Biased Support Vector Machines and Kernel Methods for Intrusion Detection

K. Yendrapalli, S. Mukkamala, A. H. Sung, B. Ribeiro[1]
Department of Computer Science
New Mexico Institute of Mining and Technology
Socorro, New Mexico 87801, U.S.A.
[1]University of Coimbra
P-3030-290 Coimbra, Portugal
krish|srinivas|sung@cs.nmt.edu, bribeiro@dei.uc.pt

Abstract – This paper describes results concerning the robustness and generalization capabilities of kernel methods in detecting intrusions using network audit trails. We use traditional support vector machines (SVM), biased support vector machine (BSVM) and leave-one-out model selection for support vector machines (looms) for model selection. We also evaluate the impact of kernel type and parameter values on the accuracy of a support vector machine (SVM) performing intrusion classification. Through a variety of comparative experiments, it is found that SVM performs the best for detecting Normal and User to Super User, BSVM performs the best for Denial of Service attacks, and looms based on BSVM performs the best for Probe and Remote to Local.

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.

Index Terms—Intrusion detection, Model selection, Kernel machines, Support vector machines

I. INTRODUCTION

Intrusion detection attempts to detect actual attacks (as opposed to potential vulnerabilities) against networked hosts by analyzing network traffic. 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 compare performances of Biased Support Vector Machine (BSVM) and Support Vector Machine (SVM) for intrusion detection 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 different SVM implementations to detect intrusions [6].

The problem of multiclass classification, especially for systems like SVMs, doesn't present an easy solution. It is generally simpler to construct classifier theory and algorithms for two mutually-exclusive classes than for N mutually-exclusive classes. In this paper, we use BSVM that constructs N-class SVMs [7,8]. Most existing approaches for model selection use the leave-one-out (loo) related estimators which are considered computationally expensive. In this paper, we use Leave-one-out model selection for support vector machines (looms) that uses advance numerical methods which lead to efficient calculation of loo rates of different models [9].

Intrusion detection data used for experiments is briefly explained in section 2. Models generated by Biased Support Vector Machine using leave-one-out model for support vector machines (looms) is given in section 3. A brief introduction to model selection using SVMs for intrusion detection is given in section 4. .In section 5, we analyze classification accuracies of SVMs using ROC curves. Section 6 presents the results and discussion. Summary and Conclusions are given in section 7.

II. INTRUSION DATA USED for ANALYSIS

A sub set of the DARPA intrusion detection data set is used for off-line 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 [10,11]. For each TCP/IP connection, 41 various quantitative and qualitative features were extracted [12]. The 41 features extracted fall into three categorties, "intrinsic" features that describe about the individual TCP/IP connections; can be obtained form network

audit trails, "content-based" features that describe about payload of the network packet; can be obtained from the data portion of the network packet, "traffic-based" features, that are computed using a specific window (connection time or no of connections). As DOS and Probe attacks involve several connections in a short time frame, whereas R2U and U2Su attacks are embedded in the data portions of the connection and often involve just a single connection; "traffic-based" features play an important role in deciding whether a particular network activity is engaged in probing or not. Attack types fall into four main categories:

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

III. BIASED SUPPORT VECTOR MACHINES

Biased support vector machine (BSVM), a decomposition method for support vector machines (SVM) for large classification problems [7,8]. BSVM uses a decomposition method to solve a bound-constrained SVM formulation. BSVM Uses a simple working set selection which leads to faster convergences for difficult cases and a bounded SVM formulation and a projected gradient optimization solver which allow BSVM to quickly and stably identify support vectors. Leave-one-out model selection for biased support vector machines (BSVM) is used for automatic model selection [9].

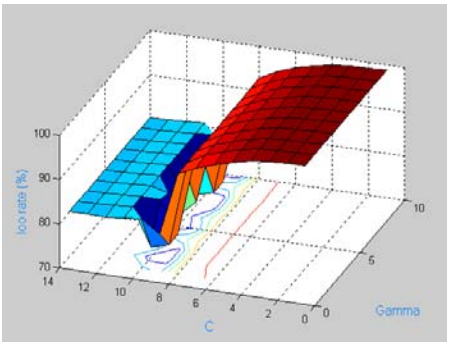

Fig.1. BSVM model for Normal

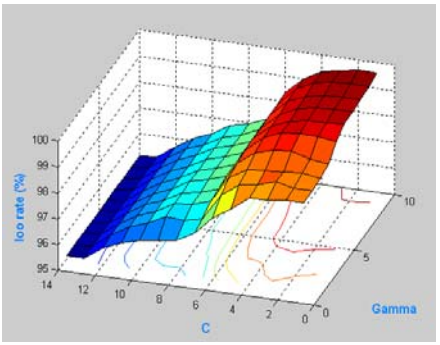

Fig.2. BSVM model for Probe

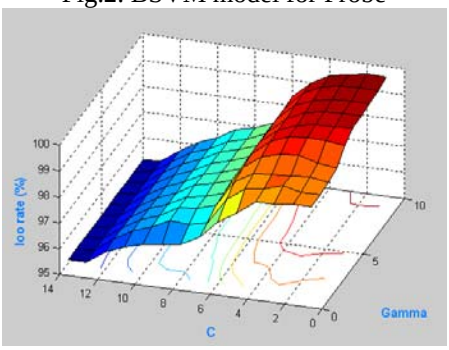

Fig.3. BSVM model for DoS

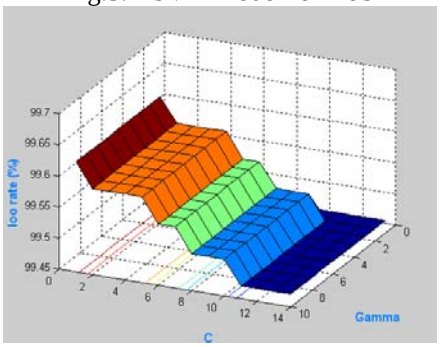

Fig. 4. BSVM model for U2Su

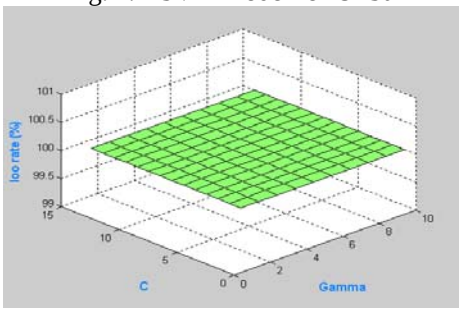

Fig.5. BSVM model for R2L

Models generated for intrusion detection data using leave-one-out model for support vector machines (looms) are given in figures 1 to 5.

IV. MODEL SELECTION SVMs

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 [13,14].

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 [15].

We performed a grid search using 10-fold cross validation for each of the five faults in our data set. First, we achieved the search of parameters C and γ in a coarse scale and then we carried through a fine tuning into the five detection faults proper space. Model selection results obtained through grid search are given in figures 6 to 10 for normal, probe, DoS, U2Su, and R2L, respectively.

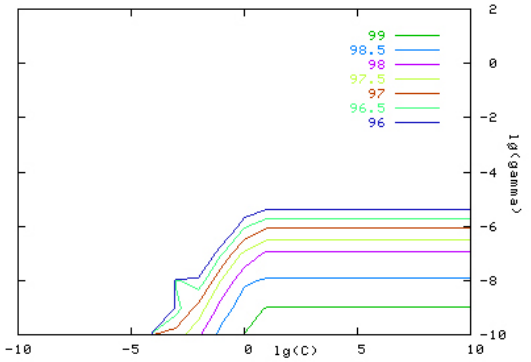

Fig.6. SVM model for Normal

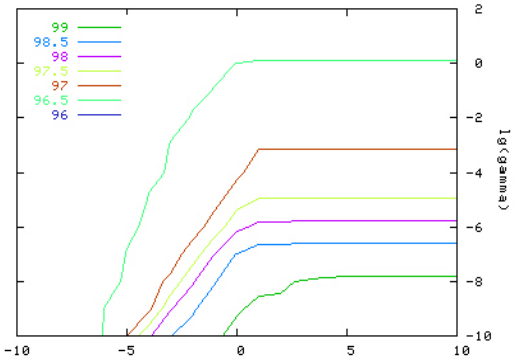

Fig.7. SVM model for Probe

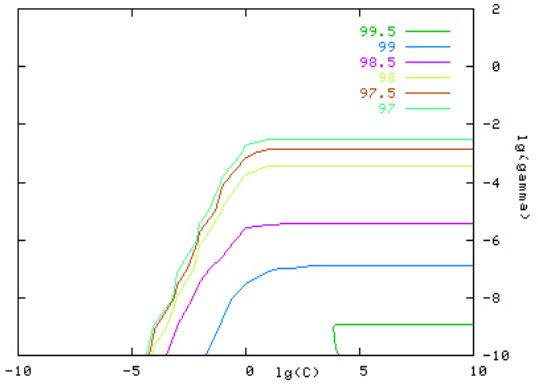

Fig.8. SVM model for DoS

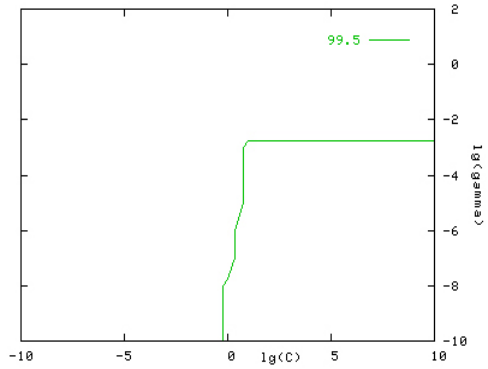

Fig.9. SVM model for U2Su

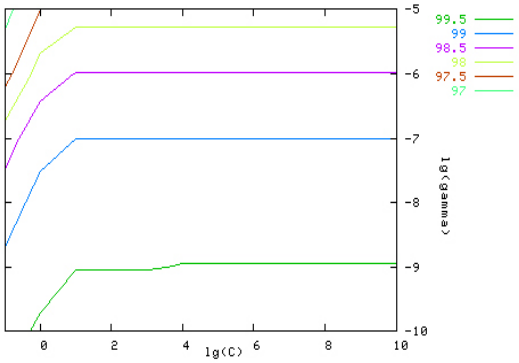

Fig.10. SVM model for R2L

V. 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 11 to 15.

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).

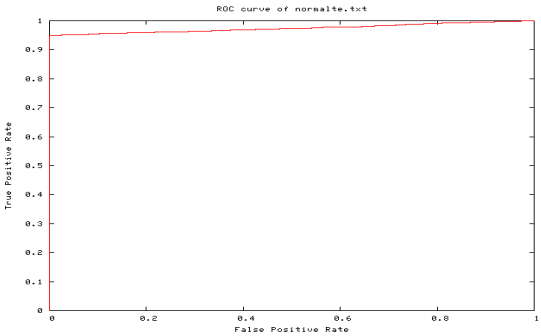

Fig.11. SVM accuracy for normal

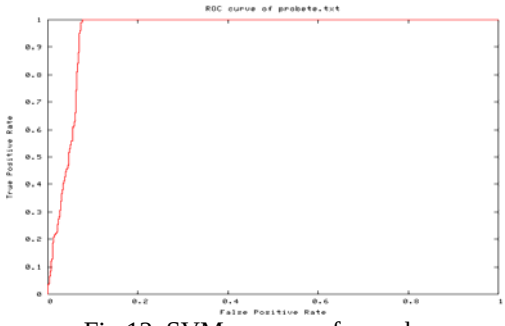

Fig.12. SVM accuracy for probe

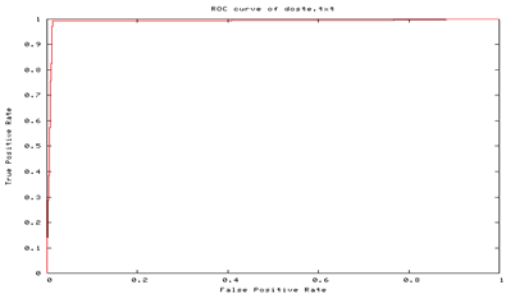

Fig.13. SVM accuracy for DoS

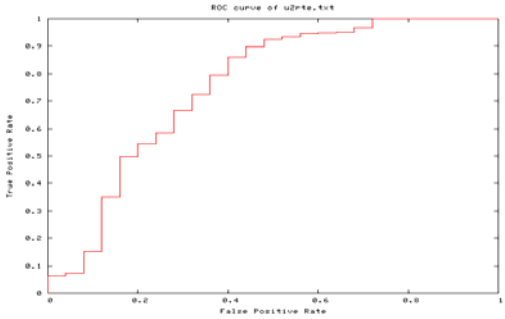

Fig.14. SVM accuracy for U2Su

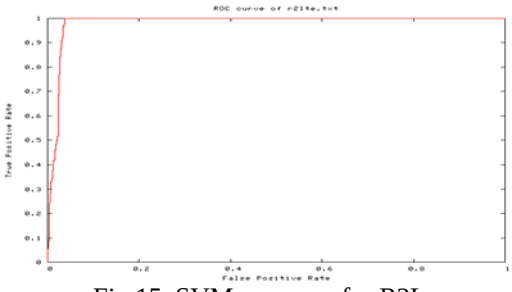

Fig.15. SVM accuracy for R2L

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 [16].

VI. RESULTS

In our experiments, we perform 5-class classification using different kernel methods [17]. The (training and testing) data set contains 11982 randomly generated points from the data set representing the five classes, with the number of data from each class proportional to its size, except that the smallest class is completely included. The set of 5092 training data and 6890 testing data are divided in to five classes: normal, probe, denial of service attacks, user to super user and remote to local attacks. Where the attack is a collection of 22 different types of instances that belong to the four classes described in section 3, and the other is the normal, and the other is the normal data. The normal data belongs to class1, probe belongs to class 2, denial of service belongs to class 3, user to super user belongs to class 4, remote to local belongs to class 5. Note two randomly generated separate data sets of sizes 5092 and 6890 are used for training and testing different implementations of support vector machines. Same training and test datasets were used for all the experiments. Table 1 summarizes the overall classification accuracy of SVMs, BSVMs and Looms (BSVMs).

Table 1. Classification accuracies of different kernel methods

Class	SVM	BSVM	Looms (BSVM)
Normal	**98.42**	98.35	95.43
Probe	98.57	99.46	**99.65**
DoS	99.11	**99.33**	95.37
U2R	**99.87**	99.58	99.65
R2L	97.33	99.33	**100.00**

VII. DISCUSSION & CONCLUSIONS

SVMs easily achieve high detection accuracy (higher than 95%) for each of the 5 classes of DARPA data. SVM performs the best for detecting Normal and User to Super User, BSVM performs the best for Denial of service attacks and Leave-one-out model selection for support vector machines (looms) based

on BSVM performs the best for Probe and Remote to Local.

Model selection results using Leave-one-out model selection for support vector machines (looms) based on BSVM are presented in (Figures 1-5). A grid search for intrusion detection using SVM (Figures 6 to 10) 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.

ACKNOWLEDGEMENTS

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 fourth author was performed during a visit to New Mexico Tech.

REFERENCES

1. S. Mukkamala, G. Janoski, A. H. Sung, "Intrusion Detection Using Neural Networks and Support Vector Machines," *Proceedings of IEEE International Joint Conference on Neural Networks*, pp.1702-1707. 2002.
2. M. Fugate, J. R. Gattiker, "Computer Intrusion Detection with Classification and Anomaly Detection Using SVMs," *International Journal of Pattern Recognition and Artificial Intelligence* 17(3), pp. 441-458 2003.
3. W. Hu, Y. Liao, V. R. Vemuri, "Robust Support Vector Machines for Anomaly Detection in Computer Security," *International Conference on Machine Learning*, pp. 168-174. 2003.
4. K. A. Heller, K. M. Svore, A. D. Keromytis, S. Stolfo, "One Class Support Vector Machines for Detecting Anomalous Window Registry Accesses," In *3rd IEEE Conference Data Mining Workshop on Data Mining for Computer Security*. 2003.
5. A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava, V. Kumar, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," *In Third SIAM Conference on Data Mining*. 2003.
6. S. Mukkamala, A. H. Sung, B. Ribeiro, "Model Selection for Kernel Based Intrusion Detection Systems," *International Conference on Adaptive and Natural Computing Algorithms* (ICANNGA05), pp. 458-461. 2005.
7. C. W. Hsu, C. J. Lin, "A comparison on methods for multi-class support vector machines," *IEEE Transactions on Neural Networks*, 13, pp. 415-425, 2002.
8. C. H. Chan, I. King, "Using Biased Support Vector Machine to Improve Retrieval Result in Image Retrieval with Self-organizing Map," *Proceedings of 11th International Conference, ICONIP. Lecture Notes in Computer Science 3316 Springer*, ISBN 3-540-23931-6, pp. 714-719, 2004.
9. J. H. Lee, C. J. Lin, "Automatic model selection for support vector machines," *Technical report*, Department of Computer Science and Information Engineering, National Taiwan University, 2000.
10. K. Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems", *Master's Thesis, Massachusetts Institute of Technology*, 1998.
11. S. E. Webster, "The Development and Analysis of Intrusion Detection Algorithms," *S.M. Thesis, Massachusetts Institute of Technology*, 1998.
12. W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," *Proc. 1998 7th USENIX Security Symposium*, 1998.
13. O. Chapelle, V. Vapnik, "Model selection for support vector machines," *Advances in Neural Information Processing Systems 12*, 1999.
14. V. Cherkassy, "Model complexity control and statistical learning theory," Journal of natural computing 1, pp. 109–133, 2002.
15. N. Cristianini, J. S. Taylor, "Support Vector Machines and Other Kernel-based Learning Algorithms," *Technical Report*, Cambridge University Press, 2000.
16. J. P. Egan, "Signal detection theory and ROC analysis," New York: Academic Press, 1975.
17. C. C. Chang, C. J. Lin, "LIBSVM: a library for support vector machines," *Technical Report*, Department of Computer Science and Information Engineering, National Taiwan University, 2001.