ISSN 2395-1621

Detailed Analysis of Knowledge Discovery Data

[#1]Prof. Prashant Wakhare, [#2]Prof. Vinod Mehetre

[1]prashant_mitr@rediffmail.com
[2]vmehetre4@gmail.com

AISSMS IOIT, SPPU, India

ABSTRACT

With the ever increasing amount of new attacks in today's world the amount of data will keep increasing, and because of the base-rate fallacy the amount of false alarms will also increase. Another problem with detection of attacks is that they usually isn't detected until after the attack has taken place, this makes defending against attacks hard and can easily lead to disclosure of sensitive information. This paper is an analysis of 10% of KDD cup'99 training dataset based on intrusion detection. We have focused on establishing a relationship between the attack types and the protocol used by the hackers, using clustered data. Analysis of data is performed using k-means clustering; we have used the Oracle 10g data miner as a tool for the analysis of dataset and build 1000 clusters to segment the 494,020 records. The investigation revealed many interesting results about the protocols and attack types preferred by the hackers for intruding the networks.

ARTICLE INFO

Article History

Received: 16th May 2016

Received in revised form :

16th May 2016

Accepted: 21th May 2016

Published online :

26th May 2016

Keyword: **KDD 99 dataset, Clustering, K-means, Intrusion detection.**

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 few products using anomaly detection approaches, and practitioners still think that it is not a mature technology yet. 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. 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 is 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 website1. 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 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

The KDD training dataset consist of 10% of original dataset that is approximately 494,020 single connection vectors each of which contains 41 features and is labeled with exact one specific attack type i.e., either normal or an attack. Each vector is labeled as either normal or an attack, with exactly one specific attack type. Deviations from 'normal behavior', everything that is not 'normal', are considered attacks. [18] Attacks labeled as normal are records with normal behavior. A smaller version 10% training dataset is also provided for memory constrained machine learning methods. The training dataset has 19.69% normal and 80.31% attack connections. KDD CUP 99 has been most widely used in attacks on network. The simulated attack falls in one of the following four categories [9]:

1. Denial of Service Attack (DOS): In this category the attacker makes some computing or memory resources too busy or too full to handle legitimate request, or deny legitimate users access to machine. DOS contains the attacks: 'neptune', 'back', 'smurf', 'pod', 'land', and 'teardrop'.

2. Users to Root Attack (U2R): In this category the attacker starts out with access to a normal user account on the system and is able to exploit some vulnerability to obtain root access to the system. U2R contains the attacks: 'buffer_overflow', 'loadmodule', 'rootkit' and 'perl'.

3. Remote to Local Attack (R2L): In this category the attacker sends packets to machine over a network but who does not have an account on that machine and exploits some vulnerability to gain local access as a user of that machine. R2L contain the attacks: 'warezclient', ' multihop', 'ftp_write', 'imap', 'guess_passwd', 'warezmaster', 'spy' and 'phf'.

4. Probing Attack (PROBE): In this category the attacker attempt to gather information about network of computers for the apparent purpose of circumventing its security. PROBE contains the attacks: 'portsweep', 'satan', 'nmap', and 'ipsweep'.

The major objectives performed by detecting network intrusion are stated as recognizing rare attack types such as U2R and R2L, increasing the accuracy detection rate for suspicious activity, and improving the efficiency of real-time intrusion detection models. This detects that the training dataset consisted of 494,019 records, among which 97,277 (19.69%) were 'normal', 391,458(79.24%) DOS, 4,107 (0.83%) Probe, 1,126 (0.23%) R2L and 52 (0.01%) U2R attacks. Each record has 41 attributes describing different features and a label assigned to each either as an 'attack' type or as 'normal'.

The protocols that are considered in KDD dataset are TCP, UDP, and ICMP that are explained below:

TCP: TCP stands for "Transmission Control Protocol". TCP is an important protocol of the Internet Protocol Suite at the Transport Layer which is the fourth layer of the OSI model. It is a reliable connection-oriented protocol which implies that data sent from one side is sure to reach the destination in the same order. TCP splits the data into labeled packets and sends them across the network. TCP is used for many protocols such as HTTP and Email Transfer.

UDP: UDP stands for "User Datagram Protocol". It is similar in behavior to TCP except that it is unreliable and connection-less protocol. As the data travels over unreliable media, the data may not reach in the same order, packets may be missing and duplication of packets is possible. This protocol is a transaction-oriented protocol which is useful in situations where delivery of data in certain time is more important than loosing few packets over the network. It is useful in situations where error checking and correction is possible in application level.

ICMP: ICMP stands for "Internet Control Message Protocol". ICMP is basically used for communication between two connected computers. The main purpose of ICMP is to send messages over networked computers. The ICMP redirect the messages and it is used by routers to provide the up-to-date routing information to hosts, which initially have minimal routing information. When a host receives an ICMP redirect message, it will modify its routing table according to the message.

Various researchers have analysed the KDD Cup 99 Dataset using various methods, but K-means clustering algorithm using ODM has not yet been applied. This investigation makes use of this technique to analyse and study the pattern of attack types in relation with the protocols.

The remainder of the paper is organized as follows. Section 2 discusses the related work with regards to analyses of the KDD cup 99 dataset. Section 3 provides the methodology of the work. Results are reported in Section 4 and Conclusions are drawn in Section 5.

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.

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%

volumes of traffic, they are easily detectable by other means and there is no need of using anomaly detection systems to find these attacks.

III. PROCESS OF DATA

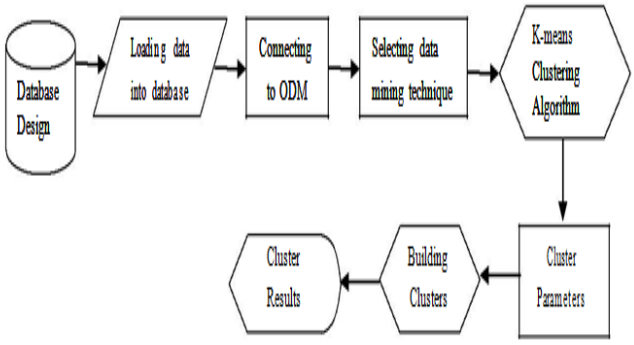

Fig 1. Modelling of KDD'CUP Data Mining Process

1. Database Design: The database has been designed using Oracle 10g Database. The source table named KDD contains 41 attributes. One primary key attribute has been added.
2. Loading data into database: The 10% of KDD 99 training dataset is a huge dataset having 494,020 records. The available dataset is in .txt format which was imported into Oracle Data Engine using SQL Loader.
3. Connecting to ODM: Oracle Data Miner 10g acts as a client to connect the ODM to Server. It requires certain system privileges from Server side; the system privilege utilized is ctxsys.ctx_ddl.
4. Selecting Data Mining Technique: We had adopted Clustering based Data Mining Technique to design the data model of KDD Cup Dataset.

   K-Means Clustering Algorithm: Clustering can be done in two ways. The first is K-Means and the second O-Means Algorithm. K-Means Algorithm is used in data pre- processing steps to identify homogenous group. Here we have used K-Means distance based algorithm with specific number of clusters.
6. Cluster Parameters: Deciding on the values of the following parameters number of cluster (k), distance function, split criterion, Maximum iterations, Number of Bins, Minimum Error Tolerance, minimum support, Block Growth.
7. Building Clusters: The clusters had been processed using ODM on the basis of desired cluster parameters.

8. Cluster Results: The results are displayed in the form of clusters along with the centroid value for each attribute.

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

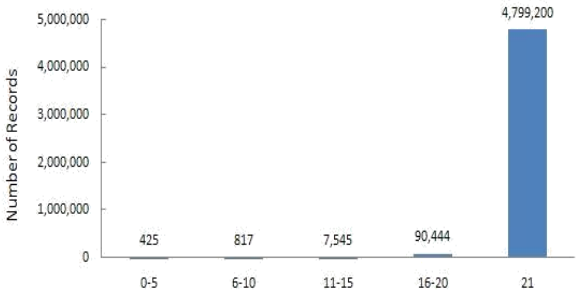

Fig. 2. The distribution of #successfulprediction values for the KDD data set records

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.

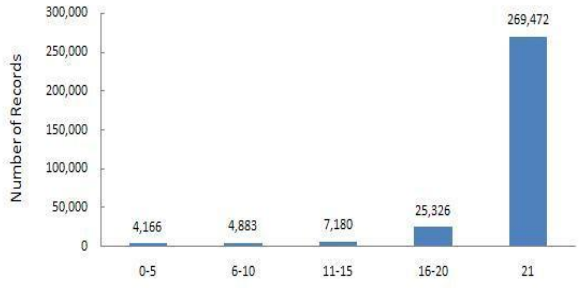

Fig. 3. The distribution of #successfulprediction values for the KDD data set records

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

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 #successful Prediction 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.

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

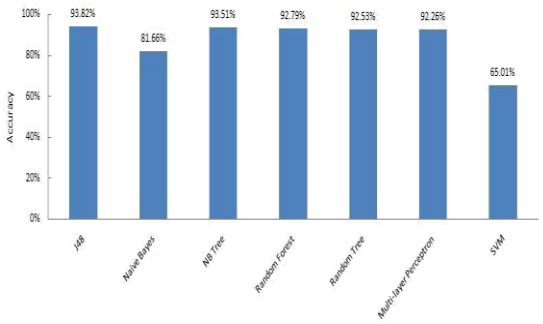

Fig. 4. The performance of the selected learning machines on KDDTest

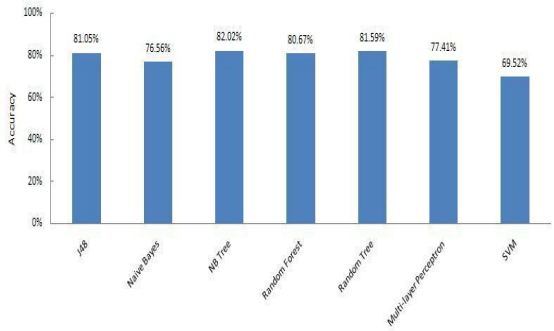

Fig. 4. The performance of the selected learning machines on KDDTest$^+$

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 accu-racy 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 clas-sifiers. 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 exper-iments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and com-parable.

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. Cun-ningham, 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 Intru-sion Detection Evaluation.Availableon: 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. Skoric,´ "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," LEC-TURE 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 optimaldiscrim-inant function," IEEE Transactions on Neural Networks, vol. 1, no. 4, 296–298, 1990.

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

[23] "Waikato environment for knowledge analysis (weka) version3.5.7."Availableon: 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/NSL-KDD/, March 2009.

