each of these ROC plots, the x-axis is the false alarm rate, calculated as the percentage of normal connections considered as intrusions; the y-axis is the detection rate, calculated as the percentage of intrusions detected. A data point in the upper left corner corresponds to optimal high performance, i.e, high detection rate with low false alarm rate [14].

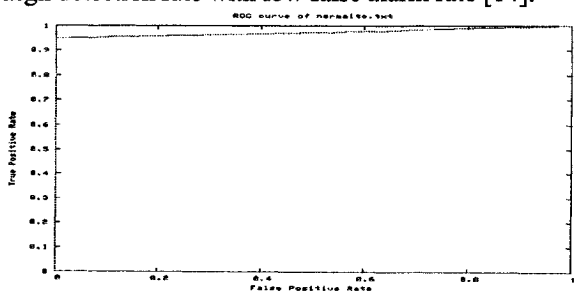

Fig. 6. SVM detection accuracy for normal.

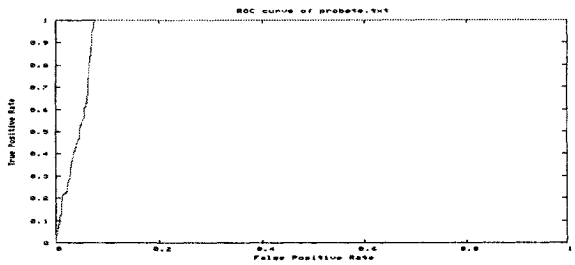

Fig. 7. SVM detection accuracy for probe.

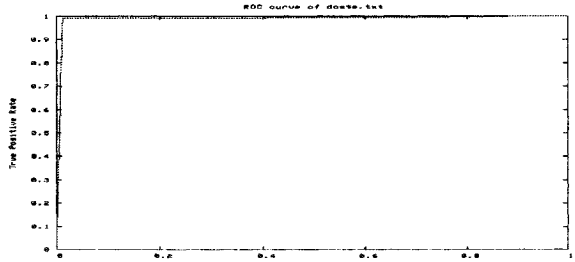

Fig. 8. SVM detection accuracy for DoS.

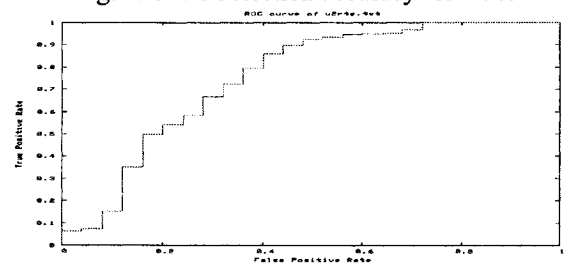

Fig. 9. SVM detection accuracy for U2Su.

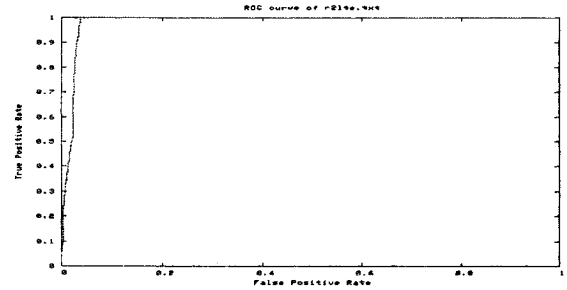

Fig. 10. SVM detection accuracy for R2L.

5 Feature Ranking and Selection

Feature selection is an important issue in intrusion detection. Of the large number of features that can be monitored for intrusion detection purpose, which are truly useful, which are less significant, and which may be useless? The question is relevant because the elimination of useless features (the so-called audit trail reduction) enhances the accuracy of detection while speeding up the computation, thus improving the overall performance of IDS. In cases where there are no useless features, by concentrating on the most important ones one may well improve the time performance of an IDS without affecting the accuracy of detection in statistically significant ways.

The feature selection problem for intrusion detection is similar in nature to various engineering problems that are characterized by:

- Having a large number of input variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ of varying degrees of importance to the output **y**; i.e., some elements of **x** are essential, some are less important, some of them may not be mutually independent, and some may be useless or noise (in determining the value of **y**)
- Lacking an analytical model that provides the basis for a mathematical formula that precisely describes the input-output relationship, $y = F(x)$
- Having available a finite set of experimental data, based on which a model (e.g. intelligent systems) can be built for simulation and prediction purposes

5.1 SVM-specific Feature Ranking Method

Information about the features and their contribution towards classification is hidden in the support vector decision function. Using this information one can rank their significance, i.e., in the equation

$$F(X) = \Sigma W_i X_i + b$$

The point X belongs to the positive class if F(X) is a positive value. The point X belongs to the negative class if F(X) is negative. The value of F(X) depends on the contribution of each value of X and W_i. The absolute value of W_i measures the strength of the classification. If W_i is a large positive value then the i^{th} feature is a key factor for positive class. If W_i is a large negative value then the i^{th} feature is a key factor for negative class. If W_i is a value close to zero on either the positive or the negative side, then the i^{th} feature does not contribute significantly to the classification. Thus, a ranking can be done by considering the support vector decision function.

We validate the ranking by comparing the performance of the classifier using all input features to that using the important features; and we also compare the performance of a classifier using the union of the important features for all fives classes.

Table 1 SVM detection accuracies

Class	Classifier Accuracy (%)	
	SVMs (41 features)	SVMs (6 features)
Normal	99.55	99.23
Probe	99.70	99.16
DoS	99.25	99.16
U2Su	99.87	99.87
R2L	99.78	99.78

Table 2 Most important feature descriptions

Class	Feature Description
6 Most Important Features	▪ source bytes: number of bytes sent from the host system to the destination system ▪ dst_host_srv_count: : number of connections from the same host with same service to the destination host during a specified time window ▪ count: number of connections made to the same host system in a given interval of time ▪ protocol type: type of protocol used to connect (e.g. tcp, udp, icmp, etc.) ▪ srv_count: number of connections to the same service as the current connection during a specified time window ▪ flag: normal or error status of the connection

6. Conclusions

A number of observations and conclusions are drawn from the results reported in this paper:

SVMs easily achieve high detection accuracy (higher than 99%) for each of the 5 classes of DARPA data, regardless of whether all 41 features are used, or only the important features for each class are used. Using the important features for each class gives the most accurate performance.

A grid search for intrusion detection (Figures 1 to 5) which seeks the optimal values of the constraint penalty for method solution and the kernel width (C,γ) has been performed. We demonstrate that the ability with which SVMs can classify intrusions is highly dependent upon both the kernel type and the parameter settings.

We note, however, that the difference in accuracy figures tend to be small and may not be statistically significant, especially in view of the fact that the 5 classes of patterns differ tremendously in their sizes. More definitive conclusions perhaps can only be drawn after analyzing more comprehensive sets of network data.

Acknowledgements

Partial support for this research received from ICASA (Institute for Complex Additive Systems Analysis, a division of New Mexico Tech), a DoD IASP, and an NSF SFS Capacity Building grants are gratefully acknowledged. We would also like to acknowledge many insightful discussions with Dr. Jean-Louis Lassez that helped clarify our ideas. The collaborative work of the third author was performed during a sabbatical visit to New Mexico Tech in 2004.

References

[1] Mukkamala, S., Janowski, G., Sung. A.H. (2002) Intrusion Detection Using Neural Networks and Support Vector Machines. Proceedings of IEEE International Joint Conference on Neural Networks 2002, IEEE press, pp. 1702-1707

[2] Fugate, M., Gattiker, J.R. (2003) Computer Intrusion Detection with Classification and Anomaly Detection, Using SVMs. International Journal of Pattern Recognition and Artificial Intelligence 17(3): 441-458

[3] Hu, W., Liao, Y., Vemuri., V.R. (2003) Robust Support Vector Machines for Anamoly Detection in Computer Security. International Conference on Machine Learning, pp. 168-174

[4] Heller, K.A., Svore, K.M., Keromytis, A.D., Stolfo, S. J. (2003) One Class Support Vector Machines for Detecting Anomalous Window Registry Accesses. *In* 3rd IEEE Conference Data Mining Workshop on Data Mining for Computer Security

[5] Lazarevic, A., Ertoz, L., Ozgur, A., Srivastava, J., Kumar, V. (2003) A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. In Third SIAM Conference on Data Mining

[6] Stolfo, J., Wei, F., Lee, W., Prodromidis, A., Chan, P.K. (1999) Cost-based Modeling and Evaluation for Data Mining with Application to Fraud and Intrusion Detection. Results from the JAM Project

[7] Mukkamala, S., Sung, A.H., (2003) Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines. Journal of the Transportation Research Board of the National Academics, Transportation Research Record No 1822: 33-39

[8] Mukkamala, S., Sung, A.H. (2003) Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligence Techniques. In International Journal on Digital Evidence, IJDE 3

[9] Kendall, K. (1998) A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems. Master's Thesis, Massachusetts Institute of Technology (MIT)

[10] Webster, S.E. (1998) The Development and Analysis of Intrusion Detection Algorithms. Master's Thesis, MIT

[11] Chapelle, O., Vapnik, V. (1999) Model selection for support vector machines. Advances in Neural Information Processing Systems 12

[12] Cherkassy, V. (2002) Model complexity control and statistical learning theory. Journal of natural computing 1: 109–133

[13] Cristianini, N., Taylor, J.S. (2000) Support Vector Machines and Other Kernel-based Learning Algorithms. Cambridge, UK: Cambridge University Press

[14] Egan, J.P. (1975) Signal detection theory and ROC analysis. New York: Academic Press