A Three-tier IDS via Data Mining Approach [*]

Tsong Song Hwang
Department of Computer Science & Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
M9315036@mail.ntust.edu.tw

Tsung-Ju Lee
Taiwan Information Security Center
National Taiwan University of Science and Technology
Taipei, Taiwan
tjlee.cs94g@nctu.edu.tw

Yuh-Jye Lee
Department of Computer Science & Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
yuh-jye@mail.ntust.edu.tw

ABSTRACT

We introduced a three-tier architecture of intrusion detection system which consists of a blacklist, a whitelist and a multi-class support vector machine classifier. The first tier is the blacklist that will filter out the known attacks from the traffic and the whitelist identifies the normal traffics. The rest traffics, the anomalies detected by the whitelist, were then be classified by a multi-class SVM classifier into four categories: PROBE, DoS, R2L and U2R. Many data mining and machine learning techniques were applied here. We design this three-tier IDS based on the KDD'99 benchmark dataset. Our system has 94.71% intrusion detection rate and 93.52% diagnosis rate. The average cost for each connection is 0.1781. All of these results are better than those of KDD'99 winner's. Our three-tier architecture design also provides the flexibility for the practical usage. The network system administrator can add the new patterns into the blacklist and allows to do fine tuning of the whitelist according to the environment of their network system and security policy.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning—*Concept Learning, Parameter learning*

General Terms

Security

Keywords

intrusion detection system, blacklist, whitelist, KDD'99, multiclass SVMs, RIPPER, activity profile, false alarm rate

1. INTRODUCTION

The tremendous growth of the advanced computer networks and network-based computer systems makes our computers exposed to the threats from our enemies and criminals. In order to assure the integrity of computer systems, more and more defense techniques are being brought out such as firewalls, anti-virus software, intrusion detection systems, etc. Intrusion detection system is a novel defense technique which can determine if or not a computer network or server has experienced or is experiencing an unauthorized intrusion. We employ data mining and machine learning techniques to build up a three-tier intrusion detection system that can detect the internet attacks and categorize the attacks into four categories: PROBE, DoS, R2L and U2R. The front tier of this IDS is a blacklist that is a set of association rules mining from the known attack connections from the training dataset. The blacklist here plays a role of misuse detector (MD) and will filter out most of the known attacks. It also classifies what kind of attack the connection belongs to. We use RIPPER [3] as our base rule-learning method, and further apply training/testing concept to prune redundant rules to achieve better performances. The whitelist is generated by some statistical descriptions based on the normal activity profiles which play a role of anomaly detector (AD). This tier will help us to detect novel attacks. The traffic rejected by AD is considered as an abnormal connection. We still have to predict the type of the attack. The third tier is a multi-class support vector machine classifier will complete this task. We design and trained this three-tier IDS based on the KDD'99 benchmark dataset. Taking the advantages of misuse detector and anomaly detector and superiority of SVMs in classification problems simultaneously, we can achieve 94.71% intrusion detection rate and 93.52% diagnosis rate. The average cost for each connection is 0.1781. All of these results are better than those of KDD'99 winner's. Our three-tier architecture design also provides the flexibility for the practical usage. The network system administrator can add the new patterns into the blacklist and allows to do fine tuning of the whitelist according to the environment of their network system and security policy.

[*] This work was supported by the iCAST project sponsored by the National Science Council, Taiwan, under the Grants No. NSC95-3114-P-001-002-Y02 and partially supported by ACM.

MineNet'07, June 12, 2007, San Diego, California, USA.
Copyright 2007 ACM 918-1-59593-792-6/07/0006 ...$5.00.

We briefly outline the contents of this paper. Section 2 provides data mining methods, including the way to build blacklist, whitelist and a very short introduction of support vector machines. The architecture of the three-tier IDS will be shown in this section as well. Section 3 gives the characteristics of KDD'99 benchmark dataset that is used in built up our three-tier IDS. We also described all experiment details in Section 3 and experimental results in Section 4. We conclude this paper in Section 5.

2. METHODOLOGIES

In the traditional cyberspace, security experts usually identify attacking patterns by their signatures. These signature-based concepts can be applied to recognize known attacks with high accuracy. However, these signature-based methods (blacklist) are weak in detecting novel attacks while security experts can easily apply their domain knowledge to maintain systems. Recently more and more researchers devote to the developments of anomaly detectors (whitelist). In empirical studies, AD has the ability of detecting novel attacks but may miss some known attacks. These two kinds of methods both have its strength and weakness. To take advantages of both methods tends to the trend of cyberspace.

Our approach is based on the ensemble of blacklist/whitelist, thus we build a blacklist at the first tier and a whitelist at the second tier. Then we use one against one multiclass SSVMs [12] classification method at the third tier to classify those anomalies detected by whitelist into the four attack categories. The architecture of the three-tier IDS is shown in Figure 1. The first method we want to present is RIPPER rule learning method, which is used to build the blacklist from the old attacks in training dataset. Next, we will describe the concept of activity profiling, which is used to build the whitelist from the normal connections in training dataset. The final method to be introduced is SVM classification algorithm.

2.1 RIPPER

RIPPER [3] is a propositional rule learning method which learns a set of if-then rules from data. RIPPER means “Repeated Incremental Pruning to Produce Error Reduction”. RIPPER is an optimized version of IREP [7], which is a rule learning algorithm proposed by Furnkranz and Widmer.

IREP combines separate-and-conquer rule learning and reduced error pruning (REP). Separate-and-conquer is a sequential-covering rule learning algorithm. First, the training data is divided into a growing set and a pruning set. Then this algorithm generates a rule set in a greedy fashion, a rule at a time and removes all examples covered by the new rule as soon as the rule is constructed. REP is an effective technique used to prune the tree learned in decision tree learning system and can be easily adapted to rule learning system.

While generating a rule RIPPER searches the most valuable rule for the current growing set in rule space. Immediately after a rule is extracted on growing set, it is pruned on pruning set. After pruning, the corresponding examples covered by that rule in the training set (growing and pruning sets) are deleted. The remaining training data is re-partitioned after each rule is learned in order to help stabilize any problems caused by a “bad-split”. This process is repeated until the terminal conditions satisfy.

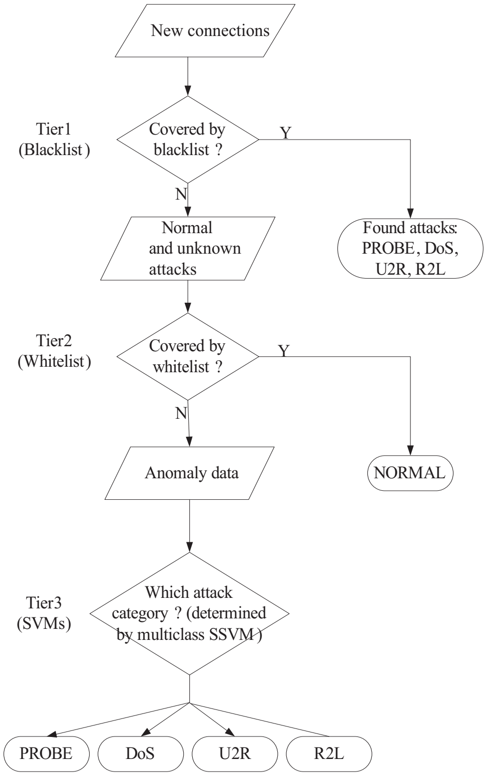

Figure 1: Architecture of the Three-tier IDS

The learned rules are easy to understand and to deploy on IDS. Each rule produced by RIPPER contains a conjunction of conditions, i.e., attribute-value tests, and a consequence, i.e., a class label. Each condition can be one of the form $A_n = v$, $A_c <= t$, or $A_c >= t$, where A_n is nominal attribute and v is a legal value for A_n, A_c is a continuous variable and t is some value for A_c. For an example whose attribute value satisfies the condition of the rule, it will be assigned to positive class, otherwise, negative class.

2.2 Activity Profiling

Network activity profiling [15] means collecting statistics that give a summary of the kinds of activities that are normally taking place on the network. We can thus obtain the pictures of the normal traffic on the network, e.g., histograms. They are compared to find similar behavior or abnormal behavior within the same activity [18]. Figures 2 and 3 are examples of feature distribution of some types of normal activities. For an abnormal activity, usually it does not behave in the same way as normal one. Thus we can find out the differences or deviations between them.

We can categorize all collected normal data into several activities according to the services provided on the network system and the protocols used. However, the more features we use to identify attacks, the higher detect rate can we achieve. In the meanwhile, we need to suffer more false alarms. The network administrators can adjust the sensitivities of their systems on the degrees of their sufferance.

2.3 Support Vector Machine Classifiers

In recent years support vector machines (SVMs) with linear or nonlinear kernels [1, 4, 19] have become one of the most promising supervised learning algorithms. For the bi-

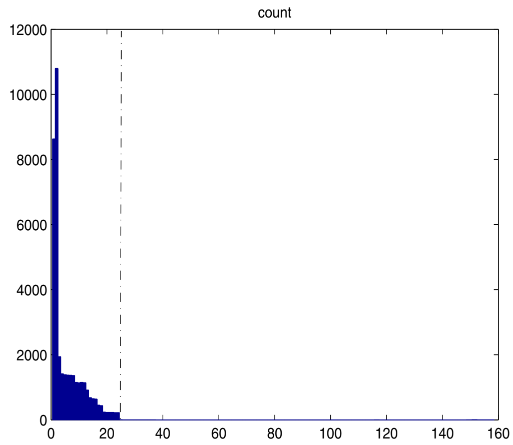

Figure 2: Histogram of Feature "count" of Activity "tcp ftp_data" , Border of the Low Frequency 1% Region is Shown in a Vertical Dashed Line

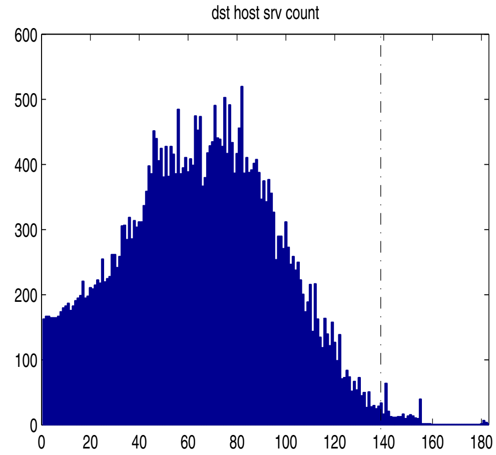

Figure 3: Histogram of Feature "dst_host_srv_count" of Activity "tcp ftp_data", Border of the Low Frequency 1% Region is Shown in a Vertical Dashed Line

nary classification problems, SVMs are able to construct a nonlinear separating surface (if it is necessary), which is implicitly defined by a kernel function [19]. For multi-class classification problems one of most popular strategy is decomposing the multi-class classification problem into a series of binary classification problems. Many variants of SVMs have been proposed [6, 16, 2].In this paper, we treated categorizing anomaly traffic detected by the whitelist into four classes: PROBE, DoS, R2L and U2R as a multi-class classification problem. We use *one vs. one* criteria [10] to decompose this multi-class classification into six binary classification problems and employ the smooth support vector machine (SSVM) [12] as the classification problem solver. The smooth support vector machine is one kind of SVM variants. It utilizes the optimality conditions and smooth techniques to convert the conventional SVM into an unconstrained convex minimization problem that can be solved very efficiently by a fast Newton-Armijo algorithm that converges globally and quadratically. Because this dataset is highly complicated we have to introduce the Gaussian (RBF) kernel into SVM classifiers. For more details, please consult [12].

3. THREE-TIER IDS CONSTRUCTION AND EXPERIMENTS

In this section we will give the details about the experiments on KDD'99 dataset [9]. First, we will describe the characters of KDD'99 dataset and then the design and conditions of the experiments will be given. We will show the results and discussion in next section.

3.1 KDD'99 Benchmark Dataset

DARPA dataset is the standard benchmark of IDS. The Lincoln Laboratory at MIT, under the Defence Advanced Research Projects Agency (DARPA) sponsorship, conducted the 1998 and 1999 comparative evaluation of intrusion detection systems [13, 14]. The goal of 1998 and 1999 DARPA Off-line Intrusion Detection Evaluation is to promote the research in developing better algorithm of high performance to detect the intrusions in network system.

KDD'99 dataset: KDD Cup 1999 contest provides a training dataset of 5 million instances of connection data and a test dataset containing various types of attacks, some of them are new to the training dataset. Each instance represents a connection on a network. There are 41 features used to represent a connection. Features 1-9 stand for the basic features of a packet, 10-22 for content features, 23-31 for traffic features and 32-41 for host based features [11]. Because of the large amount of data in training dataset, this may be an obstacle to most of the machine learning methods. Thus a smaller version 10% training dataset is also provided. The dataset provided by KDD'99 contest is in fact the same as the dataset used in DARPA 1998 evaluation except that the data unit is different: one is based on connection and the other is based on packet. In the training dataset, there are 19.86% normal connections and 80.14% attacks. While in the testing dataset, there are 19.48% normal connections, 74.50% old attacks which have been shown in the training set and 6.02% new attacks which have not been shown in training set.

3.2 Three-tier IDS

In order to design a combined detector of misuse detector and anomaly detector, or blacklist/whitelist, we design a three-tier process to include both misuse detector and anomaly detector.

First, we use the rule learning method RIPPER to learn the patterns of attacks given in training dataset and use these patterns to find corresponding specific attacks in test dataset. Next, we use the activity profiles of normal examples to build an anomaly detector to find the attacks, which are not found by misuse detector, in test dataset. Finally, we classify the detected anomaly examples into the four attack categories.

3.2.1 *Misuse Detector (Blacklist)*

To find patterns of various attacks, we use rule learning method RIPPER to learn a set of rules from training dataset to represent specific behavior of attacks. The program we used is JRip, a JAVA version of RIPPER, in Weka [5] version 3.4.5. Because of the practical difficulty of directly learning rules from 5 million instances, we used the smaller 10% training dataset, 494,021 instances, to train our rule learning system. Since there are many redundant samples in the training dataset, which contributes little to the description of attack pattern. We can delete these redundant instances and use the unique ones to learn the attack patterns. After deleting redundant data, we obtained the dataset used for training, 145,586 distinct instances.

Because we want to learn the specific patterns of each kind of attack, we performed a series of one-against-rest binary classification for them. The so-called one-against-rest binary classification is to assume the target attack type as positive class and all the other types, including normal types, as negative class, then to perform a binary classification on the two classes of instances. In order to validate the learned rules obtained from 10% training dataset, we test them on the entire training dataset. Only the rules which have high performance of better than 99.5% accuracy are selected.

3.2.2 Anomaly Detector (Whitelist)

The anomaly detector, or whitelist(WL), is designed based on profiles of the normal activities. The feature statistics of a normal activity we used is the range of feature value, minimum and maximum. The detail is stated below.

First, we utilize the rest of normal instances in the 5 million training dataset, filtered by first tier of IDS, to find various normal activities. In fact, only few normal instances are misclassified as attacks. These activities are described using two independent dimensions: protocol_type and service, two nominal features among the 41 given ones. The two features respectively have 3 and 66 legal components. Thus, in theory there should be totally 198 different combinations of them. In reality, We found only 27 different activities in normal instances of 5M training dataset. Then we can use profiles of these 27 activities to build an anomaly detector.

For each of the 27 activities, we recorded the range of value of each of the 41 features except protocol_type and service, i.e. 39 ranges of feature value are recorded. Among the 39 features, 7 are nominal and 32 are numerical. For the 7 nominal features, only feature flag contains 11 legal components while the other 6 are binary. We will treat the six binary features in the same way as numerical ones. For each normal activity, we will record the set of occurring components of feature flag in the training dataset.

Then we will state how to find out what kind of incoming connection is an attack . For an connection to be normal, it has to satisfy two conditions.

- It has to be one of the 27 normal activities.
- If it satisfies the previous condition, then each of its 39 feature values must locate in their corresponding range of normal activity found earlier.

From the viewpoint of probability, the first condition is reasonable because any new activity must be rare and we can not trust it as being normal from the existing data. Maybe the second condition is not reasonable enough for several reasons. Not all of these 39 features are independent. Only some of them are independent while others are derived from these independent ones. But before we find the minimum set of required features to describe a normal activity, we still have to check all of them. On the other hand, the range of feature value is defined by minimum and maximum value of feature. Some rarely occurring normal data with very large or very small feature value could lead to an unreasonable range, as shown in Figures 2 and 3. Thus, we can use the region which has high frequency of data, 99% here, to represent a more reasonable range of feature value.

We use all the attacks in the entire training dataset to validate our anomaly detectors, implemented in a Perl script, and found that the detection rate is higher than 99.96%. On the other hand, for those attacks in the entire training dataset not detected by the blacklist, totally 1577 attacks, whitelist can detect 1058 attacks. This means our anomaly detector can really detect those attacks not found by the blacklist, or they can detect new attacks. Hence, we adopt the this anomaly detector to detect intrusions in test dataset.

3.2.3 SVM Classifiers

To classify the detected attack by anomaly detector into the four attack categories, we use multiclass, one against one, smooth support vector machine(SSVM) as our classifier training method. The rest of attacks in entire dataset not filtered by the blacklist are used to train the SSVM classifiers.

The 41 features given in KDD'99 dataset contain both numerical and nominal features. Because SVM can not process nominal features, we transform all nominal features to a set of binary features. Thus, we used totally 118 features to perform SVM calculation.

The training parameters c and γ are automatically determined by the method, developed in Data Mining lab, NTUST [8], implemented in MATLAB. The six combinations of binary classification are calculated using the same c and γ. Then we use majority vote to determine which category of attack the test example belongs to.

4. EXPERIMENTAL RESULTS

The results of our designed three-tier IDS on KDD'99 test dataset will be described tier by tier. For all confusion tables here, column stands for class prediction and row for class membership. First, the result of applying blacklist, is shown in Table 1.

	NORM.	PROBE	DoS	U2R	R2L
NORM.	0	211	76	0	0
PROBE	0	2998	149	0	0
DoS	0	5	223898	0	0
U2R	0	0	0	2	5
R2L	0	0	0	0	418

Table 1: Confusion Table of the Result of Tier 1

The number of total instances found by blacklist is 227,762, which accounts for 73.23% of test dataset. Among these instances, only 0.20% are wrongly predicted. The result shows that our blacklist highly correctly learns the patterns of various attacks.

The rest of test dataset after tier 1 has 83,267 instances which contains normal instances and the rest of attacks, known and unknown, not filtered by the blacklist. The confusion table of the results of tier 2 is shown in Tables 2.

	NORM.	ABNORM.
NORM.	58290	2016
ABNORM.	14152	8809

Table 2: Confusion Table of the Result of Tier 2

The detection accuracies of the the anomaly detector, whitelist, is 80.58%. Thus, the concept of 1% low frequency region really works for improving detection accuracy.

Now we turn to the result of tier 3, which is shown in Table 3. For the true NORMAL instances, they are predicted to be mostly PROBE attacks. This is good because it is in accordance with the fact that the behavior of PROBE attack is similar to that of NORMAL activity. For the true DoS and U2R attacks, the prediction seems not good. The possible reason is that the number of training examples for our SVM classifier is only 1,577, and the number of instances to be predicted is 11,633. On the other hand, there are many unknown attacks to be predicted, while none of them is in the training instances.

	NORM.	PROBE	DoS	U2R	R2L
NORM.	0	1992	5	4	15
PROBE	0	871	36	1	76
DoS	0	46	164	0	334
U2R	0	155	0	6	6
R2L	0	2794	2	98	4220

Table 3: Confusion Table of the Result of Tier 3

	NORM.	ABNORM.
NORM.	(58290,60262)	(2303,331)
ABNORM.	(14152,20505)	(236284,229931)

Table 4: Confusion Table of the Results from the Viewpoint of Intrusion Detection (three-tier, KDD'99 winner)

The confusion tables for the three tiers are shown in Tables 4 and 5. The result of KDD'99 winner [17] is provided to compare with that of ours. The first number in the parentheses stands for our result and the second one stands for KDD'99 winner's. From the viewpoint of intrusion detection, our prediction accuracy 94.71% is a little higher than KDD'99 winner's 93.3%.

The above analysis means that our method can find more abnormal connections under a reasonable false alarm rate than those found by KDD'99 winner. On the other hand, from the viewpoint of intrusion diagnosis, our total prediction accuracy is also higher than KDD'99 winner's.(ours: 93.52%, KDD'99 winner's: 92.71%).

Given the cost matrix of KDD'99, see Table 6, our average cost per test example is better than that of KDD'99 winner.(ours:0.1781, KDD'99 winner's: 0.2331) Compare to KDD'99 winner's result, ours is better in every perspective except false alarm rate (KDD'99 winner's: 0.55%, ours:3.80%).

Now we turn to the subject of detecting new attacks. Detection here means that an attack is detected to be an anomaly, no matter which attack class it is classified as. The overall result of the % of detection of old and new attacks in various classes of attack is shown in Table 7. The old attacks are mostly correctly detected except U2R class. As for new attacks, both DoS and R2L classes can be detected only up to 27%. But for PROBE and U2R, we can obtain high performance for new attacks. The above analysis means that our AD needs to be modified to detect new attacks. The possible direction to modify it is to find out why it cannot detect Mailbomb and Snmpgetattack.

	NORM.	PROBE	DoS	U2R	R2L
NORM.	0	1	2	2	2
PROBE	1	0	2	2	2
DoS	2	1	0	2	2
U2R	3	2	2	0	2
R2L	4	2	2	2	0

Table 6: Cost Matrix of KDD'99 contest

CLASS	OLD	NEW	TOTAL
PROBE	99.92%	98.16%	99.16%
DOS	99.99%	18.03%	97.65%
U2R	20.51%	87.83%	76.32%
R2L	79.84%	26.94%	46.53%

Table 7: % of Detection of Old and New Attacks in Various Classes (MD and AD)

Finally, as shown in Table 8, 97.54% of old types of attack are detected by our MD and 1.91% by AD. Only 7.84% of new types of attack are detected by MD and 23.40% by AD. The result shows that old types of attack are nearly completely detected by the combination of MD and AD (99.45%). AD is more suitable to detect new attacks than MD. But MD is still useful because it can detect old attacks efficiently and precisely.

5. CONCLUSIONS

In this paper, we proposed a three-tier IDS design which consists of a blacklist, a white and a multi-class SVM classifier. This design will take the merits of MD and AD for intrusion detection purpose. The last tier, SVM classifier, will categorize the attack into four classes: PROBE, DoS, R2L and U2R. The KDD'99 benchmark dataset that generated from 1998 DARPA dataset for the KDD Cup 1999 contest was used here for training and evaluating our three-tier IDS performance.

The first tier IDS, or blacklist, can detect up to 97.54% of old attacks and 7.84% of new attacks. It can detect nearly completely all the old attacks and part of the new attacks. The second tier IDS, or whitelist, can detect 1.91% of old attacks and 23.40% of new attacks. The new attacks detected by the whitelist, or AD, are nearly three times of those detected by the whitelist. This shows that the designed whitelist can really detect new attacks and is complementary to the blacklist. The third tier IDS, SSVM classifiers, can categorize these detected anomalies into their corresponding class correctly to an acceptable level. The detection performance of our three-tier IDS is up to 94.71% and the false alarm rate is only 3.8%. All our results are better than those of KDD'99 winner's. Please note that the KDD' 99 benchmark dataset is for competition of data mining. The portion of abnormal connection is bigger than normal part. This is not the case of really network traffic. However, this framework still can be applied to the real network traffic. From the view point of a network administrator, the first

	NORM.	PROBE	DoS	U2R	R2L
NORM.	(58290,60262)	(2203,243)	(81,78)	(4,4)	(15,6)
PROBE	(35,511)	(3869,3471)	(185,184)	(1,0)	(76,0)
DoS	(5406,5299)	(51,1328)	(224062,223226)	(0,0)	(334,0)
U2R	(54,168)	(155,20)	(0,0)	(8,30)	(11,10)
R2L	(8657,14527)	(2794,294)	(2,0)	(98,8)	(4638,1360)

Table 5: Confusion Tables and Class Accuracies of the Results from the Viewpoint of Intrusion Diagnosis (three-tier,KDD'99 winner)

DETECTOR	OLD	NEW
MD	97.54%	7.84%
AD	1.91%	23.40%

Table 8: % of Detection of Old and New Attacks by MD and AD

tier IDS, blacklist, performs flexibly, allowing network administrator to add new patterns to the blacklist, and the whitelist allows the network administrator to do fine tuning according to the environment of their network system and security policy.

6. REFERENCES

[1] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] W. W. Cohen. Fast effective rule induction. In *ICML*, pages 115–123, 1995.

[4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, Cambridge, 2000.

[5] E. Frank, M. A. Hall, G. Holmes, R. Kirkby, and B. Pfahringer. WEKA - A machine learning workbench for data mining. In O. Maimon and L. Rokach, editors, *The Data Mining and Knowledge Discovery Handbook*, pages 1305–1314. Springer, 2005.

[6] G. Fung and O. Mangasarian. Proximal support vector machine classifiers. In F. Provost and R. Srikant, editors, *Proceedings KDD-2001: Knowledge Discovery and Data Mining, Auguest 26-29, 2001, San Francisco, CA*, pages 77–86, New York, 2001. Asscociation for Computing Machinery. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/01-02.ps.

[7] J. Fürnkranz and G. Widmer. Incremental reduced error pruning. In *ICML*, pages 70–77, 1994.

[8] C.-M. Huang, Y.-J. Lee, D. K. J. Lin, and S.-Y. Huang. Model selection for support vector machines via uniform design. *The special issue on Machine Learning and Robust Data Mining of Computational Statistics and Data Analysis*, 2006.

[9] University of california department of information and computer science. kdd cup 99 intrusion detection dataset task description, 1999. http://kdd.ics.uci.edu//databases/kddcup99/kddcup99.html.

[10] U. Kreβel. Pairwise classification and support vector machines. In B. Schölkopf, C.J.C. Burges and A. Smola, editors, *Advances in kernel methods: support vector learning*, pages 255–268, Cambridge, MA, USA, 1999. MIT Press.

[11] W. Lee. *A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems.* PhD thesis, Columbia University, 1999.

[12] Y.-J. Lee and O. L. Mangasarian. SSVM: A smooth support vector machine. *Computational Optimization and Applications*, 20:5–22, 2001. Data Mining Institute, University of Wisconsin, Technical Report 99-03. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-03.ps.

[13] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. In *Proceedings of the DARPA Information Survivability Conference and Exposition*, Los Alamitos, CA, 2000. IEEE Computer Society Press.

[14] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das. The 1999 DARPA off-line intrusion detection evaluation. *Computer Networks*, 34(4):579–595, 2000.

[15] D. J. Marchette. *Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[16] D. R. Musicant. ASVM Software: active set support vector machine classification software, 2000. www.cs.wisc.edu/∼musicant/data/asvm/.

[17] B. Pfahringer. Winning the KDD99 classification cup: Bagged boosting. *SIGKDD Explorations*, 1(2):65–66, 2000.

[18] S. J. Stolfo, S. Hershkop, K. Wang, O. Nimeskern, and C.-W. Hu. A behavior-based approach to securing email systems. In *MMMACNS: International Workshop on Methods, Models and Architectures for Network Security, LNCS*, 2003.

[19] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.