Svedberg Open
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Performance Comparison of J48 and Random Forest Algorithms for Network Intrusion Detection Using the KDD Cup 1999 Dataset

Pratik Jain[1], Archana Liladhar Rane[2], Madhav Mohan Vagmare[3], Rajni Jainwal[4], Dr. Nita Shinde[5], Javed Rauf Attar[6]

[1]Department of Computer Science & Engineering, IPS Academy, Institute of Engineering & Science, Indore, MP, Email id: pratikjain@ipsacademy.org, Scpous id: 60743637000
[2]Department of MCA, KK Wagh Institute of Engineering Education and Research, Nashik, Maharashtra, Email id: alrane@kkwagh.edu.in
[3]Vikrant University, Gwalior, Email id: madhav.vagmare@gmail.com
[4]Sage University, Indore, MP, Email id: rajnijainwal@gmail.com
[5]MET's Institute of Engineering, Adgaon, Nashik, Maharashtra, Email id: nitas_ioe@bkc.met.edu
[6]MET's Institute of Engineering, Nashik, Maharashtra, Email id: javeda_ioe@bkc.met.edu

ABSTRACT

The proliferation of computer networks, cloud services, Internet services, and electronic communications has led to an explosion of cyber threats in quantity and complexity. Attacks like Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), User-to-Root (U2R) are attacking the well-organized enterprises worldwide day by day, and these traditional security techniques like firewall, antivirus have limitations to sniff out the well sophisticated or zero-day attacks. As a result, Intrusion Detection Systems (IDSs) play a key role in contemporary cybersecurity frameworks by monitoring network traffic in real-time and detecting malicious activities.

Due to its ability to learn patterns from historical network traffic and classify normal and malicious network connections with high accuracy, machine learning has become a promising tool for enhancing intrusion detection. Amongst supervised learning algorithms, Decision Tree (J48) and Random Forest are some of the popular ones that are extensively applied based on its classifying capabilities, computational complexity and ease in utilizing the algorithm. Despite J48 producing a single decision tree and thus interpretable decision rules, Random Forest utilizes ensemble learning to average many decision trees to obtain more accurate predictions and less overfitting.

In this work, we compare the performances of classification algorithms J48 and Random Forest in detection of network intrusions based on the KDD Cup 1999 dataset. The experiments are performed in the data mining environment of WEKA 3. 8 under the same experimental setup to assure the compatibility of the study. The dataset is preprocessed to remove outliers and missing values before classification, and to enable the data set to be used as supervised learning materials. Both algorithms are tested against the conventional performance measure like classification accuracy, precision, recall, F-measure, ROC Area, confusion matrix, model building time and computational cost. In this work, we evaluate different classifiers and their combinations aiming at the best detection performance with a feasible training and testing time (computational feasibility), with the overall goal to recommend the classifier with best performance for practical intrusion detection systems. This comparative study enables a better understanding of the merits and demerits of both the algorithms, and also helps the researchers and cyber security experts for choosing a suitable supervised machine learning algorithm based on the required parameters for network intrusion detection.

Keywords: Intrusion Detection System (IDS), Network Security, Machine Learning, Supervised Learning, Decision Tree, J48 Algorithm, Random Forest, WEKA, KDD Cup 1999, Cybersecurity, Classification, Data Mining.

INTRODUCTION

Coupled with the increasing size of computer networks, the cloud computing, and the Internet-based services, more and more traffic appeared over the networks. Today, organizations use digital networks to communicate, do online banking, use cloud storage, conduct e-commerce, and share data. While these

technologies have led to productivity and accessibility improvements, they have also increased the quantity and sophistication of cyber threats. Attackers strive to exploit network vulnerabilities by launching Denial-of-Service (DoS) attacks, Probe attacks, Remote-to-Local (R2L) attacks, User-to-Root (U2R) attacks, virus, worm, ransomware, and other suspicious activities. These attacks can cause financial damage and lead to the leaking of sensitive information, as well as interrupting the essential service. Regular security practices such as firewalls, antivirus software, authentication systems and access controls serve as the initial barrier against cyber threats. However, these methods are based on predefined rules or attack signatures, and are too tedious to handle unknown or sophisticated attacks. Therefore, an Intrusion Detection System (IDS) has been recognized playing an important role in the security mechanism of modern information infrastructure. An IDS monitors the network traffic in real time and detects suspicious activities by examining the communication activities and it generates alerts if it detected any malicious activity. Because it can make computers learn patterns from past network data and help classify network activities with less human efforts, machine learning has attracted much attention in intrusion detection. In contrast to traditional rule-based methods, machine learning methods can capture intricate interactions between network attributes, and thus achieve higher detection performance. Machine learning algorithms applied to the intrusion detection field can be divided into supervised and unsupervised approaches. Supervised learning methods are trained on labeled data sets and can achieve high detection rates for normal and attack traffic. As a result, supervised classification methods have been the most widely used for network intrusion detection.

Signature-based NIDS

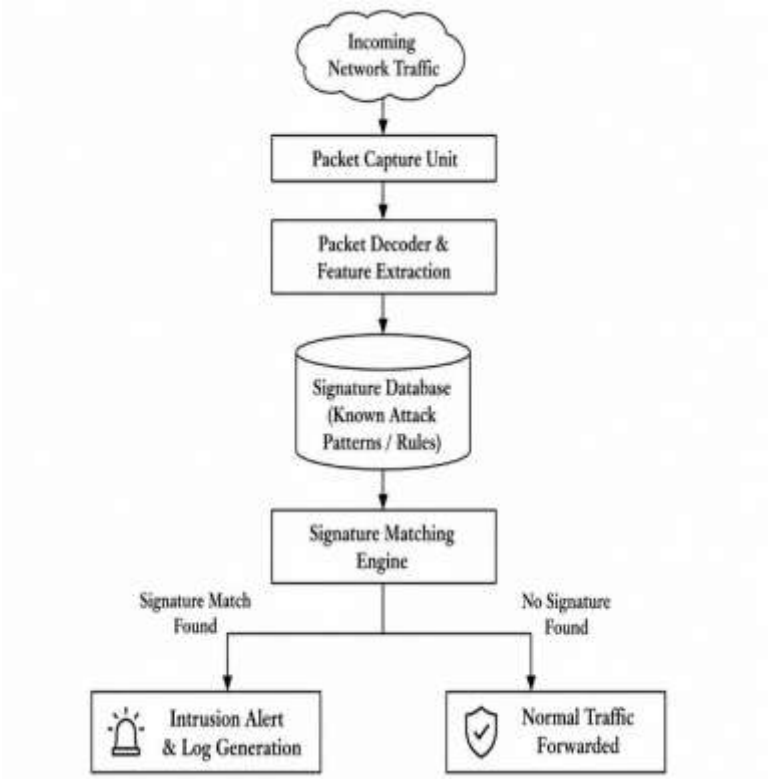


Fig 1 (b) Signature - Based NIDS diagram

Among the supervised learning algorithms, J48 Decision Tree and Random Forest are good choices as they are easy to implement, positive for their performance and fast in computational time. J48 builds a decision tree by choosing for each level the more informative features, which can be translated into easily understandable and interpretable classification rules. In contrast, Random Forest is an ensemble learning method that utilizes multiple decision trees to enhance prediction accuracy and avoid overfitting. Because of these properties, the two algorithms can be used with planning of intrusion detection applications where the classification of network traffic.

It has therefore become one of the standard benchmark datasets for testing intrusion detection algorithms. It consists of thousands of network connection entries, some of which represent normal operation and others an array of attacks, so is ideal for comparative analysis in a controlled environment. The existence of labelled data allows for researchers to assess supervised learning algorithms by conventional performance metrics, i.e., accuracy, precision, recall, F-measure, ROC Area and confusion matrix.

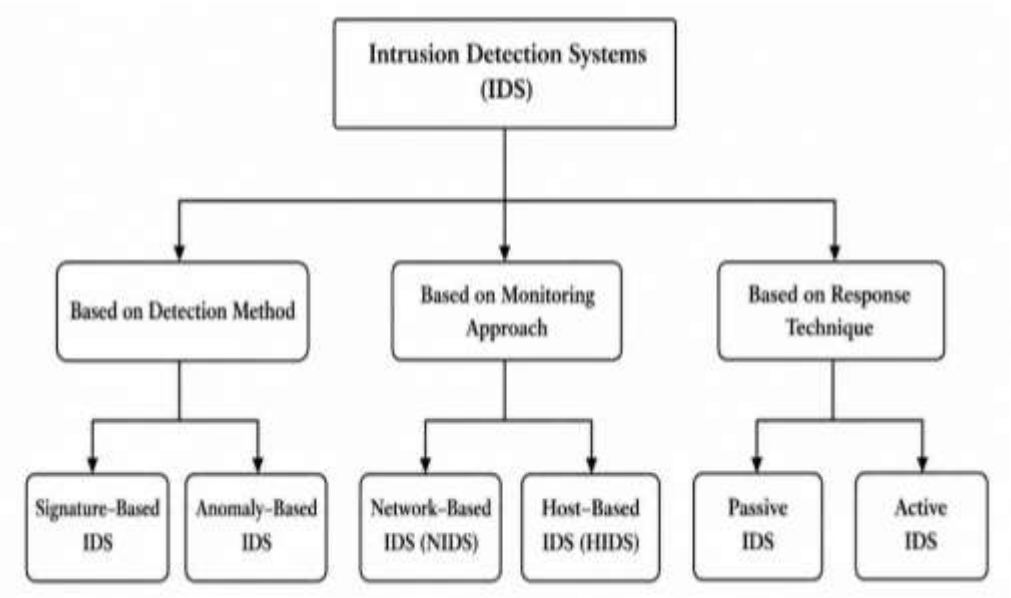


Fig 1 (b) Categories of intrusion detection systems

This study shows the comparative analysis of J48 Decision Tree and Random Forest algorithms for network intrusion detection with KDD Cup 1999 data set in WEKA 3.8 environment. The two algorithms are also compared using the same experiments setups. In this paper, we focus on their classification performance, computational cost and applicability to intrusion detection. [Result] It is anticipated that results of this work will enable researchers and cyber security professionals to choose a good supervised machine learning algorithm for building high-performance and robust IDS.

Literature Tracery:

Denning [1] was the first to present the anomaly-based intrusion detection model, which is also the basis of the typical IDS nowadays. Lee and Stolfo [2] developed a data mining-based framework for misuse detection by building features and classification models from network audit data. The KDD Cup 1999 dataset [3] is widely used as a benchmark dataset for testing machine learning algorithms in intrusion detection domain. Witten, Frank, Hall, and Pal [4] describe practical machine learning: their book is now one of the primary resources for learning WEKA and implementing classification algorithms. 5] introduced several enhancements, which added more sophisticated preprocessing, classification, clustering, and feature selection methods. C4.5 Decision Tree algorithm was introduced by Quinlan [6] which is the basis of J48 classifier in WEKA. Random Forest is an ensemble learning method for classification and regression which operates by constructing a multitude of decision trees during training time and outputting the class that is the mode of the classes of the individual trees. Mitchell [8] outlined instructional ideas of ML algorithm that could be employed to classify problems. Bishop [9] also described probabilistic pattern recognition and supervised learning methodologies that are commonplace in intelligent classification. Axelsson [10] described a comprehensive taxonomy of intrusion detection systems and subsequentally introduced a taxonomy to distinguish different IDS techniques. Tavallaee et al. [11] examined the KDD Cup 1999 benchmark and highlighted the problems of redundancy and class imbalance while they also proposed the NSL-KDD dataset as a better benchmark. The authors Moustafa and Slay [12] also presented the UNSW-NB15 dataset to address the drawbacks of previous intrusion detection datasets. Kendall [13] provided benchmarking attack datasets that facilitated testing IDS. García-Teodoro et al. [14] gave an overview of anomaly based intrusion detection methodologies and the benefits as well as research issues thereof. Patcha and Park [15] gave a high level overview on the types of approaches for anomaly detection in network security. Mukkamala, Janoski and Sung [16] studied the use of neural networks (NNs) and support vector machines (SVMs) in intrusion detection. Mell and Scarfone [17] released NIST recommendations on the use and assessment of Intrusion Detection and Prevention Systems (IDPS). Sommer and Paxson [18] focused on the the limits of applying machine learning to real-world intrusion detection systems. Core data mining tasks such as classification, feature selection, and predictive modeling are covered by Han, Kamber, and Pei [19]. They also provide an "accessible treatment of sophisticated statistical learning procedures that have influenced recent research in machine learning" (Hastie et al. [20]).

Witten and Frank [21] discussed practicalities of running the machine learning algorithms in WEKA for real applications. Frank, Hall, and Witten [22] also describe the WEKA Workbench for preprocessing and analysing data with supervised learning. Revathi and Malathi [23] compared the performance of different

machine learning techniques such as support vector machine, decision tree and K-nearest neighbor with intrusion detection using NSL-KDD dataset. McHugh [24] reviewed the limitations of the DARPA intrusion detection evaluation method. Bace and Mell [25] described the intrusion detection system architecture and capability for realistic applications. Roesch [26] created Snort, a lightweight network intrusion detection system that uses signature matching. Northcutt and Novak [27] addressed many of the practical issues involved in constructing and operating a network intrusion detection system. Kruegel and Vigna [28] introduced anomaly detection techniques to detect web-based attacks. Folino, Pizzuti, and Spezzano [29] presented an ensemble-based scheme for distributed intrusion detection (DID). Wang, Stolfo, and Li [30] explored the use of machine learning techniques in real-time intrusion detection. Abraham, Jain, Thomas and Han [31] described a cooperative soft computing intrusion detection system using intelligent learning model. Panda and Patra [32] studied the effectiveness of the Naïve Bayes classifier for a network intrusion detection and show its classificationality. Sabhnani and Serpen [33] evaluated multiple machine learning algorithms on the KDD intrusion detection data set. Barbara et al. [34] proposed ADAM where they use data mining for intrusion detection. Chebrolu, Abraham and Thomas [35] presented novel feature reduction and ensemble methods for improving intrusion detection accuracy. Liao and Vemuri [36] studied the use of K-Nearest Neighbor (KNN) classifier for intrusion detection. Lee, Fan, Miller, Stolfo and Zadok [37] introduced cost-sensitive learning methods to enhance performance of intrusion detection. Lazarevic et al. [38] reviewed network intrusion detection via anomaly detection. Mukherjee, Heberlein and Levitt [39] gave a relatively comprehensive early study of network intrusion detection architectures. Paxson [40] developed the Bro intrusion detection system for high-speed network monitoring. Debar, Dacier and Wespi [41] suggested a taxonomy of intrusion detection systems with detection methodologies and deployment strategies. Finally, Cannady [42] illustrated the use of artificial neural networks for misuse detection and suggested the promise of intelligent learning approaches in the next generation intrusion detection systems.

Research Gap:

A survey of the literature shows that a large amount of work has been done on the use of machine learning techniques for Intrusion Detection Systems (IDSs). Some works have also been devoted to increase the detection accuracy by means of supervised learning algorithms, while also a few works have analyzed clustering algorithms for anomaly detection using commonly known benchmark datasets including KDD Cup 1999, NSL-KDD, and UNSW-NB15. K-Means has enjoyed widespread popularity due to its ease of implementation and efficiency, however, the EM algorithm is also well known for its probabilistic clustering nature, and modeling complex data distributions. While several studies exist comparing these two algorithms in general, there is lack or a compressive study comparing these algorithms under the same experimental setup and using the same data and implementation platform. The previous works tend to vary in the preprocessing technique, parameter setting, performance evaluation, and testing scenario, so that it is not really feasible to have an objective judgment on which one suits better for anomaly-based intrusion detection. In addition, a lot of early work are focus on detection accuracy but pay little attention on computational complexity, execution time, convergence, property of clusterings. These are also more relevant in designing intrusion detection systems for large scale network installations as Rapid Analysis is important there. In that sense, a comparative performance analysis of K-Means and Expectation-Maximization with the KDD Cup 1999 dataset under the WEKA platform is still needed. The objective of this work is to fill this gap by running the two algorithms in the same setting and comparing their performance in terms of execution time, clustering quality, convergence behavior and overall computational complexity.

Problem Recognition:

However, with the ever-expanding sophis-tication of computer networks and Internet based services, organizations are now generating huge amounts of network traffic daily. Identifying legitimate network traffic from attack traffic has become more chal-lenging due to various and complex cyber threats nowadays. An efficient Intrusion Detection System (IDS) should be able to classify different types of attacks accurately with a low computational overhead and high processing speed. Hence, choosing the suitable machine learning classification algorithm is one of the critical criteria to enhance the performance of network intrusion detection systems. J48 Decision tree and Random Forest are commonly used supervised learning algorithms for intrusion detection. Based on information gain of input attributes, J48 builds a single decision tree and the convinient classification rules can be obtained. In comparison, Random Forest is an ensemble method based on the aggregation of many decision trees, achieving higher classification accuracy and avoiding overfitting. Although both methods have shown favorable results when applied to cybersecurity problems, they differ in terms of the classification technique, computational requirements, model building and prediction cabability. Many existing research have analyzed these algorithms separately, but when it comes to a comparative study using the same preprocessing method on the same benchmark dataset within the same development environment, studies are very few. Variations in preparing datasets, parameter configurations, and evaluating

the results can also make it difficult to decide which algorithm best suits real-world intrusion detection applications. In order to solve this problem, in this paper, a comparative study between J48 and Random Forest is presented with the well-known dataset KDD cup 1999 in WEKA 3.8 platform. The algorithms are assessed under the same experimental setting on the basis of well-known classification performance measurements, such as accuracy, precision, recall, F-measure, ROC area, confusion matrix, and time for model construction. This study is intended to discover which classifier offers superior detection rate on NSD while keeping reasonable computational cost for NIDS.

Research Objectives:

In this paper, we present the J48 Decision Tree, Random Forest and k-NN classifier for NID and also compare the results of three classifiers on the WEKA 3.8 platform with KDD Cup 1999 dataset.
The following are the specific objectives of the study:

1. To process KDD Cup 1999 dataset for classification through WEKA data mining tool.
2. To run J48 Decision Tree algorithm on the processed data set.
3. To run Random Forest classifier on the same data set with the same experimental set up.
4. To measure the classification performance of these two methods with some common measures of performances.
5. To compare the accuracy, precision, recall, F-measure and ROC Area between the two-class J48 and the Random Forest.
6. To investigate the CPU time and model building time associated with these two algorithms.
7. Identify the strengths and weaknesses of the J48 Decision Tree and Random Forest algorithms for detecting network intrusion.
8. Based on the experimental results, recommend the best fitted supervised machine learning algorithm for fast and reliable intrusion detection.

Proposed Methodology:

In this paper, we evaluate the efficiency of the two popular supervised machine learning algorithms, J48 Decision Tree and Random Forest for network intrusion detection using KDD Cup 1999 dataset. To guarantee a fair and unbiased comparison, both approaches are executed within the same experimental framework in the WEKA 3.8 data mining environment. The suggested approach has five stages: dataset collection, data preprocessing, classification, performance evaluation and comparative analysis.

Dataset Acquisition

The work makes use of the KDD Cup 1999 dataset, which is an established dataset with widespread use in intrusion detection research. The data set consists of network connection records, each of which represents a series of two-way communication between two entities on the network and is labeled as either normal traffic or one of several different types of cyberattacks. 41 input features describe each connection, and one output (class) attribute identifies the connection as either normal or as an attack of a specific category.
The dataset contains the following attack classes.

- Denial of Service (DoS)
- Probe
- Remote-to-Local (R2L)
- User-to-Root (U2R)

Data Preprocessing

Before the classification algorithms are applied, the dataset is cleaned and preprocessed to enhance the quality of data and to get more reliable classification results.
The preprocessing step includes:

1. Loading the dataset in WEKA.
2. Categorical attributes such as protocol_type,service,flag are treated using WEKAs internal preprocessing methods.
3. Remove duplicate or conflicting records if any.
4. Apply normalization on numerical attributes to prevent features with large value from dominating the clustering.
5. Make sure all the attributes are clusterable.

So far we have seen that these pre-processing steps play a role in the efficiency and robustness of clustering.

Clustering Algorithms
Two supervised ML algorithms are chosen to be compared.

A. J48 Decision Tree
The J48 algorithm is an implementation of the C4.5 algorithm in WEKA. It recursively partitions the data set until it "leafs out" the tree in classification rules, based on a tree structure, by choosing the most informative attributes using information gain. The resultant decision tree can be easily understood and it allows the efficient classification of the network traffic.
For this study:

1. Classification Algorithm = J48 Decision Tree
2. Training Method = 10-Fold Cross Validation
3. Pruning = Enabled (Default Settings of WEKA)
4. Implementation Tool = WEKA 3.8

The classifier divides the network connections into:

1. Normal Traffic
2. Attack Traffic

B. Random Forest
Random Forest is a nonlinear, ensemble-based learning algorithm. The prediction is finalized using majority voting among all the trees. This method reduces overfitting, becomes more generalizable for unseen network traffic, and better improves classification accuracy.
For this study:

1. Classification Algorithm = Random Forest
2. Number of Trees = WEKA Default Settings
3. Training Method = 10-Fold Cross Validation
4. Implementation Tool = WEKA 3.8

The classifier divides the network connections into:

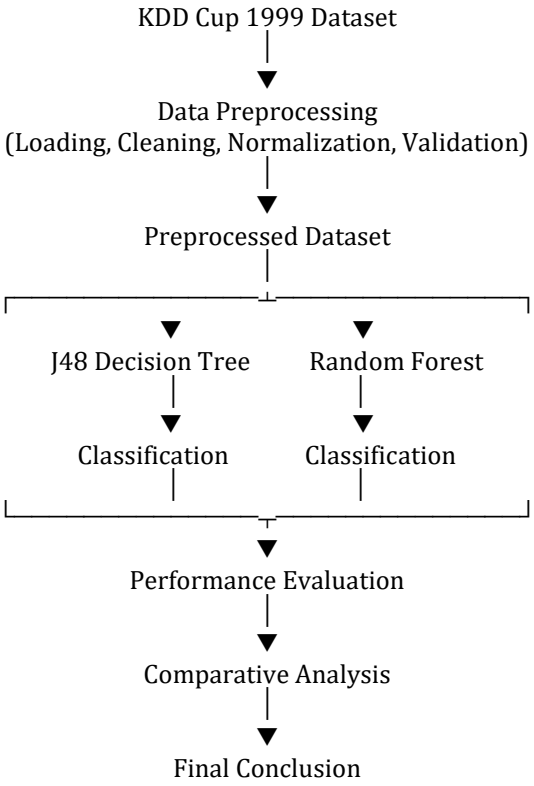
1. Normal Traffic
2. Attack Traffic Performance Evaluation

After executing both classification algorithms, their performance is evaluated using the following parameters:

- Classification Accuracy
- Precision
- Recall
- F-Measure
- ROC Area
- Confusion Matrix
- Model Building Time
- Computational Efficiency

Comparative Analysis
Finally, J48 Decision Tree and Random Forest algorithms are compared on the same experimental setting. The comparison divides into their classification accuracy, prediction power, computational cost, and model building time. From the experimental results produced by WEKA, the one that gives the best performance for network intrusion detection is identified.

Workflow of the Proposed Method



Methodology Summary

Phase	Description
Dataset Acquisition	Load the KDD Cup 1999 dataset into WEKA 3.8
Data Preprocessing	Clean, validate, and prepare the dataset for classification
J48 Classification	Train and evaluate the J48 Decision Tree classifier
Random Forest Classification	Train and evaluate the Random Forest classifier
Performance Evaluation	Measure accuracy, precision, recall, F-measure, ROC Area, confusion matrix, and model building time
Comparative Analysis	Compare both algorithms and determine the better-performing classifier

Experiments and Results:

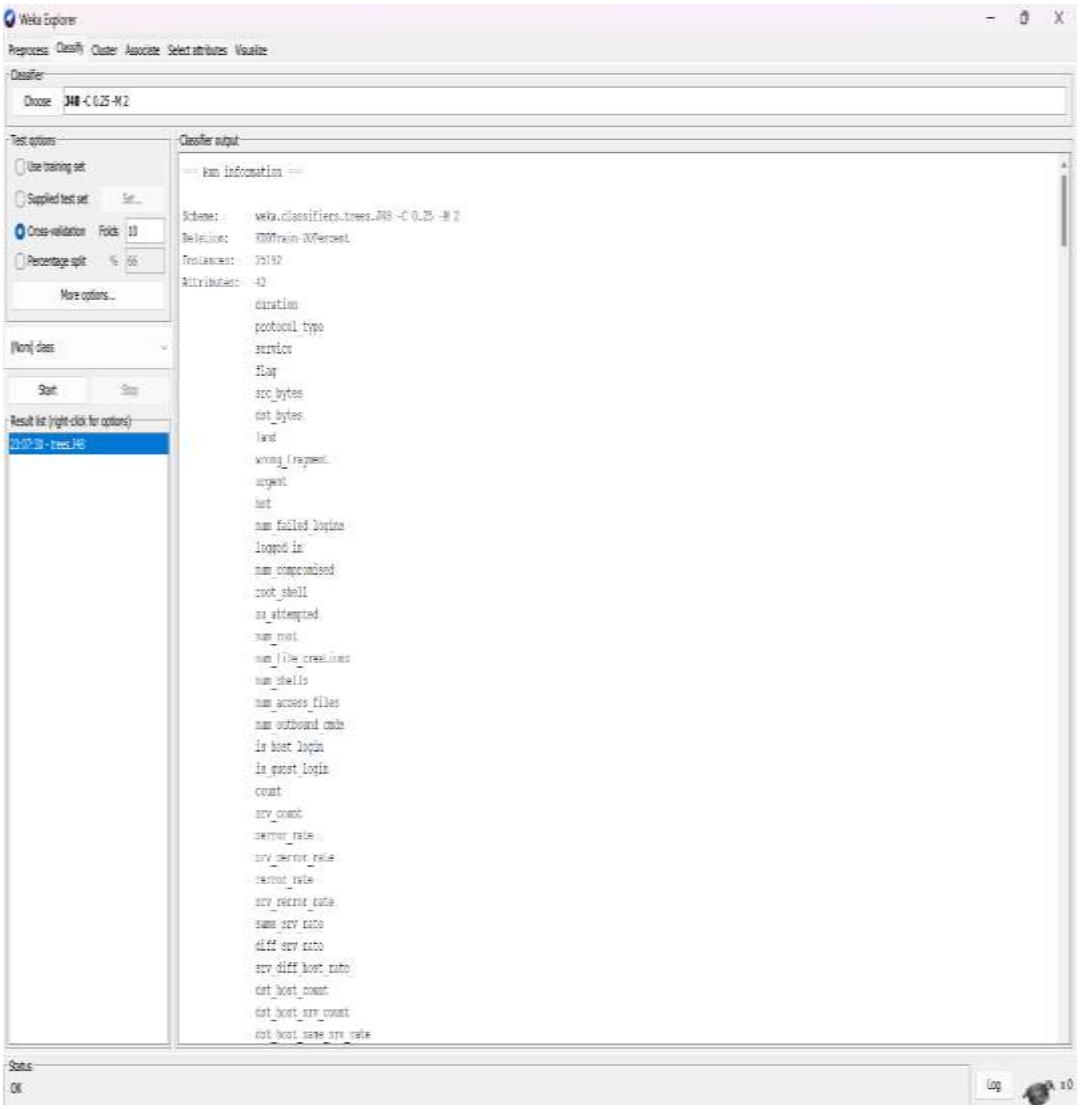
The experiments were performed on KDDTrain-20Percent data set which has 25192 instances having 42 features among them class attribute is one using data mining tool WEKA 3.8 [63]. Two popular supervised machine learning algorithms, J48 and Random Forest, were applied under same experimental configuration. To acquire credible and objective performances for both the classifiers, these were tested with 10-fold cross-validation. The performance metrics taken into account for this work are the classification accuracy, the precision, the recall, the F-measure, the ROC Area, the model building time and the confusion matrix.

A. Performance of J48 Decision Tree

The J48 classifier correctly classified 25,081 instances out of total 25,192, resulting total classification accuracy of 99.5594%. The model building took 0.97 seconds, which shows good compactional efficiency. The weighted average of precision, recall and f-measure was 0.996 and roc area was 0.998, which implies that the classification performance is very good. The confusion matrix reveals that indeed only 111 instances are misclassified, which underlines the accuracy of the J48 in separating normal and anomalous network traffic.

Table 1. Experimental Results of J48 Decision Tree

Parameter	Value
Dataset	KDDTrain-20Percent
Total Instances	25,192
Correctly Classified Instances	25,081
Incorrectly Classified Instances	111
Classification Accuracy	99.5594%
Precision	0.996
Recall	0.996
F-Measure	0.996
ROC Area	0.998
Model Building Time	0.97 sec



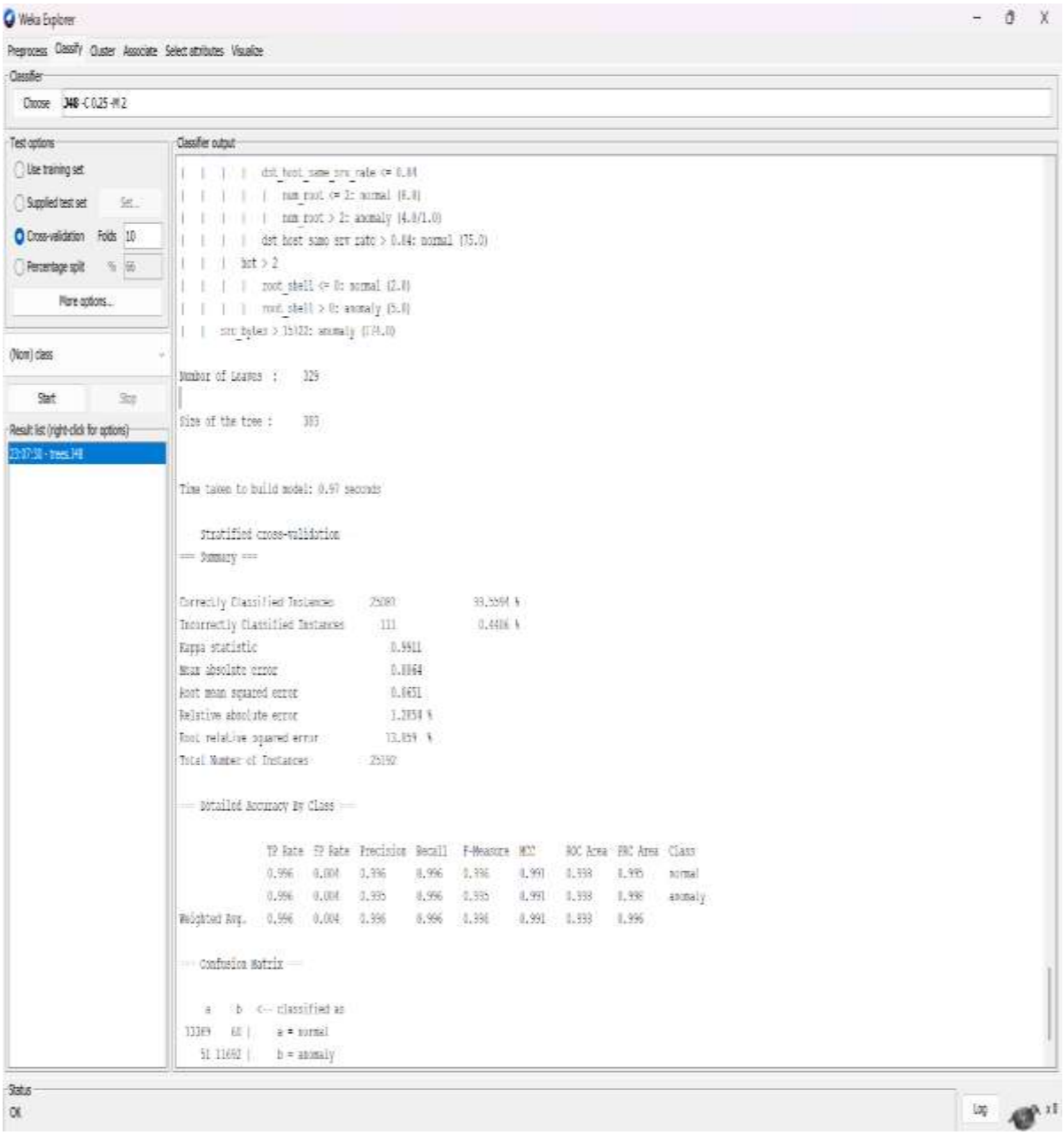


Figure 1.1 Performance of J48 Decision Tree

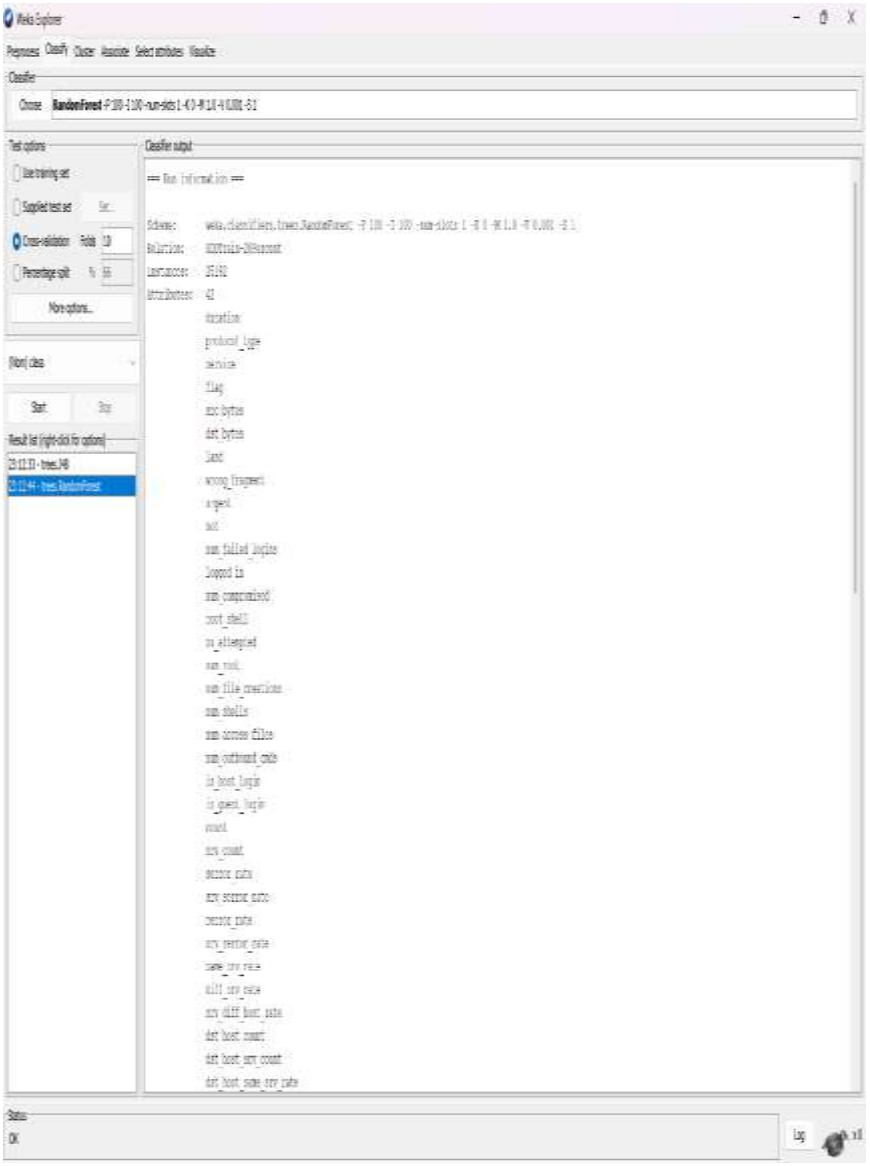
B. Performance of Random Forest

The Random Forest classifier outperforms other algorithms with the number of correctly classified samples 25,141 and the number of misclassified samples is just 51. The total classification accuracy was improved to 99.7976%, which is better than the J48 classifier. The weighted average precision, recall, and f-measure was 0.998, while the ROC Area was 1,000 indicating an almost perfect classification. Since ensemble learning builds multiple decision trees, the model training time 1.93 s is a little longer than J48.

Table 2. Experimental Results of Random Forest

Parameter	Value
Dataset	KDDTrain-20Percent
Total Instances	25,192
Correctly Classified Instances	25,141
Incorrectly Classified Instances	51
Classification Accuracy	99.7976%
Precision	0.998

Parameter	Value
Recall	0.998
F-Measure	0.998
ROC Area	1.000
Model Building Time	1.93 sec



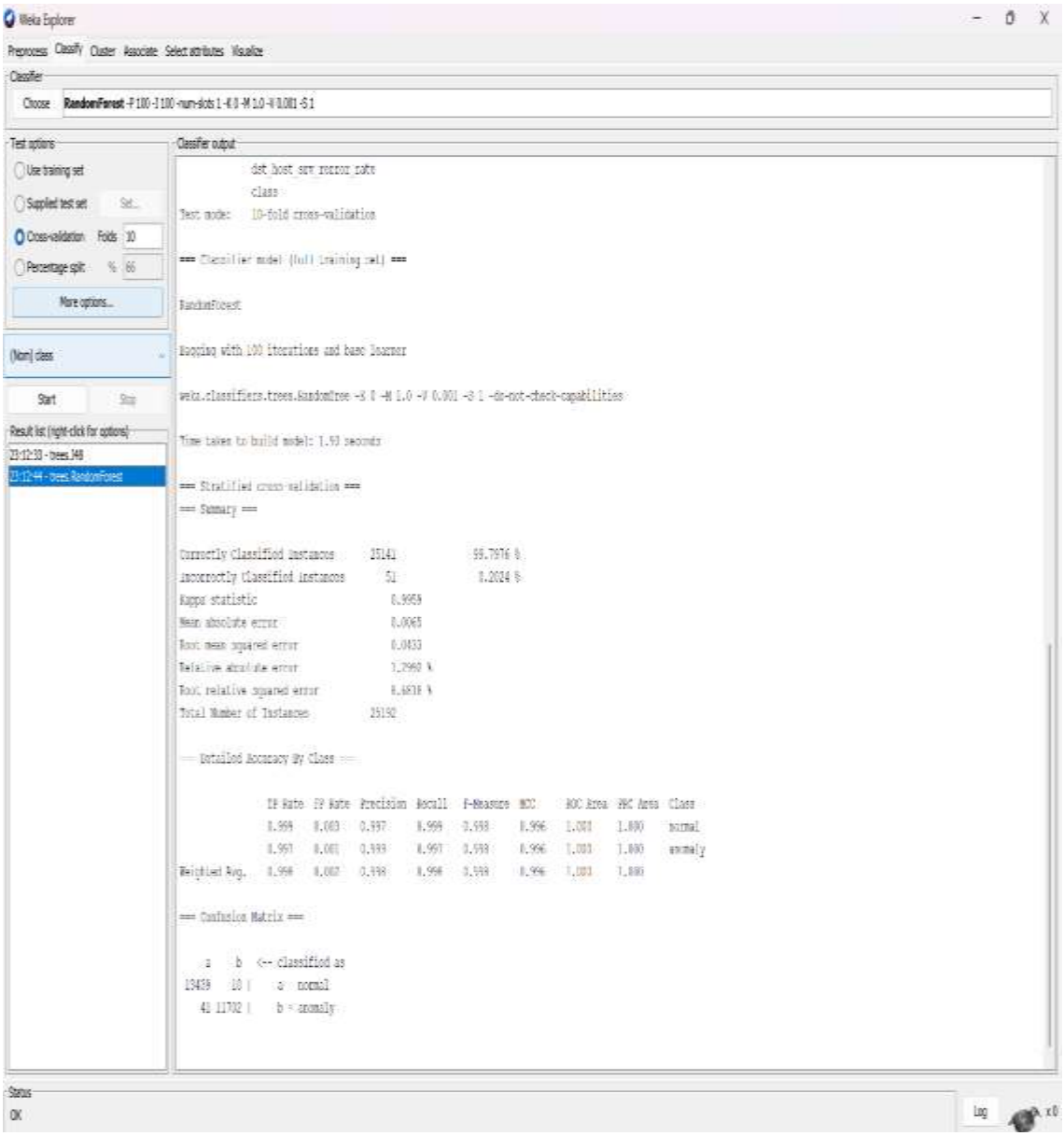


Figure 1.2 Performance of Random Forest

Performance Comparison

The results show that these two classifiers offer a good detection performance on KDDTrain-20Percent. Nonetheless, the Random Forest classifier performed better than the J48 Decision Tree with respect to the accuracy of the classification, precision, recall, F-measure, ROC Area, and the number of the correctly classified instances. Although the time to build the model with Random Forest was about twice as much as J48, the increase in computational cost is quite small when considering the great improvement in prediction performance. Hence, in terms of network intrusion detection, Random Forest can be regarded as a better classifier when compared with others for the experimental setup in this paper.

DISCUSSION

The performance evaluation shows that J48 Decision Tree and Random Forest may be good candidates to be considered in the ensemble for NID. J48 provides a faster model building process and decision rules are simpler which allows it to be used effectively in situations where the computational efficiency or the interpretability of the model are of importance. Unlike Random Forest, which exploits its ensemble learning strategy to achieve higher classification accuracy, less misclassified objects and nearly perfect ROC Area. Although Random Forest takes a bit longer to build than some of the other models, the extra computational expense is worth it for the superior predictions. From the results, it is clear that Random Forest is more appropriate algorithm for accurately detecting the network intrusions in KDDTrain-20Percent dataset in WEKA environment.

References

1. D. E. Denning, "An Intrusion Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.
2. W. Lee and S. J. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 227–261, Nov. 2000.
3. KDD Cup 1999 Data, UCI KDD Repository, University of California, Irvine, 1999.
4. I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4th ed. Burlington, MA, USA: Morgan Kaufmann, 2017.
5. M. A. Hall et al., "The WEKA Data Mining Software: An Update," ACM SIGKDD Explorations Newsletter, vol. 11, no. 1, pp. 10–18, 2009.
6. J. R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA, USA: Morgan Kaufmann, 1993.
7. L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
8. T. M. Mitchell, Machine Learning. New York, NY, USA: McGraw-Hill, 1997.
9. C. M. Bishop, Pattern Recognition and Machine Learning. New York, NY, USA: Springer, 2006.
10. S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," Technical Report, Chalmers University of Technology, Sweden, 2000.
11. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD Cup 99 Dataset," in Proc. IEEE CISDA, 2009, pp. 1–6.
12. N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," Military Communications and Information Systems Conference, 2015.
13. K. Kendall, "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems," MIT Lincoln Laboratory, 1999.
14. P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges," Computers & Security, vol. 28, no. 1–2, pp. 18–28, 2009.
15. A. Patcha and J. M. Park, "An Overview of Anomaly Detection Techniques," Computer Networks, vol. 51, no. 12, pp. 3448–3470, 2007.
16. S. Mukkamala, G. Janoski, and A. Sung, "Intrusion Detection Using Neural Networks and Support Vector Machines," Proc. IEEE IJCNN, 2002.
17. K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," NIST Special Publication 800-94, 2007.
18. R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," IEEE Symposium on Security and Privacy, 2010.
19. J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann, 2012.
20. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2nd ed. Springer, 2009.
21. I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Morgan Kaufmann, 2011.
22. E. Frank, M. Hall, and I. H. Witten, The WEKA Workbench. Morgan Kaufmann, 2016.
23. S. Revathi and A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques," International Journal of Engineering Research & Technology, 2013.
24. J. McHugh, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection Evaluations," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000.
25. R. Bace and P. Mell, "Intrusion Detection Systems," NIST Special Publication, 2001.
26. M. Roesch, "Snort—Lightweight Intrusion Detection for Networks," Proc. LISA, 1999.
27. S. Northcutt and J. Novak, Network Intrusion Detection. New Riders Publishing, 2002.
28. C. Kruegel and G. Vigna, "Anomaly Detection of Web-Based Attacks," Proc. ACM CCS, 2003.
29. G. Folino, C. Pizzuti, and G. Spezzano, "An Ensemble-Based Evolutionary Framework for Coping with Distributed Intrusion Detection," Genetic Programming and Evolvable Machines, 2005.
30. K. Wang, S. J. Stolfo, and B. Li, "Towards Real-Time Intrusion Detection Using Machine Learning Approaches," IEEE Workshop on Information Assurance, 2004.
31. A. Abraham, R. Jain, J. Thomas, and S. Han, "D-SCIDS: Distributed Soft Computing Intrusion Detection System," Journal of Network and Computer Applications, 2007.
32. M. Panda and M. R. Patra, "Network Intrusion Detection Using Naïve Bayes," International Journal of Computer Science and Network Security, 2007.
33. M. Sabhnani and G. Serpen, "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset," Proc. IEEE SMC, 2003.
34. D. Barbara, J. Couto, S. Jajodia, and N. Wu, "ADAM: Detecting Intrusions by Data Mining," IEEE Workshop on Information Assurance, 2001.
35. S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature Deduction and Ensemble Design of Intrusion

Detection Systems," Computers & Security, vol. 24, no. 4, pp. 295–307, 2005.
36. Y. Liao and V. R. Vemuri, "Use of K-Nearest Neighbor Classifier for Intrusion Detection," Computers & Security, vol. 21, no. 5, pp. 439–448, 2002.
37. W. Lee, W. Fan, M. Miller, S. Stolfo, and E. Zadok, "Toward Cost-Sensitive Modeling for Intrusion Detection," Journal of Computer Security, 2002.
38. A. Lazarevic et al., "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," Proc. SIAM ICDM, 2003.
39. B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network Intrusion Detection," IEEE Network, vol. 8, no. 3, pp. 26–41, 1994.
40. V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time," Computer Networks, vol. 31, pp. 2435–2463, 1999.
41. H. Debar, M. Dacier, and A. Wespi, "Towards a Taxonomy of Intrusion Detection Systems," Computer Networks, vol. 31, pp. 805–822, 1999.
42. J. Cannady, "Artificial Neural Networks for Misuse Detection," National Information Systems Security Conference, 1998.