# Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection

Jiong Zhang and Mohammad Zulkernine
School of Computing
Queen's University, Kingston
Ontario, Canada K7L 3N6
{zhang, mzulker} @cs.queensu.ca

***Abstract*-Anomaly detection is a critical issue in Network Intrusion Detection Systems (NIDSs). Most anomaly based NIDSs employ supervised algorithms, whose performances highly depend on attack-free training data. However, this kind of training data is difficult to obtain in real world network environment. Moreover, with changing network environment or services, patterns of normal traffic will be changed. This leads to high false positive rate of supervised NIDSs. Unsupervised outlier detection can overcome the drawbacks of supervised anomaly detection. Therefore, we apply one of the efficient data mining algorithms called random forests algorithm in anomaly based NIDSs. Without attack-free training data, random forests algorithm can detect outliers in datasets of network traffic. In this paper, we discuss our framework of anomaly based network intrusion detection. In the framework, patterns of network services are built by random forests algorithm over traffic data. Intrusions are detected by determining outliers related to the built patterns. We present the modification on the outlier detection algorithm of random forests. We also report our experimental results over the KDD'99 dataset. The results show that the proposed approach is comparable to previously reported unsupervised anomaly detection approaches evaluated over the KDD'99 dataset.**

## I. INTRODUCTION

With the tremendous growth of network-based services and sensitive information on networks, the number and the severity of network-based computer attacks have significantly increased. Although a wide range of security technologies such as information encryption, access control, and intrusion prevention can protect network-based systems, there are still many undetected intrusions. Thus, Intrusion Detection Systems (IDSs) play a vital role in network security. Network Intrusion Detection Systems (NIDSs) detect attacks by observing various network activities, while Host-based Intrusion Detection Systems (HIDSs) detect intrusions in an individual host.

There are two major intrusion detection techniques: misuse detection and anomaly detection. Misuse detection discovers attacks based on the patterns extracted from known intrusions. Anomaly detection identifies attacks based on the deviations from the established profiles of normal activities. Activities that exceed thresholds of the deviations are detected as attacks. Misuse detection has low false positive rate, but cannot detect new types of attacks. Anomaly detection can detect unknown attacks, under a basic assumption that attacks deviate from normal behavior.

Currently, many NIDSs such as Snort [14] are rule-based systems, which employ misuse detection techniques and have limited extensibility for novel attacks. To detect novel attacks, many anomaly detection systems are developed. Most of them are based on supervised approaches [3, 5, 23]. For instance, ADAM [23] employs association rules algorithm in intrusion detection. ADAM builds a profile of normal activities over attack-free training data, and then detects attacks with the previously built profile. The problem of ADAM is the high dependency on training data for normal activities. However, the attack-free training data is difficult to come by, since there is no guarantee that we can prevent all attacks in real world networks. Actually, one of the most popular ways to undermine anomaly based IDSs is to incorporate some intrusive activities into the training data [13]. The IDSs trained by the training data with intrusive activities will lose the ability to detect this kind of intrusions. Another problem of the supervised anomaly based IDS is high false positive rate when network environment or services are changed. Since training data only contain historical activities, profile of normal activities can only include historical patterns of normal behavior. Therefore, new activities due to changing of network environment or services will deviate from the previously built profile and are detected as attacks. That will raise false positives.

To overcome the limitations of supervised anomaly based systems, a number of IDSs employ unsupervised approaches [1, 2, 9]. Unsupervised anomaly detection does not need attack-free training data. It detects attacks by determining unusual activities from data under two assumptions [9]:

- The majority of activities are normal.
- Attacks statistically deviate from normal activities.

The unusual activities are outliers that are inconsistent with the remainder of data set [11]. Thus, outlier detection techniques can be applied in unsupervised anomaly detection. Actually, outlier detection has been used in a number of practical applications such as credit card fraud detection, voting irregularity analysis, and severe weather prediction [12].



1-4244-0355-3/06/$20.00 (c) 2006 IEEE

We propose an approach to use outlier detection technique provided by random forests algorithm in anomaly intrusion detection. Random forests is an ensemble classification and regression approach, which is unsurpassable in accuracy among current data mining algorithms [15]. Random forests algorithm has been used extensively in different applications. For instance, it has been applied to prediction [16, 17], probability estimation [24], and pattern analysis in multimedia information retrieval and bioinformatics [18]. Unfortunately, to the best of our knowledge, random forests algorithm has not been applied in anomaly intrusion detection.

The main challenge of anomaly intrusion detection is to minimize false positives. The outlier detection technique is effective to reduce false positive rate with a desirable detection rate. The proposed approach is evaluated using the KDD'99 dataset, which were used for the third International Knowledge Discovery and Data Mining Tools Competition [19]. Our experimental results show that the detection performance is improved by our approach of using the outlier detection technique.

The paper is organized as follows. In Section II, we discuss the related work. In Section III, we describe in detail the approach to detect outliers using random forests algorithm. The experiments and performance evaluations are presented in Section IV. Finally, we summarize the paper and outline our future research plans in Section V.

## II. RELATED WORK

Anomaly detection has been an important subject in intrusion detection research. Various anomaly detection approaches have been proposed and implemented.

Unsupervised anomaly detection in NIDSs as discussed below is a new research area [9]. Eskin, et al. [1] investigate three algorithms in unsupervised anomaly detection: cluster-based estimation, k-nearest neighbor, and one class SVM (Support Vector Machine). Other researchers [2, 9] apply clustering approaches in unsupervised NIDSs.

Supervised anomaly detection has been studied extensively. ADAM (Audit Data Analysis and Mining) [23] is widely known and well published project in the field. It is an on-line network-based IDS. ADAM can detect known attacks as well as unknown attacks. ADAM builds the profile of normal behavior from attack-free training data and represents the profile as a set of association rules. At run-time, ADAM detects suspicious connections according to the profile. Other supervised approaches are also applied to anomaly detection, such as fuzzy data mining and genetic algorithm [3], neural networks [4, 8], and SVM [5].

In our previous work, we applied random forests algorithm in misuse detection [7]. In this paper, we employ the outlier detection function provided by random forests algorithm for unsupervised anomaly detection. Random forests algorithm is more accurate and efficient on large datasets with many features such as datasets of network traffic.

## III. DETECTING OUTLIERS

In this section, we describe the proposed framework of the NIDS, and illustrate how to use random forests algorithm to detect outliers over datasets of network traffic.

### A. Overview of the framework

The proposed framework applies random forests algorithm to detect novel intrusions. The framework is shown in Fig. 1.

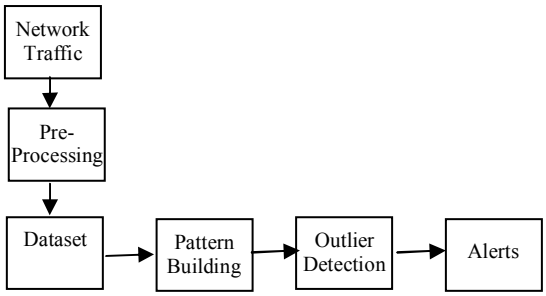


Figure 1. The framework of the unsupervised anomaly NIDS

The NIDS captures the network traffic and constructs dataset by pre-processing. After that, the service-based patterns are built over the dataset using random forests algorithm. With the built patterns, we can find outliers related to each pattern. Then the system will raise alerts when outliers are detected.

After capturing network traffic, the processing is off-line. Due to the high computational requirements by the outlier detection algorithm, on-line processing is not suitable in real network environment.

### B. Random forests algorithm

The random forests [15] is an ensemble of un-pruned classification or regression trees. It is unsurpassable in accuracy among the current data mining algorithms, especially for large datasets with many features.

Random forests algorithm generates many classification trees. Each tree is constructed by a different bootstrap sample from the original data using a tree classification algorithm. After the forest is formed, a new object that needs to be classified is put down each of the tree in the forest for classification. Each tree gives a vote that indicates the tree's decision about the class of the object. The forest chooses the class with the most votes for the object.

In random forests algorithm, there is no need for cross-validation or a test set to get an unbiased estimate of the test error. Since each tree is constructed using the bootstrap sample, approximately one-third of the cases are left out of the bootstrap samples and not used in training. These cases are called out of bag (oob) cases. These oob cases are used to get a run-time unbiased estimate of the classification error as trees are added to the forest.

### C. Mining patterns of network services

Network traffic can be categorized by services (*e.g.,* http, telnet, and ftp). Each network service has its own pattern.



*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.*

Therefore, we can build patterns of network services using random forests algorithm. However, random forests algorithm is supervised, so we need datasets labeled by network services. Since the information of network services is in network packets, network traffic can be labeled by the services automatically instead of time consuming manual processing. Actually, many datasets used to evaluate NIDSs can be labeled by network services with a little effort. For example, one of features in the KDD'99 dataset is service type which can be used as label.

Before building the patterns, we need to optimize the parameters of random forests algorithm. When the forest is growing, random features are selected at random out of the all features in the training data. The best split on these random features is used to split the node. The number of random features (*Mtry*) is held constant. The number of features employed in splitting each node for each tree is the primary tuning parameter (*Mtry*). To improve the performance of random forests algorithm, this parameter should be optimized. Another parameter is the number of trees in a forest.

We use the dataset to find the optimal value of the parameter *Mtry* and the number of the trees. The minimum error rate corresponds to the optimal values. Therefore, we use the different value of *Mtry* and number of the trees to build the forest, and evaluate the error rate of the forest. Then, we select the value corresponding to the minimum error rate to build the patterns of the services.

### *D. Unsupervised outlier detection*

We can detect intrusions by finding unusual activities or outliers. There are two types of outliers in the proposed NIDS. The first type is the activities that deviate significantly from others in the same network service. The second type is the activities whose patterns belong to other services other than their own service. For instance, if an http activity is classified as ftp service, the activity will be determined as an outlier.

Random forests algorithm uses proximities to find outliers whose proximities to all other cases in the entire data are generally small. The proximities are one of the most useful tools in random forests algorithm [15]. After the forest is constructed, all cases in the dataset are put down each tree in the forest. If cases $k$ and $n$ are in the same leaf of a tree, their proximity is increased by one. Finally, the proximities are normalized by dividing by the number of the trees.

For a dataset with N cases, the proximities originally formed a N×N matrix. The complexity of calculation is N×N. Datasets of network traffic are huge, so the calculation needs a lot of memory and CPU time. To improve the performance, we modify the algorithm to calculate the proximities. As we mentioned above, if a service activity is classified as another service, it will be determined as outlier. Therefore, we do not care about the proximity between two cases that belong to different services. $S_i$ denotes the number of cases in service $i$. The complexity will be reduced to $\sum S_i \times S_i$ after the modification.

With respect to random forests algorithm, outliers can be defined as the cases whose proximities to other cases in the dataset are generally small [15]. Outlier-ness indicates a degree of being an outlier. It can be calculated over proximities. *class(k) = j* denotes that $k$ belongs to class $j$. *prox(n,k)* denotes the proximity between cases $n$ and $k$. The average proximity from case $n$ in class $j$ to case $k$ (the rest of data in class $j$) is computed as:

$$\overline{P}(n) = \sum_{class(k)=j} prox^2(n,k) \qquad (1)$$

$N$ denotes the number of cases in the dataset. The raw outlier-ness of case $n$ is defines as:

$$N / \overline{P}(n) \qquad (2)$$

In each class, the median and the absolute deviation of all raw outlier-ness are calculated. The median is subtracted from each raw outlier-ness. The result of the subtraction is divided by the absolute deviation to get the final outlier-ness. If the outlier-ness of a case is large, the proximity is small, and the case is determined as an outlier.

To detect outliers in a dataset of network traffic, we build patterns of services over the dataset. Then, we calculate the proximity and outlier-ness for each activity. An activity that exceeds a specified threshold will be determined as an outlier.

## IV. EXPERIMENTS AND RESULTS

In this section, we summarize our experimental results to detect intrusions using the unsupervised outlier detection technique over the KDD'99 dataset. We first describe the datasets used in the experiments. Then we evaluate our approach and discuss the results.

### *A. Dataset and preprocessing*

Under the sponsorship of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), MIT Lincoln Laboratory has collected and distributed the datasets for the evaluation of computer network intrusion detection systems [20, 21]. The DARPA dataset is the most popular dataset used to test and evaluate a large number of IDSs. The KDD'99 dataset is a subset of the DARPA dataset prepared by Sal Stofo and Wenke Lee [25]. The data was preprocessed by extracting 41 features (*e.g.,* protocol type, service, and flag) from the tcpdump data in the 1998 DARPA dataset. The KDD'99 dataset can be used without further time-consuming preprocessing and different IDSs can compare with each other by working on the same dataset. Therefore, we carry out our experiments on the KDD'99 dataset.

The full training set, one of the KDD'99 datasets, has 4,898,431 connections, which contains attacks. The attacks in the dataset fall into four categories [19]: DoS (Denial of Service), R2L (unauthorized access from a remote machine),



*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.*

U2R (unauthorized access to root privileges), and probing. The dataset is labeled by type of attacks. Since our approach is unsupervised, the dataset does not satisfy the needs of our experiments. We must remove the labels that indicate types of attacks from the dataset.

To generate new datasets for our experiments, we first separate the dataset into two pools according to the labels. One includes normal connections. Another includes attacks. Then, we remove all the labels from the pools. However, we need the data labeled by service to build patterns of services, so we use service feature in the dataset as label. As a result, all the data contains 40 features and is labeled by service.

For our experiments, we choose five most popular network services: ftp, http, pop, smtp, and telnet. By selecting ftp, pop, telnet, 5% http, and 10% smtp normal connections, we generate a dataset called normal dataset, which contains 47,426 normal connections. Finally, by injecting anomalies from the pool of attacks into normal dataset, we generate four new datasets: 1%, 2%, 5%, and 10% dataset. 1% (2%, 5%, and 10%) dataset means that 1% (2%, 5%, and 10%) of connections in the dataset are attacks.

### *B. Evaluation and discussion*

We carry out the first experiment over the 1% attack dataset. We first optimize the parameters (*Mtry* and the number of trees) of random forests algorithm by feeding the dataset into the NIDS. The NIDS builds patterns of the network services with different values of the parameters, and then calculates the oob error rate. The values corresponding to the lowest oob error rate are optimized.

With the optimized parameters, we build the patterns of the network services. Over the built patterns, the NIDS calculates the outlier-ness of each connection. Fig. 2 plots the outlier-ness of the 1% attack dataset. Since the attacks are injected at the beginning of the dataset, the figure shows the outlier-ness of the attacks is much higher than most of normal activities. Some normal activities also have high outlier-ness. That leads to false positives. The NIDS will raise an alert if an outlier-ness of a connection exceeds a specified threshold.

We evaluate the performance of our system by the detection rate and the false positive rate. The detection rate is the number of attacks detected by the system divided by the number of attacks in the dataset. The false positive rate is the number of normal connections that are misclassified as attacks divided by the number of normal connections in the dataset. We can evaluate the performance by varying the threshold of outlier-ness.

In intrusion detection, ROC (Receiver Operating Characteristic) curve is often used to measure performance of IDSs. The ROC curve is a plot of the detection rate against the false positive rate. Fig. 3 plots ROC curve to show the relationship between the detection rates and the false positive rates over the dataset.

The result indicates that our system can achieve a high detection rate with a low false positive rate. Compared to other unsupervised anomaly based systems [1, 9], our system provides better performance over the KDD'99 dataset while the false positive rate is low. Table I lists some results from Eskin, et al. [1]. The results from the other detection systems

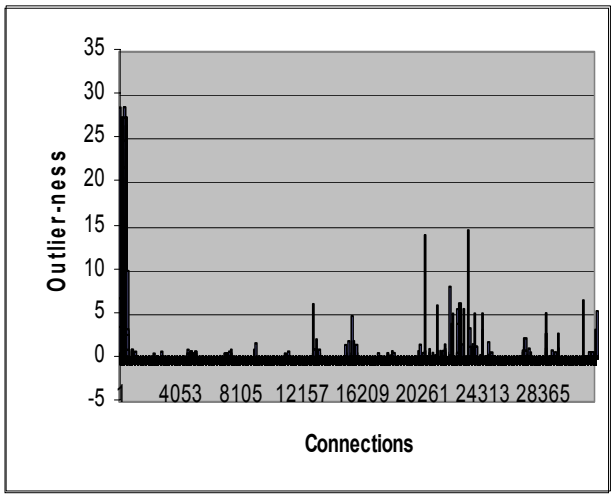


Figure 2. The outlier-ness of the 1% attack dataset

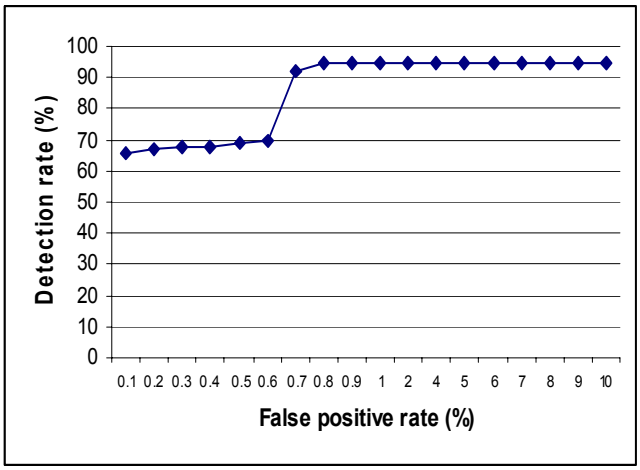


Figure 3. The ROC curve for the 1% attack dataset

TABLE I
THE PERFORMANCE OF EACH ALGORITHM OVER THE KDD'99 DATASET [1]

| Algorithm | Detection rate | False positive rate |
|---|---|---|
| Cluster | 66% | 2% |
| Cluster | 28% | 0.5% |
| K-NN | 11% | 4% |
| K-NN | 5% | 2% |
| SVM | 67% | 4% |
| SVM | 5% | 3% |

show that the detection rate is reduced significantly when the false positive rate is low (below 1%). Although our



*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.*

experiments are carried out under different conditions, Fig. 3 shows that our system still provides relatively higher detection rates when the false positive rates are low. For example, the detection rate is 95% when the false positive rate is 1%. When the false positive rate is reduced to 0.1%, the detection rate is still over 60%.

To evaluate our system under different number of attacks, we carry out the experiments over the 1%, 2%, 5%, and 10% attack dataset. Fig. 4 plots the ROCs for each dataset. The result shows that the performance tends to be reduced while increasing number of attacks.

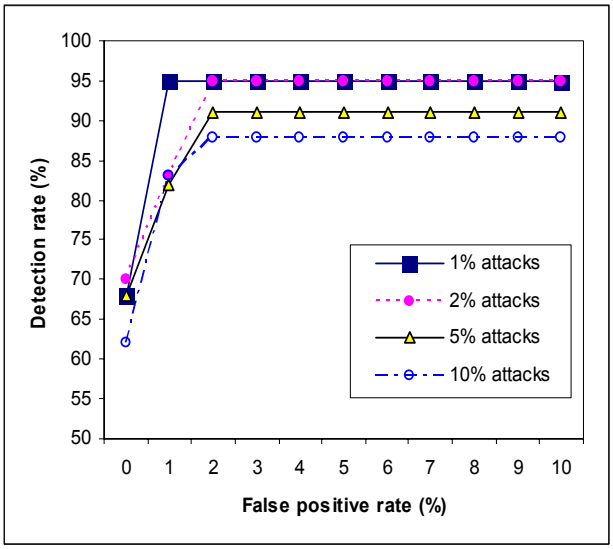


Figure 4. The ROC curves for the datasets

### C. *Implementation*

We develop the NIDS using WEKA (Waikato Environment for Knowledge Analysis) [22]. WEKA is an open source Java package which contains machine learning algorithms for data mining tasks. However, WEKA does not implement the outlier detection function in the random forests algorithm. Therefore, we modify the source code of WEKA to implement outlier detection.

## V. CONCLUSION AND FUTURE WORK

In this paper, we propose a new framework of unsupervised anomaly NIDS based on the outlier detection technique in random forests algorithm. The framework builds the patterns of network services over datasets labeled by the services. With the built patterns, the framework detects attacks in the datasets using the outlier detection algorithm.

Due to large population of datasets used in NIDSs, the process to detect outliers is very time-consuming and costs a large amount of memory. To improve the performance, we modify the original outlier detection algorithm to reduce the calculation complexity, under the assumption that each network service has its own pattern for normal activities.

Compared to supervised approaches, our approach breaks the dependency on attack-free training datasets. The experimental results over the KDD'99 dataset confirm the effectiveness of our approach using the unsupervised detection technique.

The performance of our system is comparable to that of other reported unsupervised anomaly detection approaches. Especially, our approach achieve higher detection rate when the false positive rate is low. It is more significant for NIDSs, since high false positive rate will make NIDSs useless.

Due to high complexity of the unsupervised anomaly detection algorithm, low detection speed performance of the approach makes real time detection impossible. However, the approach can detect novel intrusions without attack-free training data. The detected novel intrusions can be used to train real time supervised misuse detection systems. Therefore, the trained misuse detection systems can detect the novel intrusions in real time.

The results also show that the performance tends to be reduced with increasing number of attack connections. That is a problem of unsupervised systems. Some attacks (*e.g.,* DoS) produce a large number of connections, which may undermine an unsupervised anomaly detection system. To overcome the problem, we will incorporate both anomaly based and misuse based approaches into the NIDS in the future. Misuse approach can detect known attacks. By removing known attacks, the number of attacks can be reduced significantly in datasets for unsupervised anomaly detection. Misuse detection has high detection rate with low false positive rate. Anomaly detection can detect novel attacks to increase the detection rate. Therefore, combining misuse and anomaly detection can improve the overall performance of the NIDS.

## ACKNOWLEDGMENT

This research work is funded by The Mathematics of Information Technology and Complex Systems (MITACS).

## REFERENCES

[1] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data", *Applications of Data Mining in Computer Security*, Kluwer, 2002.

[2] Rasheda Smith, Alan Bivens, Mark Embrechts, Chandrika Palagiri, and Boleslaw Szymanski, "Clustering Approaches for Anomaly Based Intrusion Detection", *Walter Lincoln Hawkins Graduate Research Conference 2002 Proceedings*, New York, USA, October 2002.

[3] Susan M.Bridges, and Rayford B. Vaughn, "Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection", *Proceedings of the National Information Systems Security Conference (NISSC)*, Baltimore, MD, October, 2000.

[4] Alan Bivens, Mark Embrechts, Chandrika Palagiri, Rasheda Smith, and Boleslaw Szymanski, "Network-Based Intrusion Detection Using Neural Networks", *Artificial Neural Networks In Engineering*, St. Louis, Missouri, November 2002.

[5] Q.A. Tran, H. Duan, and X. Li, "One-class Support Vector Machine for Anomaly Network Traffic Detection", *The 2nd Network Research Workshop of the 18th APAN*, Cairns, Australia, 2004.

[6] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava & V. Kumar, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection", *Proceedings of Third SIAM Conference on Data Mining*, San Francisco, May 2003.



*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.*

[7] J. Zhang and M. Zulkernine, "Network Intrusion Detection Using Random Forests", *Proc. of the Third Annual Conference on Privacy, Security and Trust*, St. Andrews, New Brunswick, Canada, October 2005.
[8] M. Ramadas, S. Ostermann and B. Tjaden, "Detecting Anomalous Network Traffic with Self-Organizing Maps", *RAID*, 2003.
[9] Kingsly Leung and Christopher Leckie, "Unsupervised Anomaly Detection in Network Intrusion Detection Using Clusters", *Australasian Computer Science Conference*, Newcastle, NSW, Australia, 2005.
[10] Ian H. Witten, and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann publishers, October 1999.
[11] V. Barnett and T. Lewis, *Outliers in Statistical Data*, John Wiley, 1994.
[12] C. T. Lu, D. Chen, and Y. Kou, "Algorithms for Spatial Outlier Detection", *Proceedings of 3rd IEEE International Conference on Data Mining*, Melbourne, Florida, November 2003.
[13] K. Tan, K. Killourhy, and R. Maxion, "Undermining an anomaly based intrusion detection system using common exploits", *RAID*, Zurich, Switzerland, Oct. 2002.
[14] Snort, Network Intrusion Detection System, http://www.snort.org.
[15] L. Breiman, "Random Forests", *Machine Learning* 45(1):5–32, 2001.
[16] Lan Guo, Yan Ma, Bojan Cukic, and Harshinder Singh, "Robust Prediction of Fault-Proneness by Random Forests", *Proceedings of the 15th International Symposium on Software Reliability Engineering (ISSRE'04)*, pp. 417-428, Brittany, France, November 2004.
[17] Bogdan E. Popescu, and Jerome H. Friedman, *Ensemble Learning for Prediction*, Doctoral Thesis, Stanford University, January 2004.
[18] Yimin Wu, *High-dimensional Pattern Analysis in Multimedia Information Retrieval and Bioinformatics*, Doctoral Thesis, State University of New York, January 2004.
[19] KDD'99 datasets, The UCI KDD Archive, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Irvine, CA, USA, 1999.
[20] M. Mahoney and P. Chan, "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection", *Proceeding of Recent Advances in Intrusion Detection (RAID),* Pittsburgh, USA, September 2003.
[21] MIT Lincoln Laboratory, DARPA Intrusion Detection Evaluation, http://www.ll.mit.edu/IST/ideval/, MA, USA.
[22] WEKA software, Machine Learning, http://www.cs.waikato.ac.nz/ml/weka/, The University of Waikato, Hamilton, New Zealand.
[23] Daniel Barbarra, Julia Couto, Sushil Jajodia, Leonard Popyack, and Ningning Wu, "ADAM: Detecting Intrusions by Data Mining", *Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security*, T1A3 1100 United States Military Academy, West Point, NY, June 2001.
[24] Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng, "Probability Estimates for Multi-class Classification by Pairwise Coupling", *The Journal of Machine Learning Research*, Volume 5, December 2004.
[25] Charles Elkan, "Results of the KDD'99 Classifier Learning", *SIGKDD Explorations* 1(2): 63-64, 2000.



*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2006 proceedings.*