Proc. 13th European Symposium on Artificial Neural Networks ESANN05
Burges, Belgium, April 2005, pp. 345-350

FUZZY ROC CURVES FOR THE 1 CLASS SVM: APPLICATION TO INTRUSION DETECTION

Paul F. Evangelista, Piero Bonnisone, Mark J. Embrechts
Department of Decision Sciences and Engineering Systems

Boleslaw K. Szymanski
Department of Computer Science

Rensselaer Polytechnic Institute
Troy, New York 12180

Abstract. A novel method for receiver operating characteristic (ROC) curve analysis and anomoly detection is proposed. The ROC curve provides a measure of effectiveness for binary classification problems, and this paper specifically addresses unbalanced, unsupervised, binary classification problems. Furthermore, this work explores techniques in fusing decision values from classifiers and using ROC curves to illustrate the effectiveness of the fusion techniques. In describing an unbalanced classification problem, we are addressing a problem that has a low occurrence of the positive class (generally less than 10%). Since the problem is unsupervised, the 1 class SVM is utilized. We discuss the curse of dimensionality experienced with the 1 class SVM, and to overcome this problem we create subspaces of our variables. For each subspace created, the 1 class SVM produces a decision value. The aggregation of the decision values occurs through the use of fuzzy logic, creating the fuzzy ROC curve. The primary source of data for this research is a host based computer intrusion detection dataset.

1 Introduction

The purpose of this paper is to illustrate synergistic combinations of multiple classifiers for the unbalanced, unsupervised binary classification problem. The data explored in this paper is commonly referred to as the Schonlau et. al. or SEA dataset, originally discussed in [5, 6, 7, 17, 16]. Although this is a host based computer intrusion detection dataset, the applications of this work extend beyond computer intrusion detection.

Combinations of multiple classifiers (CMC) is an active area of research today. Given the numerous classification techniques and vast problem area domain, research within this field seems only bounded by creativity and computational power. A particularly interesting problem that involves CMC is the unbalanced, unsupervised binary classification problem. Consider the following example that we will refer to as the ”airport security problem”. An airport security system exists in layers, all the way up to and including the aircraft while it is airborne. Each layer in this security system can be considered a classifier; the objective of each layer is to determine whether or not a bad guy, or intruder let us say, is attempting to breach security. How should these classifiers be arranged? How can the results of each classifier be combined to achieve synergistic results?

An unsupervised classifier is crippled by the fact that it cannot learn from true positive (intruders) examples. The only examples available to learn from are true negatives (non-intruders). Furthermore, the problem we have identified is unbalanced. This means that the frequency of intruders is very small; in an airport, this number of true positives is a fraction of a percent. Some airports work for years without experiencing an intruder. Yet the cost of not identifying a true positive can be catastrophic, and the cost of falsely identifying non-intruders, or having too many false positives, can create massive inefficiency.

Given a classification problem of high dimension (perhaps $>$10 variables), initial experimental results indicate that the creation of subspaces and aggregation of the subspace classification decision values results in improved classification over a model that utilizes all variables at once (the unsupervised classification model that will be utilized is the 1 class SVM [15]). This is often known as the ”curse of dimensionality”. Creating subspaces for outlier detection, which is essentially what we are describing, is not a new concept. However, considering this problem as a function of 1 class SVM outputs to create a “fuzzy ROC curve” has not been addresses in the literature. Furthermore, it is through careful analysis of receiver operating characteristic (ROC) curves that we will measure performance. As we combine multiple classifiers, we will seek to gain synergistic improvement in our ROC curves. Fuzzy logic is the basis for the aggregation. Each classifier will create a decision value that ranges from -1 to +1, where +1 should indicate the negative (non-intruder) class, and -1 indicates the positive (intruder) class. These decision values represent a degree of membership as an intruder or a non-intruder. The decision values must be combined to make a final decision, and fundamentals of fuzzy logic can be used to aggregate these decision values.

As mentioned earlier, unsupervised learning does not perform well in higher dimensions. A valid question would be to ask why not try a dimension reduction technique, such as principal components. Principal components work well with balanced classification problems, however caution is necessary with unbalanced problems. Often the difference between an intruder and non-intruder is subtle, and principle components can dilute the information content of the variables. Therefore, other techniques should be considered.

2 Recent Work

This paper explores many facets of recent research, to include intrusion detection models, outlier detection, and fusion of classifiers using fuzzy ROC curves. Recent work with classifier fusion and fuzzy ROC curves will be discussed during the presentation of the methods for that material.

Schonlau et. al. [5, 6, 7, 17, 16] conducted the original work with this data, hence the reference to the data as the Schonlau et. al. data, or SEA data. Their contributions included a thorough analysis of several statistical techniques for identifying masqueraders. Schonlau et. al. explored approaches that include: Bayes one-step Markov model, hybrid multistep Markov model, text compression, Incremental Probabilistic Action Modeling (IPAM), sequence matching, and a uniqueness algorithm[5]. Schonlau stressed the importance of minimizing false positives, setting a goal of 1% or less for all of his classification techqniques. Schonlau’s uniqueness algorithm, explained in [17], achieved a 40% true positive rating before crossing the 1% false positive boundary. Wang [19] used one-class training based on data representative of only one user and demonstrated that it worked as well as multi-class training. Coull [4] applied bioinformatics matching algorithm for a semi-global alignment to this problem. Lee [12] built a data mining framework for constructing features and model for intrusion detection. Evangelista et. al. [8] applied supervised learning through Kernel Partial Least Squares to the SEA dataset.

Roy Maxion contributed insightful work with this data that challenged both the design of the data set and previous techniques used on this data [14, 13]. Maxion uses a 1v49 approach in [14], where he trains a Naive Bayes Classifier one user at a time using the training data from one user as true negative examples versus data from the forty-nine other users (hence 1v49) as true positive (masquerader) examples. Maxion claimed the best performance to date in [14], achieving a true positive rating of 60% while maintaining a false positive rating of 1% or less. Maxion also examines masquerade detection with a similar data set that contain command arguments in [13].

The 1 Class SVM is an outlier detection technique originally proposed in [15]. Stolfo and Wang [18] successfully apply the 1 Class SVM to this dataset and compare it with several of the techniques mentioned above. Chen uses the 1 Class SVM for image retrieval[3]. The simplest way to express the 1 class SVM is to envision a sphere or ball, and the object is to squeeze all of the training data into the tightest ball feasible. This is analagous to the idea of variance reduction for distribution estimation; given a set of data, we want to estimate a distribution that tightly defines this data, and any other data that does not look the same will not fit this distribution. In other words, once the distribution is estimated, data that does not fit the distribution will be considered an outlier or not a member of the negative class. Consider the following formulation of the 1 Class SVM originally from [15] and also clearly explained in [3]:

If we consider $X_1, X_2, ..., X_l \in \chi$ instances of training observations, and Φ is a mapping into the feature space from $\chi \rightarrow F$.

$$\begin{aligned} & \min_{R\in\mathbb{R},\zeta\in\mathbb{R}^l,c\in F} && R^2 + \frac{1}{vl}\sum_i \zeta_i \\ & \text{subject to} && \parallel \Phi(X_i) - c \parallel^2 \leq R^2 + \zeta_i, \quad \zeta_i \geq 0 \text{ for } i \in [l] \end{aligned} \tag{1}$$

This minimization function attempts to squeeze R, which can be thought of as the radius of a ball, as small as possible in order to fit all of the training samples. If a training sample will not fit, $zeta_i$ is a slack variable to allow for this. A free parameter, v, enables the modeler to adjust the impact of the slack variables. The output, or decision value for a 1 Class SVM, takes on a values from -1 to +1, where values close to +1 indicate datapoints that fit into the ball and values of -1 indicate datapoints lying outside of the ball.

2.1 Curse of Dimensionality

It is largely understood that high dimensional data suffers from a curse of dimensionality. This curse of dimensionality involves the inability to distinguish distances between points because as dimensionality increases, every point tends to become equidistant as volume grows exponentially. This same curse of dimensionality occurs in the 1 class SVM. (The SVM tool used for this research is LIBSVM [2], and for the purpose of consistency we only use the linear kernel. We have experimented with the non-linear kernel options in LIBSVM, and the variance created by additional parameter tuning would distract from the fundamental message of this paper.)

Throughout these experiments we consistently Mahalanobis scale our data (subtract the mean and divide by the standard deviation). The dataset contains 5000 observations and a host of variables to measure these observations (see [8, **?**] for a description of variables), and we utilize 2500 observations for training and 2500 observations for testing. After eliminating all positive cases from the training data, 2391 negative cases remain which are used for training the 1 class SVM. In the testing data, there are 122 positive cases out of the 2500 observations.

Firgure 1 illustrates an experimental example of the curse of dimensionality where there are originally 27 meaningful variables, however meaningless probe variables (uniform (0,1) random variables) are added to create degradation. This area under the ROC curve, or AUC, will serve as a measure of classifier performance. Tom Fawcett provides an excellent discussion of ROC curves in [9] for the reader who is not familiar with ROC curves.

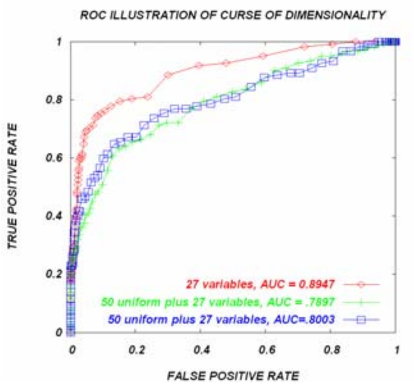

Figure 1. Curse of Dimensionality induced by the introduction of probe variables.

3 Method

We propose a technique to overcome this curse of dimensionality. The technique involves creating subspaces of the variables and aggregating the outputs of the 1 class SVM for each of these subspaces.

3.1 Subspace Modeling

Intelligent subspace modeling is an important first step. Orthogonal subspaces are desired, because we are interested in subspaces that measure different aspects of the data. The idea of creating diverse classifiers is not novel [1, 10, 11, 11], however in the literature the measures of classifier diversity involve functions of the classifier output. This is feasible with supervised learning, however in unsupervised learning this is more difficult because there are no true positive examples to measure diversity against. We propose measuring diversity through the actual data. Our method involves an analysis of the correlation between principal components of each subspace. This is by no means the only mesure for subspace diversity, however we have experienced good results with this model.

Given a Mahalanobis scaled (for each variable, subtract mean and divide by standard deviation) data matrix $\mathbf{X}$, containing m variables that measure n observations, create l mutually exclusive subspaces from the m variables. Assume ther are k variables in every subspace if m is divisible by l. Our experience with the 1 class SVM indicates that for $k > 7$ increased dimensionality begins to degrade performance, however this is simply a heuristic and may vary depending upon the unsupervised classifier selected. For each subspace, principal components can be calculated. We will refer to the matrix that contains the principal component loading vectors (eigenvectors) as $\mathbf{L}$. To determine correlation between principal components, calculate the principal component scores for each subspace where **S**=**XL**. Let π_i represent subspace i, and consider $\mathbf{S}_i$ as the score matrix for the π_i. Calculate the pairwise comparison for every column vector in $\mathbf{S}_i$ against every column vector in $\mathbf{S}_j$, $i \neq j$. This would be the equivalent of concatenating $\mathbf{S}_i$ for all i and calculating the correlation matrix, Σ.

We are interested in values approaching zero for every pairwise correlation across subspaces (principal components within subspaces are orthogonal and therefore their correlation is zero, as seen in Figure 3). However, there are a combinatoric number of subspace combinations to explore.

$$\text{Number of subspace combinations} = \binom{m}{k}\binom{m-k}{k}...\binom{2k}{k} \tag{2}$$

Equation 2 assumes that m is divisible by l, and even if this is not true the equation is almost identical and on the same order of magnitude. Our approach to search this subspace involved the implementation of a simple genetic algorithm, utilizing a chromosome with m distinct integer elements representing each variable. There are many possible objective functions that could pursue minimizing principal component correlation between subspaces, and we utilized the following letting $q \in (1, 2, ..., l)$:

$$\min \max_{\forall \pi_q} \mid \rho_{ij} \mid \quad \forall (i \neq j) \tag{3}$$

The fitness of each member is simply the maximum $\mid \rho_{ij} \mid$ value from the correlation matrix such that ρ_{ij} measures two principal components that are not in the same subspace.

3.2 Output Processing

After selecting the subspaces, prediction modeling begins. As mentioned previously, our choice for a prediction model is the 1 Class SVM with a linear kernel. However, the problem of classifier fusion still persists. Classifier fusion techniques have been discussed in [1, 10, 11, **?**], and the fusion techniques that we will present consider the aspects in these references with . Classifier fusion is a relatively new field and it is often criticized for lack of theoretical framework and too many heuristics [10]. We do not claim to provide a solution to this criticism. Our method of classifier fusion is a blend of techniques from fuzzy logic and classifier fusion, and although it may be considered another heuristic it is operational and should generalize to other security problems.

3.2.1 Mapping into Comparable Decision Spaces

For each observation within each subspace selected, the classifier will produce a decision value, d_{ij}, where d_{ij} represents the decision value from the j^{th} classifier for the i^{th} observation. Since the distribution of the output from almost any classification technique is questionable, we first consider a nonparametric measure for the decision value, a simple ranking. o_{ij} represents the ordinal position of d_{ij} (for the same classifier, meaning j remains constant). For example, if d_{71} is the smallest value for the 1^{st} classifier, $o_{71} = 1$. This nonparametric measure allows comparison of classifiers without considering the distribution. However, we do not rule out the distribution altogether. We also create p_{ij}, which is the Mahalanobis scaled (normalized) value for d_{ij}. In order to incorporate fuzzy logic, o_{ij} and p_{ij} must be mapped into a new space of real numbers, let us call Λ, where $\Lambda \in (0, 1)$.

This mapping will be $p_{ij} \rightarrow \delta_{ij}$ and $o_{ij} \rightarrow \theta_(ij)$ such that $\delta_{ij}, \theta_{ij} \in \Lambda$. For $o_{ij} \rightarrow \theta_{ij}$ this is a simple scaling procedure where all o_{ij} are divided by the number of observations, m, such that $\theta_{ij} = o_{ij}/m$. For $p_{ij} \rightarrow \theta_{ij}$, all p_{ij} values < -1 become -1, all p_{ij} values > 1 become 1, and from this point $\theta_{ij} = (p_{ij} + 1)/2$.

3.2.2 Fuzzy Logic and Decisions with Contention

There are now twice as many decision values for every observation as there were numbers of classifiers. Utilizing fuzzy logic theory, T-conorms and T-norms can be considered for fusion. The choice between T-norms and T-conorms depends upon the type of decision. The medical community is caution of false negative tests, meaning that they would rather have error on the side of falsely telling someone that they have cancer as opposed to letting it go undetected. The intrusion detection community is concerned about minimizing false positives, because too many false positives render an intrusion detection system useless as analysts slog through countless false alarms. In the realm of 1 Class SVMs, the original decision values will take on values ranging generally from -1 to +1, where values closer to +1 indicate observations that fit inside the ball or estimated distribution (indicating non-intruders), and values closer to -1 indicate outliers (potential intruders). Consider the max and min, simple examples of a respective T-conorm and T-norm. Systems that need to be cautious against false negatives will operate in the realm of the T-norms, creating a few more false alarms but missing fewer true positives. Systems that need to be cautious agains false negatives will operate in the realm of the T-conorms, perhaps missing a few true positives but generating fewer false positives. Figure 2 illustrates the domain of aggregation operators.

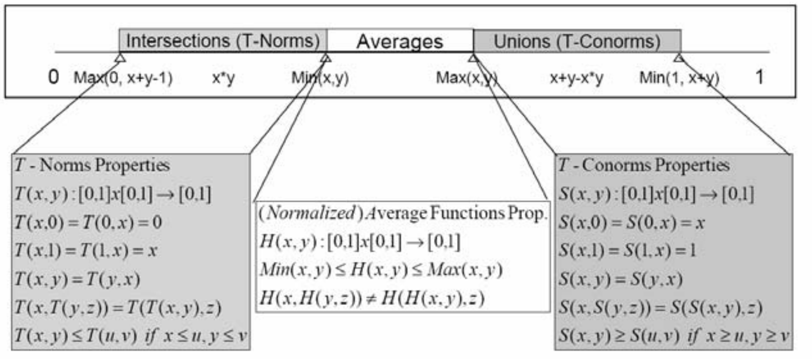
Intersections (T-Norms)
Averages
Unions (T-Conorms)
0
Max(0, x+y-1)
x*y
Min(x,y)
Max(x,y)
x+y-x*y
Min(1, x+y)
1
T - Norms Properties
T(x,y) : [0,1]x[0,1] → [0,1]
T(x,0) = T(0,x) = 0
T(x,1) = T(1,x) = x
T(x,y) = T(y,x)
T(x,T(y,z)) = T(T(x,y),z)
T(x,y) ≤ T(u,v) if x ≤ u, y ≤ v
(Normalized) Average Functions Prop.
H(x,y) : [0,1]x[0,1] → [0,1]
Min(x,y) ≤ H(x,y) ≤ Max(x,y)
H(x,H(y,z)) ≠ H(H(x,y),z)
T - Conorms Properties
S(x,y) : [0,1]x[0,1] → [0,1]
S(x,0) = S(0,x) = x
S(x,1) = S(1,x) = 1
S(x,y) = S(y,x)
S(x,S(y,z)) = S(S(x,y),z)
S(x,y) ≥ S(u,v) if x ≥ u, y ≥ v

Figure 2. Aggregation Operators

One problem with T-norms and T-conorms is that contention within aggregation is not captured. By contention I am referring to a vast difference of decision values between classifiers. However, contention can be captured and considered appropriately. Typically, if contention exists a system needs to reflect caution. In other words, if we are minimizing false positives, if contention exists in a decision we may simply choose negative or choose a different aggregator for contentious decisions. If contention exists in a medical decision, it is likely that the initial diagnosis will report positive (cancer detected) and then further tests will be pursued. There are numerous ways to measure contention, and one of the simplest is to consider the difference between the max and min decision values. If this difference exceeds a certain threshold, contention exists and it may be best to choose a different aggregator or make a cautious decision.

4 Results

Experimental results involved the SEA dataset that was mentioned earlier. There are m=26 variables and n=2500 observations in the training dat. For our subspace selection, there are l=3 subspaces creating subspaces containing 9, 9, and 8 variables respectively. For each subspace we consider three principal components. Our gentic algorithm used the fitness function shown in Equation 3, roullette wheel selection, a crossover rate of .6 and mutation rate of .01. Our number of generations = population size = 50. Figure 3 is the correlation matrix of the principal components from our selected subspaces, where $\max_{\forall \pi_q} \mid \rho_{ij} \mid = .4 \ \forall (i \neq j)$.

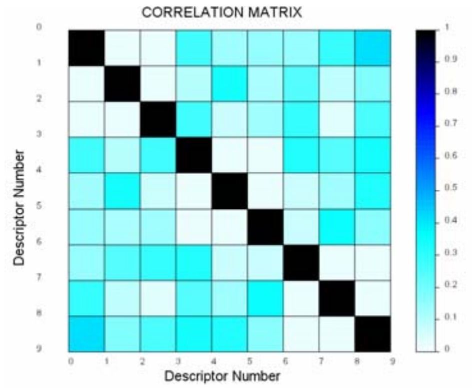

Figure 3. Correlation matrix of subspace principal components.

Given this subspace configuration, we utilized LIBSVM to calculate the 1 Class SVM decision variables. Given the decision variables d_{ij}, we mapped $d_{ij} \rightarrow o_{ij} \rightarrow \theta_{ij}$ and $d_{ij} \rightarrow p_{ij} \rightarrow \delta_{ij}$ as described in section 3.2.1. Our decision rule was to take the maximum value unless there was contention $>$ .5, and in this case we take the median of all decision values. The ROC curves shown in figure 4 illustrate the results.

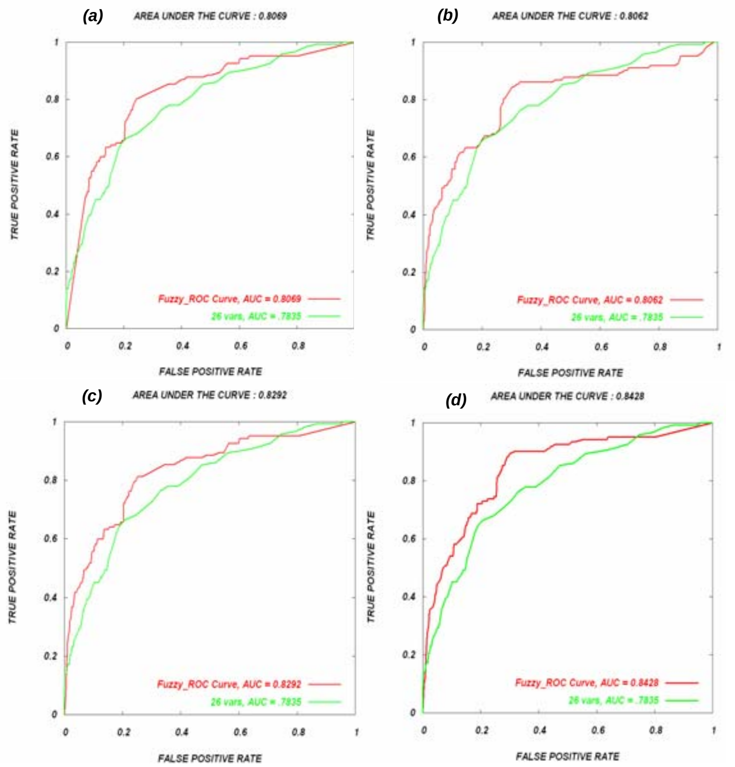

Figure 4. ROC plots illustrating effect of different decision rules

As Figure 4 illustrates, different decision create various outcomes. The best plot, (d), integrates all decision variables and contention. The table below describes the decision rules.

Table 1. Decision rules for ROC plots in Figure 4

ROC Plot	**Decision Rule for Each Observation (i); (t = threshold for contention**
(a)	$max(\delta_{ij})\forall j$
(b)	$max(\theta_{ij})\forall j$
(c)	$max(\delta_{ij}, \theta_{ij})\forall j$
(d)	if $t < .5$, $max(\delta_{ij}, \theta_{ij})\forall j$; if $t \geq .5$, median $(\delta_{ij}, \theta_{ij})\forall j$

5 Conclusions

This paper discusses a framework for a difficult domain of decision making: the unsupervised, unbalanced, binary classification problem with high dimensionality. It is common to encounter this domain in both the medical community and the security community. However, different risk aversion creates different policies for decisions. This framework capitalizes on theory from multivariate statistics, optimization, and information theory to present an approach for decision making and creation of such policies. Future work includes finding alternate approaches for finding optimal orthogonal subspaces. The goal is that the research enclosed in this paper will improve our ability to find synergistic combinations of classifiers and subspaces, and more importantly that this research will grow into applications for the improvement of security policies and other policies that address unbalanced, unsupervised, binary classification problems.

Acknowledgments

The authors would like to acknowledge Yongqiang Zhang for graciously sharing his data and providing insightful discussions.

References

1. Piero Bonissone, Kai Goebel, and Weizhong Yan. Classifier fusion using triangular norms. Cagliari, Italy, June 2004. Proceedings of Multiple Classifier Systems (MCS) 2004.
2. Chih Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. http://www.scie.ntu.edu.tw/ cjlin/libsvm, Accessed 5 September, 2004.
3. Yunqiang Chen, Xiang Zhou, and Thomas S. Huang. One-class svm for learning in image retrieval. Thessaloniki, Greece, 2001. Proceedings of IEEE International Conference on Image Processing.
4. Scott Coull, Joel Branch, and Boleslaw K. Szymanski. Intrusion detection: A bioinformatics approach. Las Vegas, Nevada, December 2001. Proceedings of the 19th Annual Computer Security Applications Conference.
5. William DuMouchel, Wen Hua Ju, Alan F. Karr, Matthius Schonlau, Martin Theus, and Yehuda Vardi. Computer intrusion: Detecting masquerades. *Statistical Science*, 16(1):1–17, 2001.
6. William DuMouchel and Matthius Schonlau. A fast computer intrusion detection algorithm based on hypothesis testing of command transition probabilities. pages 189–193. The Fourth International Conference of Knowledge Discovery and Data Mining, August 1998.
7. William DuMouchel and Matthius Schonlau. A comparison of test statistics for computer intrusion detection based on principal components regression of transition probabilities. pages 404–413. Proceedings of the 30th Symposium on the Interface: Computing Science and Statistics, 1999.
8. Paul F. Evangelista, Mark J. Embrechts, and Boleslaw K. Szymanski. Computer intrusion detection through predictive models. St. Louis, Missouri, November 2004. Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining and Complex Systems.
9. Tom Fawcett. ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. Palo Alto, CA, 2003. Technical Report HPL-2003-4, Hewlett Packard.
10. Ludmila I. Kuncheva. That Elusive Diversity in Classifier Ensembles. Mallorca, Spain, 2003. Proceedings of 1st Iberian Conference on Pattern Recognition and Image Analysis.
11. Ludmila I. Kuncheva and C.J. Whitaker. Measures of Diversity in Classifier Ensembles. *Machine Learning*, 51:181–207, 2003.
12. Wenke Lee and Salvatore J. Stolfo. A framework for constructing features

and models for intrusion detection systems. *ACM Transactions on Information and System Security (TISSEC)*, 3(4):227–261, 2000.

13. Roy A. Maxion. Masquerade detection using enriched command lines. San Francisco, CA, June 2003. International Conference on Dependable Systems and Networks.
14. Roy A. Maxion and Tahlia N. Townsend. Masquerade detection using truncated command lines. Washington, D.C., June 2002. International Conference on Dependable Systems and Networks.
15. Bernhard Scholkopf, John C. Platt, John Shawe Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.
16. Matthius Schonlau and Martin Theus. Intrusion detection based on structural zeroes. *Statistical Computing and Graphics Newsletter*, 9(1):12–17, 1998.
17. Matthius Schonlau and Martin Theus. Detecting masquerades in intrusion detection based on unpopular commands. *Information Processing Letters*, 76(1-2):33–38, 2000.
18. Salvatore Stolfo and Ke Wang. One class training for masquerade detection. Florida, 19 November 2003. 3rd IEEE Conference Data Mining Workshop on Data Mining for Computer Security.
19. Geoffrey I. Webb and Zijan Zheng. Multi-strategy ensemble learning: Reducing error by combining ensemble learning techniques. *IEEE Transactions on Knowledge and Data Engineering*, 16(8):980–991, 2004.