A Detailed Analysis of the KDD CUP 99 Data Set

Tavallaee, Mahbod; Bagheri, Ebrahim; Lu, Wei; Ghorbani, Ali-A.

Publisher's version / Version de l'éditeur:

Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009, 2009-07-10

A Detailed Analysis of the KDD CUP 99 Data Set

Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani

Abstract— **During the last decade, anomaly detection has attracted the attention of many researchers to overcome the weakness of signature-based IDSs in detecting novel attacks, and KDDCUP'99 is the mostly widely used data set for the evaluation of these systems. Having conducted an statistical analysis on this data set, we found two important issues which highly affects the performance of evaluated systems, and results in a very poor evaluation of anomaly detection approaches. To solve these issues, we have proposed a new data set, NSL-KDD, which consists of selected records of the complete KDD data set and does not suffer from any of mentioned shortcomings.**

I. Introduction

With the enormous growth of computer networks usage and the huge increase in the number of applications running on top of it, network security is becoming increasingly more important. As it is shown in [1], all the computer systems suffer from security vulnerabilities which are both technically difficult and economically costly to be solved by the manufacturers. Therefore, the role of Intrusion Detection Systems (IDSs), as special-purpose devices to detect anomalies and attacks in the network, is becoming more important. The research in the intrusion detection field has been mostly focused on anomaly-based and misuse-based detection techniques for a long time. While misuse-based detection is generally favored in commercial products due to its predictability and high accuracy, in academic research anomaly detection is typically conceived as a more powerful method due to its theoretical potential for addressing novel attacks.

Conducting a thorough analysis of the recent research trend in anomaly detection, one will encounter several machine learning methods reported to have a very high detection rate of 98% while keeping the false alarm rate at 1% [2]. However, when we look at the state of the art IDS solutions and commercial tools, there is no evidence of using anomaly detection approaches, and practitioners still think that it is an immature technology. To find the reason of this contrast, we studied the details of the research done in anomaly detection and considered various aspects such as learning and detection approaches, training data sets, testing data sets, and evaluation methods. Our study shows that there are some inherent problems in the KDDCUP'99 data set [3], which is widely used as one of the few publicly available data sets for network-based anomaly detection systems.

Mahbod Tavallaee, Wei Lu, and Ali A. Ghorbani are with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada (email: {m.tavallaee, wlu, ghorbani}@unb.ca), and Ebrahim Bagheri is with the Institute for Information Technology, National Research Council Canada (email: ebrahim.bagheri@nrc-cnrc.gc.ca)

This work was supported by the funding from the Atlantic Canada Opportunity Agency (ACOA) through the Atlantic Innovation Fund (AIF) to Dr. Ali Ghorbani.

The first important deficiency in the KDD data set is the huge number of redundant records. Analyzing KDD train and test sets, we found that about 78% and 75% of the records are duplicated in the train and test set, respectively. This large amount of redundant records in the train set will cause learning algorithms to be biased towards the more frequent records, and thus prevent it from learning unfrequent records which are usually more harmful to networks such as U2R attacks. The existence of these repeated records in the test set, on the other hand, will cause the evaluation results to be biased by the methods which have better detection rates on the frequent records.

In addition, to analyze the difficulty level of the records in KDD data set, we employed 21 learned machines (7 learners, each trained 3 times with different train sets) to label the records of the entire KDD train and test sets, which provides us with 21 predicted labels for each record. Surprisingly, about 98% of the records in the train set and 86% of the records in the test set were correctly classified with all the 21 learners. The reason we got these statistics on both KDD train and test sets is that in many papers, random parts of the KDD train set are used as test sets. As a result, they achieve about 98% classification rate applying very simple machine learning methods. Even applying the KDD test set will result in having a minimum classification rate of 86%, which makes the comparison of IDSs quite difficult since they all vary in the range of 86% to 100%.

In this paper, we have provided a solution to solve the two mentioned issues, resulting in new train and test sets which consist of selected records of the complete KDD data set. The provided data set does not suffer from any of the mentioned problems. Furthermore, the number of records in the train and test sets are reasonable. This advantage makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research work will be consistent and comparable.

The new version of KDD data set, NSL-KDD is publicly available for researchers through our website[1]. Although, the data set still suffers from some of the problems discussed by McHugh [4] and may not be a perfect representative of existing real networks, because of the lack of public data sets for network-based IDSs, we believe it still can be applied as an effective benchmark data set to help researchers compare different intrusion detection methods.

The rest of the paper is organized as follows. Section II introduces the KDDCUP99 data set which is wildly used in anomaly detection. In Section III, we first review the issues

[1] http://nsl.cs.unb.ca/KDD/NSL-KDD.html

in DARPA'98 and then discuss the possible existence of those problems in KDD'99. The statistical observations of the KDD data set will be explained in Section IV. Section V provides some solutions for the existing problems in the KDD data set. Finally, in Section VI we draw conclusion.

II. KDD CUP 99 DATA SET DESCRIPTION

Since 1999, KDD'99 [3] has been the most wildly used data set for the evaluation of anomaly detection methods. This data set is prepared by Stolfo et al. [5] and is built based on the data captured in DARPA'98 IDS evaluation program [6]. DARPA'98 is about 4 gigabytes of compressed raw (binary) tcpdump data of 7 weeks of network traffic, which can be processed into about 5 million connection records, each with about 100 bytes. The two weeks of test data have around 2 million connection records. KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type. The simulated attacks fall in one of the following four categories:

1) **Denial of Service Attack (DoS):** is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.
2) **User to Root Attack (U2R):** is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.
3) **Remote to Local Attack (R2L):** occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.
4) **Probing Attack:** is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data which make the task more realistic. Some intrusion experts believe that most novel attacks are variants of known attacks and the signature of known attacks can be sufficient to catch novel variants. The datasets contain a total number of 24 training attack types, with an additional 14 types in the test data only. The name and detail description of the training attack types are listed in [7].

KDD'99 features can be classified into three groups:

1) **Basic features:** this category encapsulates all the attributes that can be extracted from a TCP/IP connection. Most of these features leading to an implicit delay in detection.
2) **Traffic features:** this category includes features that are computed with respect to a window interval and is divided into two groups:
   a) **"same host" features:** examine only the connections in the past 2 seconds that have the same destination host as the current connection, and calculate statistics related to protocol behavior, service, etc.
   b) **"same service" features:** examine only the connections in the past 2 seconds that have the same service as the current connection.

   The two aforementioned types of "traffic" features are called time-based. However, there are several slow probing attacks that scan the hosts (or ports) using a much larger time interval than 2 seconds, for example, one in every minute. As a result, these attacks do not produce intrusion patterns with a time window of 2 seconds. To solve this problem, the "same host" and "same service" features are re-calculated but based on the connection window of 100 connections rather than a time window of 2 seconds. These features are called connection-based traffic features.
3) **Content features:** unlike most of the DoS and Probing attacks, the R2L and U2R attacks don't have any intrusion frequent sequential patterns. This is because the DoS and Probing attacks involve many connections to some host(s) in a very short period of time; however the R2L and U2R attacks are embedded in the data portions of the packets, and normally involves only a single connection. To detect these kinds of attacks, we need some features to be able to look for suspicious behavior in the data portion, e.g., number of failed login attempts. These features are called content features.

III. INHERENT PROBLEMS OF KDD'99 DATA SET

As it is mentioned in the previous section, KDD'99 is built based on the data captured in DARPA'98 which has been criticized by McHugh [4], mainly because of the characteristics of the synthetic data. As a result, some of the existing problems in DARPA'98 remain in KDD'99. However, there are some deliberate or unintentional improvements, along with additional problems. In the following we first review the issues in DARPA'98 and then discuss the possible existence of those problems in KDD'99. Finally, we discuss new issues observed in the KDD data set.

1) For the sake of privacy, the experiments chose to synthesize both the background and the attack data, and the data is claimed to be similar to that observed during several month of sampling data from a number of Air Force bases. However, neither analytical nor experimental validation of the data's false alarm characteristics were undertaken. Furthermore, the workload of the synthesized data does not seem to be similar to the traffic in real networks.
2) Traffic collectors such as TCPdump, which is used in DARPA'98, are very likely to become overloaded and drop packets in heavy traffic load. However, there was

no examination to check the possibility of the dropped packets.

3) There is no exact definition of the attacks. For example, probing is not necessarily an attack type unless the number of iterations exceeds an specific threshold. Similarly, a packet that causes a buffer overflow is not always representative of an attack. Under such conditions, there should be an agreement on the definitions between the evaluator and evaluated. In DARPA'98, however, there is no specific definitions of the network attacks.

In addition, there are some critiques of attack taxonomies and performance measures. However, these issues are not of much interest in this paper since most of the anomaly detection systems work with binary labels, i.e., anomalous and normal, rather than identifying the detailed information of the attacks. Besides, the performance measure applied in DARPA'98 Evaluation, ROC Curves, has been widely criticized, and since then many researchers have proposed new measures to overcome the existing deficiencies [8], [9], [10], [11], [12].

While McHugh's critique was mainly based on the procedure to generate the data set rather than analysis of the data, Mahoney and Chan [13] analyzed DARPA background network traffic and found evidence of simulation artifacts that could result in an overestimation of the performance of some anomaly detection techniques. In their paper, authors mentioned five types of anomalies leading to attack detection. However, analysis of the attacks in the DARPA data set revealed that many did not fit into any of these categories which are likely caused by simulation artifacts. As an example, the TTL (time to live) values of 126 and 253 appear only in hostile traffic, whereas in most background traffic the value is 127 and 254. Similarly, some attacks can be identified by anomalous source IP addresses or anomalies in the TCP window size field.

Fortunately the aforementioned simulation artifacts do not affect the KDD data set since the 41 features used in KDD are not related to any of the weaknesses mentioned in [13]. However, KDD suffers from additional problems not existing in the DARPA data set.

In [14], Portnoy et al. partitioned the KDD data set into ten subsets, each containing approximately 490,000 instances or 10% of the data. However, they observed that the distribution of the attacks in the KDD data set is very uneven which made cross-validation very difficult. Many of these subsets contained instances of only a single type. For example, the 4th, 5th, 6th, and 7th, 10% portions of the full data set contained only *smurf* attacks, and the data instances in the 8th subset were almost entirely *neptune* intrusions.

Similarly, same problem with *smurf* and *neptune* attacks in the KDD training data set is reported in [15]. The authors have mentioned two problems caused by including these attacks in the data set. First, these two types of DoS attacks constitute over 71% of the testing data set which completely affects the evaluation. Secondly, since they generate large volumes of traffic, they are easily detectable by other means and there is no need of using anomaly detection systems to find these attacks.

TABLE I
STATISTICS OF REDUNDANT RECORDS IN THE KDD TRAIN SET

	Original Records	**Distinct Records**	**Reduction Rate**
Attacks	3,925,650	262,178	93.32%
Normal	972,781	812,814	16.44%
Total	4,898,431	1,074,992	78.05%

TABLE II
STATISTICS OF REDUNDANT RECORDS IN THE KDD TEST SET

	Original Records	**Distinct Records**	**Reduction Rate**
Attacks	250,436	29,378	88.26%
Normal	60,591	47,911	20.92%
Total	311,027	77,289	75.15%

IV. STATISTICAL OBSERVATIONS

As was mentioned earlier, there are some problems in the KDD data set which cause the evaluation results on this data set to be unreliable. In this section we perform a set of experiments to show the existing deficiencies in KDD.

A. Redundant Records

One of the most important deficiencies in the KDD data set is the huge number of redundant records, which causes the learning algorithms to be biased towards the frequent records, and thus prevent them from learning unfrequent records which are usually more harmful to networks such as U2R and R2L attacks. In addition, the existence of these repeated records in the test set will cause the evaluation results to be biased by the methods which have better detection rates on the frequent records.

To solve this issue, we removed all the repeated records in the entire KDD train and test set, and kept only one copy of each record. Tables I and II illustrate the statistics of the reduction of repeated records in the KDD train and test sets, respectively.

While doing this process, we encountered two invalid records in the KDD test set, number 136,489 and 136,497. These two records contain an invalid value, ICMP, as their *service* feature. Therefore, we removed them from the KDD test set.

B. Level of Difficulty

The typical approach for performing anomaly detection using the KDD data set is to employ a customized machine learning algorithm to learn the general behavior of the data set in order to be able to differentiate between normal and malicious activities. For this purpose, the data set is divided into test and training segments, where the learner is trained using the training portion of the data set and is then evaluated

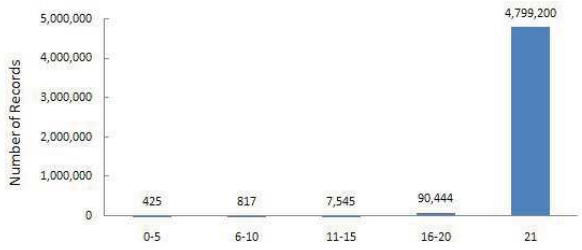

Fig. 1. The distribution of `#successfulPrediction` values for the KDD data set records

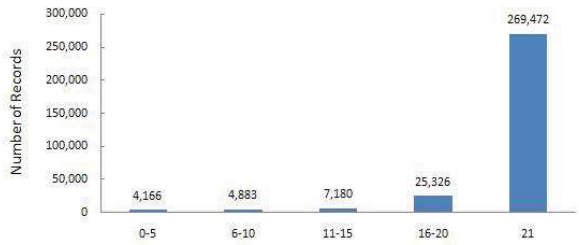

Fig. 2. The distribution of `#successfulPrediction` values for the KDD data set records

for its efficiency on the test portion. Many researchers within the general field of machine learning have attempted to devise complex learners to optimize accuracy and detection rate over the KDD'99 data set. In a similar approach, we have selected seven widely used machine learning techniques, namely J48 decision tree learning [16], Naive Bayes [17], NBTree [18], Random Forest [19], Random Tree [20], Multilayer Perceptron [21], and Support Vector Machine (SVM) [22] from the Weka [23] collection to learn the overall behavior of the KDD'99 data set. For the experiments, we applied Weka's default values as the input parameters of these methods.

Investigating the existing papers on the anomaly detection which have used the KDD data set, we found that there are two common approaches to apply KDD. In the first, KDD'99 training portion is employed for sampling both the train and test sets. However, in the second approach, the training samples are randomly collected from the KDD train set, while the samples for testing are arbitrarily selected from the KDD test set.

In order to perform our experiments, we randomly created three smaller subsets of the KDD train set each of which included fifty thousand records of information. Each of the learners where trained over the created train sets. We then employed the 21 learned machines (7 learners, each trained 3 times) to label the records of the entire KDD train and test sets, which provides us with 21 predicated labels for each record. Further, we annotated each record of the data set with a `#successfulPrediction` value, which was initialized to zero. Now, since the KDD data set provides the correct label for each record, we compared the predicated label of each record given by a specific learner with the actual label, where we incremented `#successfulPrediction` by one if a match was found. Through this process, we calculated the number of learners that were able to correctly label that given record. The highest value for `#successfulPrediction` is 21, which conveys the fact that all learners were able to correctly predict the label of that record. Figure 1 and 2 illustrate the distribution of `#successfulPrediction` values for the KDD train and test sets, respectively.

It can be clearly seen from Figure 1 and 2 that 97.97% and 86.64% of the records in the KDD train and test sets have been correctly labeled by all 21 classifiers. The obvious observation from these figures is that the application of typical learning machines to this data set would result in high accuracy rates. This shows that evaluating methods on the basis of accuracy, detection rate and false positive rate on the KDD data set is not an appropriate option.

TABLE III
STATISTICS OF RANDOMLY SELECTED RECORDS FROM KDD TRAIN SET

	Distinct Records	Percentage	Selected Records
0-5	407	0.04	407
6-10	768	0.07	767
11-15	6,525	0.61	6,485
16-20	58,995	5.49	55,757
21	1,008,297	93.80	62,557
Total	1,074,992	100.00	125,973

V. OUR SOLUTION

To solve the issues mentioned in the previous section, we first removed all the redundant records in both train and test sets. Furthermore, to create a more challenging subset of the KDD data set, we randomly sampled records from the `#successfulPrediction` value groups shown in Figure 1 and 2 in such a way that the number of records selected from each group is inversely proportional to the percentage of records in the original `#successfulPrediction` value groups. For instance, the number of records in the 0-5 `#successfulPrediction` value group of the KDD train set constitutes 0.04% of the original records, therefore, 99.96% of the records in this group are included in the generated sample. Tables III and IV show the detailed statistics of randomly selected records.

The generated data sets, KDDTrain$^+$ and KDDTest$^+$,

TABLE IV
STATISTICS OF RANDOMLY SELECTED RECORDS FROM KDD TEST SET

	Distinct Records	Percentage	Selected Records
0-5	589	0.76	585
6-10	847	1.10	838
11-15	3,540	4.58	3,378
16-20	7,845	10.15	7,049
21	64,468	83.41	10,694
Total	77,289	100.00	22,544

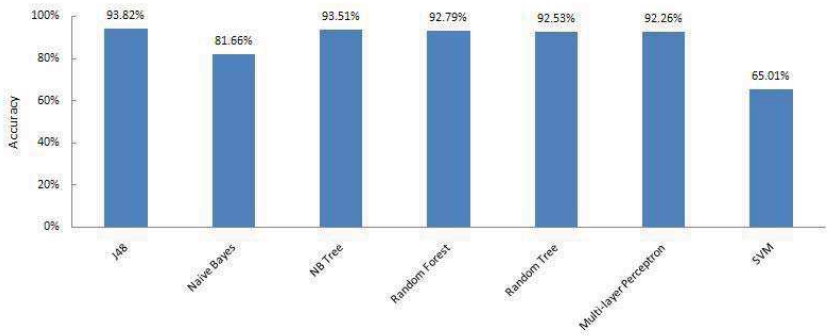

Fig. 3. The performance of the selected learning machines on KDDTest

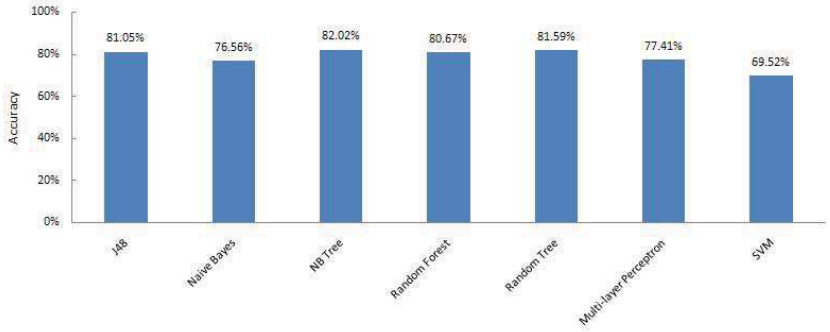

Fig. 4. The performance of the selected learning machines on KDDTest^{+}

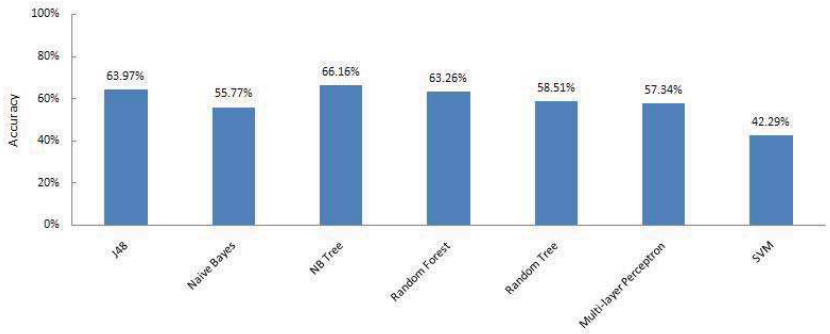

Fig. 5. The performance of the selected learning machines on KDDTest^{-21}

included 125,973 and 22,544 records, respectively. Furthermore, one more test set was generated that did not include any of the records that had been correctly classified by all 21 learners, KDDTest^{-21}, which incorporated 11,850 records. For experimental purposes, we employed the first 20% of the records in KDDTrain^{+} as the train set, having trained the learning methods, we applied the learned models on three test sets, namely KDDTest (original KDD test set), KDDTest^{+} and KDDTest^{-21}. The result of the evaluation of the learners on these data sets are shown in Figures 3, 4 and 5, respectively.

As can be seen in Figure 3, the accuracy rate of the classifiers on KDDTest is relatively high. This shows that the original KDD test set is skewed and unproportionately distributed, which makes it unsuitable for testing network-based anomaly detection classifiers. The results of the accuracy and performance of learning machines on the KDD'99 data set are hence unreliable and cannot be used as good indicators of the ability of the classifier to serve as a discriminative tool in network-based anomaly detection. On the contrary, KDDTest^{+} and KDDTest^{-21} test set provide more accurate information about the capability of the classifiers. As an example, classification of SVM on KDDTest is 65.01% which is quite poor compared to other learning

approaches. However, SVM is the only learning technique whose performance is improved on KDDTest$^+$. Analyzing both test sets, we found that SVM wrongly detects one of the most frequent records in KDDTest, which highly affects its detection performance. In contrast, in KDDTest$^+$ since this record is only occurred once, it does not have any effects on the classification rate of SVM, and provides better evaluation of learning methods.

VI. Concluding Remarks

In this paper, we statistically analyzed the entire KDD data set. The analysis showed that there are two important issues in the data set which highly affects the performance of evaluated systems, and results in a very poor evaluation of anomaly detection approaches. To solve these issues, we have proposed a new data set, NSL-KDD [24], which consists of selected records of the complete KDD data set. This data set is publicly available for researchers through our website and has the following advantages over the original KDD data set:

- It does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records.
- There is no duplicate records in the proposed test sets; therefore, the performance of the learners are not biased by the methods which have better detection rates on the frequent records.
- The number of selected records from each difficulty-level group is inversely proportional to the percentage of records in the original KDD data set. As a result, the classification rates of distinct machine learning methods vary in a wider range, which makes it more efficient to have an accurate evaluation of different learning techniques.
- The number of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.

Although, the proposed data set still suffers from some of the problems discussed by McHugh [4] and may not be a perfect representative of existing real networks, because of the lack of public data sets for network-based IDSs, we believe it still can be applied as an effective benchmark data set to help researchers compare different intrusion detection methods.

References

[1] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, "A taxonomy of computer program security flaws," *ACM Comput. Surv.*, vol. 26, no. 3, pp. 211–254, 1994.

[2] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," *Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM03)*, pp. 172–179, 2003.

[3] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Ocotber 2007.

[4] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262–294, 2000.

[5] S. J. Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K. Chan, "Cost-based modeling for fraud and intrusion detection: Results from the jam project," *discex*, vol. 02, p. 1130, 2000.

[6] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation," *discex*, vol. 02, p. 1012, 2000.

[7] MIT Lincoln Labs, 1998 DARPA Intrusion Detection Evaluation. Available on: http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/index.html, February 2008.

[8] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 3, pp. 186–205, 2000.

[9] J. Gaffney Jr and J. Ulvila, "Evaluation of intrusion detectors: A decision theory approach," in *Proceedings of IEEE Symposium on Security and Privacy, (S&P)*, pp. 50–61, 2001.

[10] G. Di Crescenzo, A. Ghosh, and R. Talpade, "Towards a theory of intrusion detection," *Lecture notes in computer science*, vol. 3679, p. 267, 2005.

[11] A. Cardenas, J. Baras, and K. Seamon, "A framework for the evaluation of intrusion detection systems," in *Proceedings of IEEE Symposium on Security and Privacy, (S&P)*, p. 15, 2006.

[12] G. Gu, P. Fogla, D. Dagon, W. Lee, and B. Skorić, "Measuring intrusion detection capability: An information-theoretic approach," in *Proceedings of ACM Symposium on Information, computer and communications security (ASIACCS06)*, pp. 90–101, ACM New York, NY, USA, 2006.

[13] M. Mahoney and P. Chan, "An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection," *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 220–238, 2003.

[14] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," *Proceedings of ACM CSS Workshop on Data Mining Applied to Security, Philadelphia, PA, November*, 2001.

[15] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38*, pp. 333–342, 2005.

[16] J. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[17] G. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, 1995.

[18] R. Kohavi, "Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, vol. 7, 1996.

[19] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] D. Aldous, "The continuum random tree. I," *The Annals of Probability*, pp. 1–28, 1991.

[21] D. Ruck, S. Rogers, M. Kabrisky, M. Oxley, and B. Suter, "The multilayer perceptron as an approximation to a Bayes optimaldiscriminant function," *IEEE Transactions on Neural Networks*, vol. 1, no. 4, pp. 296–298, 1990.

[22] C. Chang and C. Lin, "LIBSVM: a library for support vector machines," 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[23] "Waikato environment for knowledge analysis (weka) version 3.5.7." Available on: http://www.cs.waikato.ac.nz/ml/weka/, June, 2008.

[24] "Nsl-kdd data set for network-based intrusion detection systems." Available on: http://nsl.cs.unb.ca/KDD/NSL-KDD.html, March 2009.