ISSN 2278-3083
Volume 12, No.5, September - October 2023
International Journal of Science and Applied Information Technology
Available Online at http://www.warse.org/ijsait/static/pdf/file/ijsait031252023.pdf
https://doi.org/10.30534/ijsait/2023/031252023

Machine Learning-Based Intrusion Detection System for Cyber Attacks in Private and Public Organizations

Datti Useni Emmanuel[1], Justice Gokir Ali[2], Bilshak Yakubu[3], Atiku Baba Shidawa[4], Goteng Kuwunidi Job[5] & Mustapha Abdulrahman Lawal[6]
[1,2,3]Computer Department, Federal College of Education, Nigeria.
[4]National Institute for Policy and Strategic Studies (NIPPS), Kuru Plateau State, Nigeria.
[5]Computer Science Department, Plateau State Polytechnic Barkin Ladi, Nigeria.
[6]Department of Data Management, National Center for Remote Sensing Jos, Nigeria.

Received Date : September 03, 2023 Accepted Date : September 28, 2023 Published Date : October 07, 2023

ABSTRACT

Cyber-attacks have proven to be a force for hacking groups and state-sponsored organizations seeking to level the playing field with competitors. The hacker threat paired with the enormously hazardous and costly danger of fraud or intellectual property theft by insiders has created a volatile situation in private and public organizations. While a majority of internal breaches are due to employee negligence or human error, attacks by malicious insiders with access to sensitive company information have increased dramatically in recent years. Threats of financial loss, theft of sensitive information, and destruction to critical sectors have made cybersecurity a top security priority around the globe. Whereas the increase in frequency and complexity of attacks on the industry has increased the danger of being unprepared, it also has influenced the cost of preventing and recovering from cyber-attacks. To construct a machine learning bases instruction detection system is capable of detecting Cyber-attacks in the private and public sectors in Nigeria and the whole world. The results show that Random Forest and Random Tree algorithms outperform the other algorithms in their level of precision and F-measure as they are above 99% and 98% respectively, while the Random Forest outperforms the others by its detection rate. However, the Random Forest and Random Tree algorithms are more efficient in performing classification exercise on the Test datasets

Key words: Intrusion Detection System, Cyber Attacks, private and public organizations, Host-based Intrusion Detection System, Network Intrusion Detection System.

1. INTRODUCTION

Cyber-attack is an effort by hackers to damage or terminate a computer network or system for purposes of mischief, fraud, and/or hedonism. To say that the incidences of cyber-attack are increasing swiftly in Nigeria is not only an understatement but also a platitude. From the organized private sector to public service, hackers have not spared any entity. Intrusion Detection Systems (IDSs) play a key role in inert defense [1] targeting to detect malicious actions in different application areas such as the ones described in [2] and [3]. IDSs have been deployed in concurrence with active defense systems, such as honeypots. Two well-known approaches exist in IDS research which are namely: Host-based Intrusion Detection System (HIDS) and Network Intrusion Detection System (NIDS). The first method monitors the target machine's network interfaces and configurations, requiring specific settings attuned to the host machine as described by [4]. For instance, Microsoft Windows has different Operating system configurations in comparison to Linux-based systems, such as log files and OS calls. In contrast to the host-based activity, a NIDS monitors all incoming and outgoing packets on the computer network and is designed upon signature- and anomaly-based approaches. By 2019, the cost to the global economy due to cybercrime is projected to reach $2 trillion as reported by Juniper Networks. Among the contributory felonies to cybercrime is intrusions, which is defined as illegal or unauthorized use of a network or a system by attackers [5], an intrusion detection system (IDS) is used to identify the said malicious activity.

Most research conducted for IDS are traditional (signature) methods and expert rules rules-based methods are not efficient and too tedious because it involves manual procedures making the methods not sufficient [6] hence the introduction of machine learning techniques, here the procedures are automated.

1.1 Objectives of the Research

One of the objectives of this paper is to review the rate of cyber-attacks in public and private organizations in Nigeria, also to develop an intrusion detection system in public and private organizations for classifying the classes of attacks on different machine learning techniques.

2. LITERATURE REVIEW

2.1 Cyber-Attacks in Nigerian Organizations

Cyber-attacks unswerving in Nigeria are more than in any other country in Africa. World ranking in cyber-attack indicate that Nigeria is on top of the list after the United States and Britain but first in Sub-Saharan Africa [7]. Documented cases of cyber-attacks most widespread in Nigeria include yahoo attack, hacking, software piracy, pornography, credit card or ATM fraud, denial of service attack, internet relay chat (IRC) crime, virus dissemination, phishing, cyber plagiarism, spoofing, cyberstalking, cyber defamation, salami attack and cyber terrorism [8]. Indeed, Nigeria which boasts of a 29% internet penetration rate, 40 million internet users as of 2013, and a projected 70 million users in 2015, the highest in Africa, has suffered for years from cyber-related crimes [9]. According to Isaac (cited in the Guardian Nigeria, 2013), Nigeria as a fast-emerging market risks higher foreign invasion of cyber-attacks because of the glut in capacity utilization. It is this influx of foreign investors into the country and opportunities that result from such that puts the country in the international sport light in contemporary cyber-related crimes. Intrusion detection systems offer organizations several benefits, starting with the ability to identify security incidents. An IDS can be used to help analyze the quantity and types of attacks; organizations can use this information to change their security systems or implement more effective controls. An intrusion detection system can also help companies identify bugs or problems with their network device configurations. These metrics can then be used to assess future risks.

2.2 Machine Learning

Machine learning is the study of algorithms that improve their performance with experience and are meant to computerize exercises; the machine takes every necessary step exceptionally furthermore in a maintained way. It is a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed [10]. It includes various learning techniques classified as supervised, unsupervised, and reinforcement learning depending on the presence or the absence of labeled data. Supervised learning trains the program with labeled samples; thereby the trained program can predict similar unlabeled samples. It includes Prediction, Knowledge extraction, and Compression tasks. Unsupervised learning doesn't have any training samples; it uses the statistical approach of density estimation. Unsupervised learning works by the principle of finding the hidden design of the data by clustering or grouping data of a similar kind. It includes works like Pattern Recognition and Outlier Detection. Reinforcement learning is focused on software agents that need to take action in an environment so that it maximizes cumulative reward [11]. Each step of the agent is not considered individually for success or failure but on a sequence of actions taken together should have a direction towards good policy.

3. PROPOSED SYSTEM

In this section we present the proposed system in figure below consisting of six major parts to form the ML system: Dataset, Preprocessing, Machine Learning classifiers and detection and classification. The proposed system is an architecture proposed on testing to compare the four different algorithms that Bayes Net, J48, Random Forest and Random Tree.

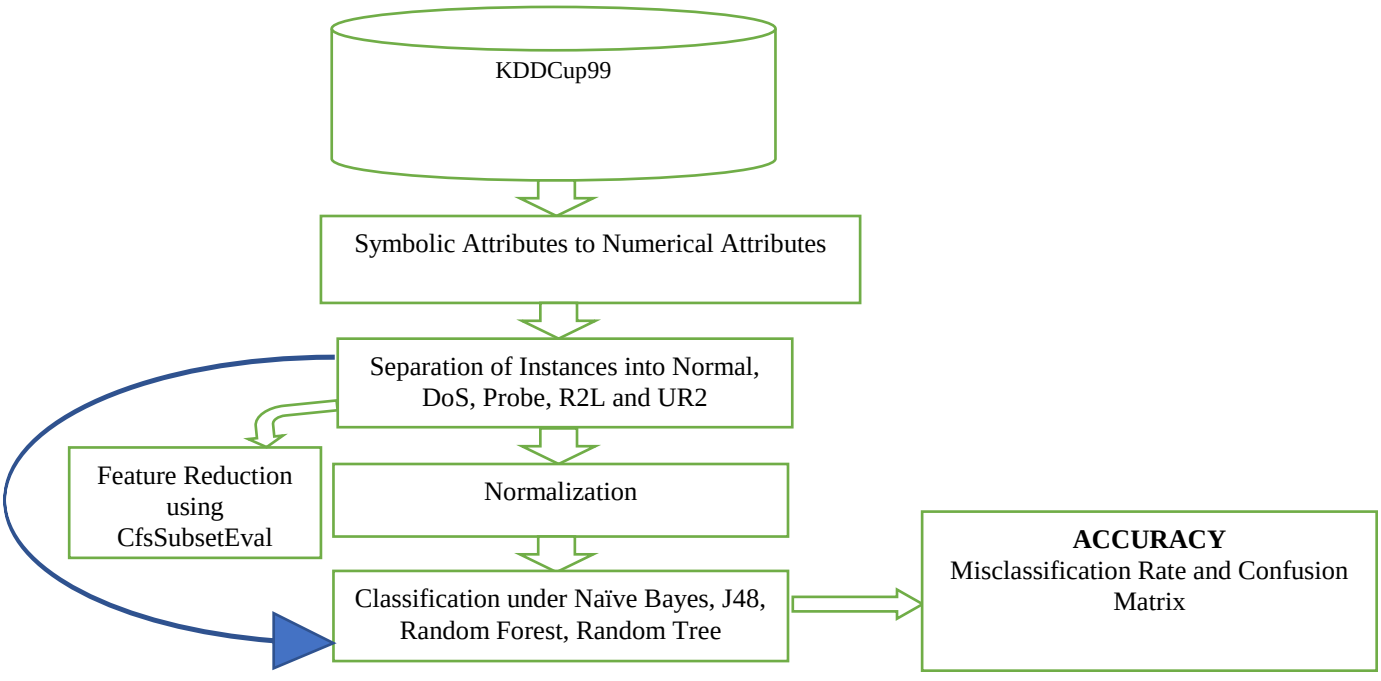


Figure 1: Architecture of the proposed machine learning system

3.1 Dataset

Dataset is a collection of data. Dataset corresponds to the contents of a single database table where every column of the table represents a particular variable and each row corresponds to a given member of the dataset in question. KDDCUP99 created by DARPA in 1998 consisted of 4,900,000 connections, each connection consists of 41 attributes and labels for this type of attack and are divided into four categories, namely attacks Denial of Services (DoS), Probe / Scan, Remote to User (R2L) and User to Root (U2R). KDDCUP99 a dataset that is extensively used for training as well as to evaluate the performance of IDS implemented by researchers.

3.2 Preprocessing

Preprocessing is a technique that is used to convert the raw data into a clean dataset. In other words, whenever the data is gathered from different sources it is collected in raw format which is not is not feasible for the analysis. Because the existing data in the database is composed by numeric and text, then Normalization was performed to convert it to numeric forms. As in protocol_type attributes, tcp to 0, udp to 1 and icmp to 2, then the attack attribute name each layer consists of two classes, 0 for normal, 1 to attack and for other attributes also done the same thing. There are several attributes that have very large numeric data, so it is necessary to scale, duration attribute (0-60000) was changed to (0.0-4.99), attributes src_bytes (0-693376000) were changed to (0.0-9.9), dst_bytes (0- 5204000) changed to (0.0-9.99).

3.3 Training

In this thesis KDDCup99 dataset was used consisting of 125973 instances with 42 attributes then grouped into four categories attack and used all the attributes of a dataset. The dataset was trained on the following (Bayes Net, J48, Random Forest and Random Tree). The table 3.2 below gives a detailed description of training attacks on training data.

Table 1: List of Training attack on Training data

DoS	Probe	R2L	U2R
Back	ipsweep	ftp_write	buffer_overflow
Land	nmap	guess_passwd	loadmodule
Neptune	portsweep	Imap	Perl
Pod	satan	multihop	Rootkit
Smurf		phf	
Teardrop		spy	

A. Symbolic Attributes to Numerical Attributes:

After, labeling Pre-processing is done to convert nominal attribute to binary attribute. In order to obtain improved performance of intrusion detection system, non-numeric features get removed.

B. Separation to instances:

Comparative analysis will be done between SVM and Naïve Bayes for classification of dataset, to analyze their accuracy and Misclassification Rate. At first raw dataset will be taken and the class attribute contains 24 different types of attack which get labeled under 4 categories. They are normal, Dos, Probe, r2l.

C. CfsSubsetEval:

is one of the methods of attribute selection. It calculates the value of attributes by considering the individual predicting estimation of all features along with the degree of redundancy between them.

D. Normalization:

In order to get different result and to improve the performance of the two datasets, methodologies like CfsSubsetEval is done for feature reduction. The given dataset after preprocessing under goes feature reduction and normalization.

E. Classification:

About classification under SVM, it comes under supervised learning method, in which various types of data from different subjects get trained. In a given high dimensional space, Support Vector Machine creates hyperplane or multiple hyperplanes in a high dimensional space. SVM creates hyperplane or multiple hyperplanes. The hyperplane which optimally separates the given data into various classes with the major partition, consider as a best hyperplane. For evaluate the margins between hyperplanes, a non-linear classifier applies various kernel functions. Maximizing margins between hyperplanes is the main aim of these kernel functions like linear, polynomial, radial basis, and sigmoid. Same, process is done using Naïve Bayes. Bayesian classifiers are statistical classifiers. They are capable to forecast the probability that whether the given model fits to a particular class. It is based on Bayes' theorem. It works on the hypothesis that, for a given class, the attribute value is independent to the values of the attributes. This theory is called class conditional independence. Other classifiers used includes J48, Random Forest and random tree.

F. Accuracy:

The accuracy and Misclassification rate will be taken as evaluation metrics.

4. IMPLEMENTATION AND RESULTS

In this section, this paper described the analysis of the learned representations in intrusion detecting of classes of attack using different machine learning techniques (Bayes Net, J48, Random Forest and Forest Tree) and compare the performance of all the classifiers with the existing system and other systems. The dataset used in this thesis is an open source, downloaded from the KDD website. The table 4.1 below shows the distribution of records in different classes for testing dataset used in the experiments.

Table 2: Distribution for test Dataset

Attack Category	Number of Samples
DoS	100776
R2L	4900
U2R	350
Probe	10042
Normal	238729
Total	354,797

The experiment was carried on WEKA 3.9.4 an open-source machine learning scripting software.

4.1 Loading Kddcup99 Train Dataset

The figure 2 below shows the initial process of data training by loading the dataset in WEKA machine learning environment. After and loading and training the dataset; which contains 125973 instances and 42 attributes.

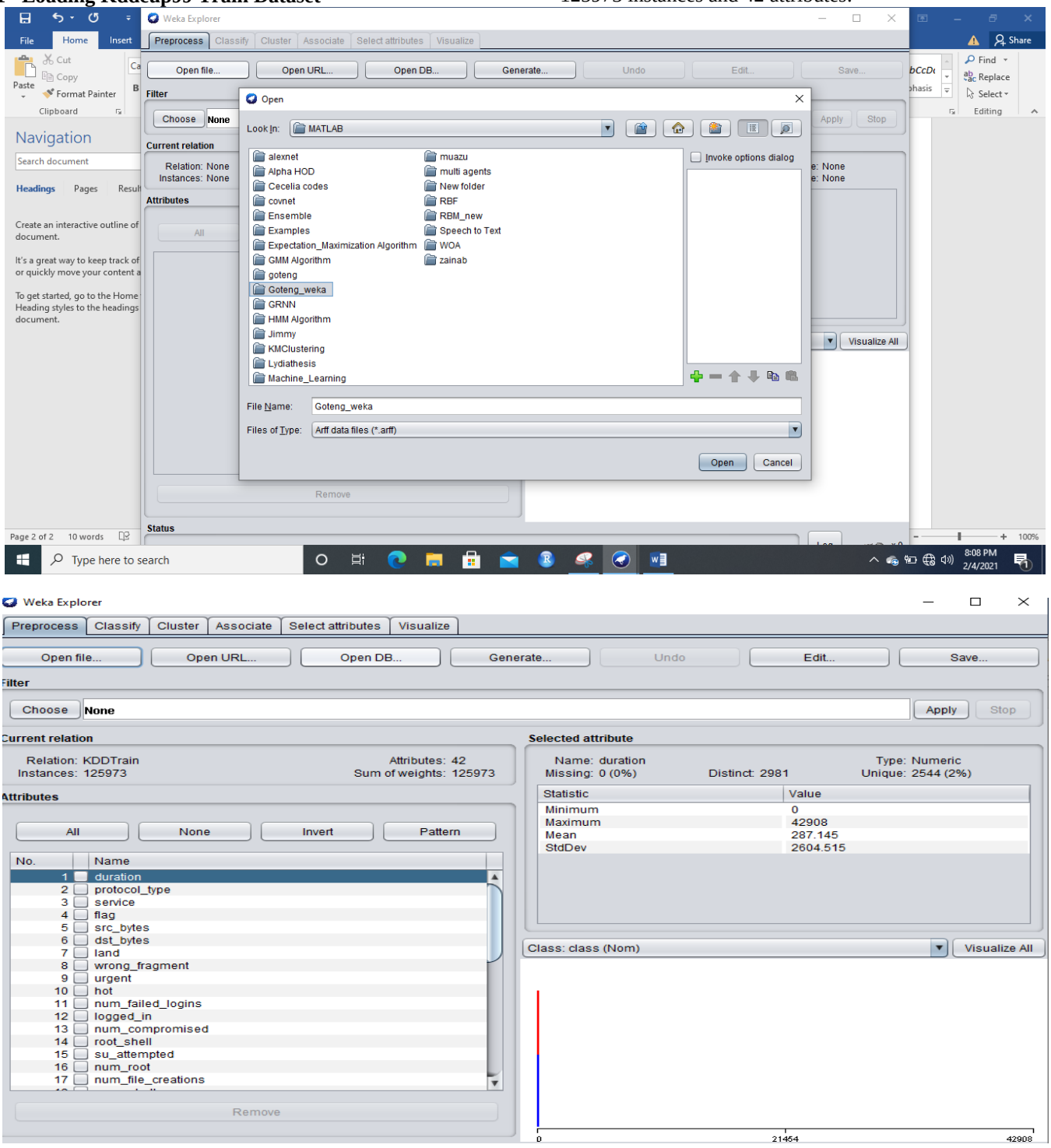


Figure 2: Loading of KDDCup99 dataset in WEKA

4.1 Output Results of the Four Classifier

Below is the output of all the four classifiers. The output is further divided into subsections for better understanding on how the trend continues.

A. Bayes Net Classifier

The figure 3 shows the output of the Bayes Net Classifier with 122426 Correctly Classified Instances (97.2%) and 3547 Incorrectly Classified Instances (2.82%).

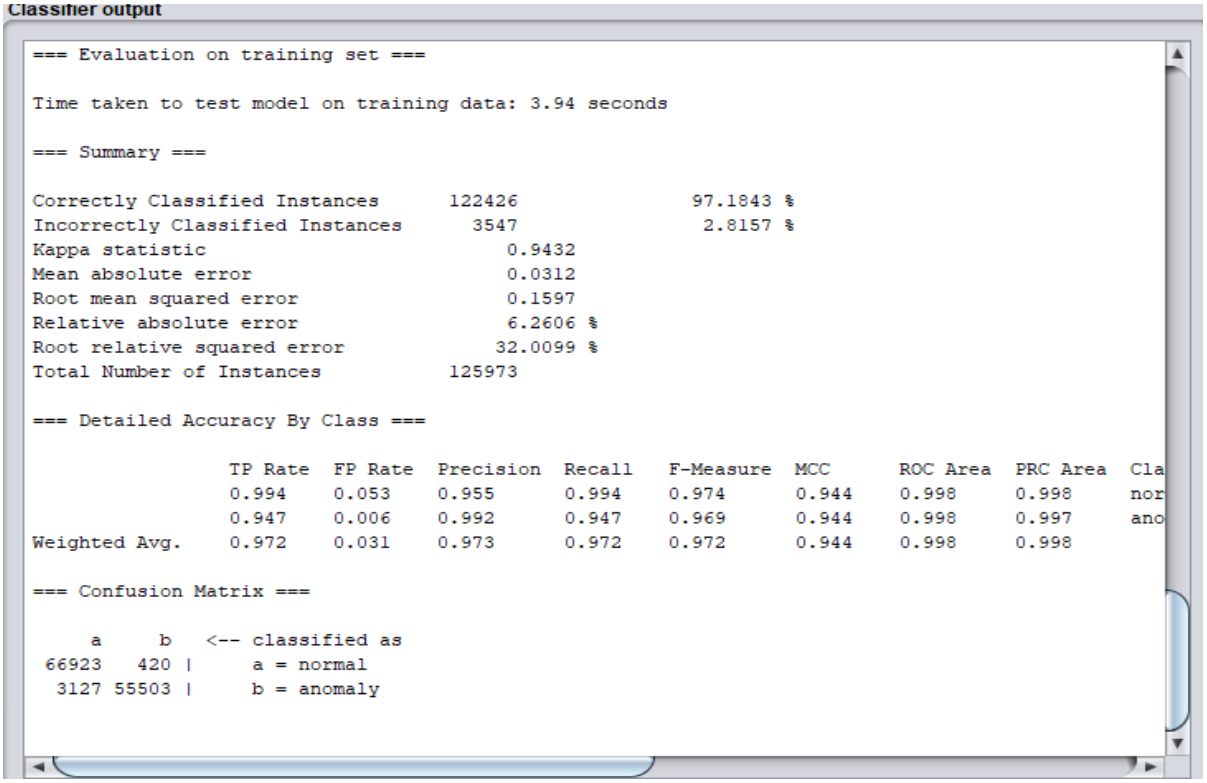


Figure 3: Output of Bayes Net Classifier

B. J48 Classifier

The figure 4&5 shows the output of the J48 Classifier with 125698 Correctly Classified Instances (99.7%) and 275 Incorrectly Classified Instances (0.2%). This suggests a very good classification performance.

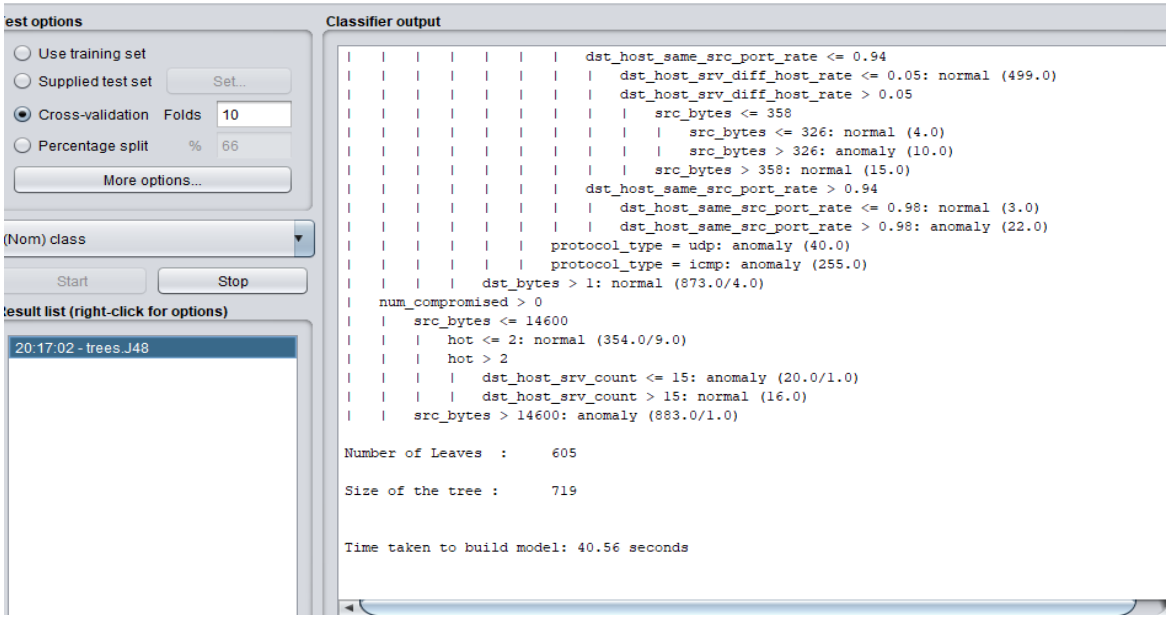
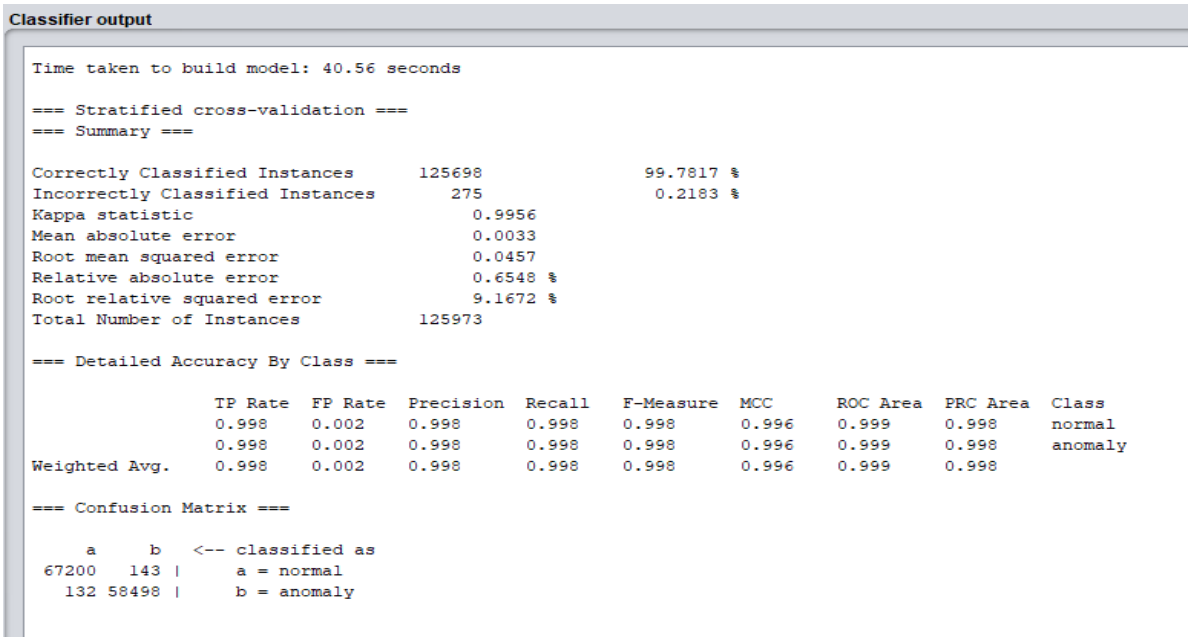


Figure 4: Output of J48 Classifier



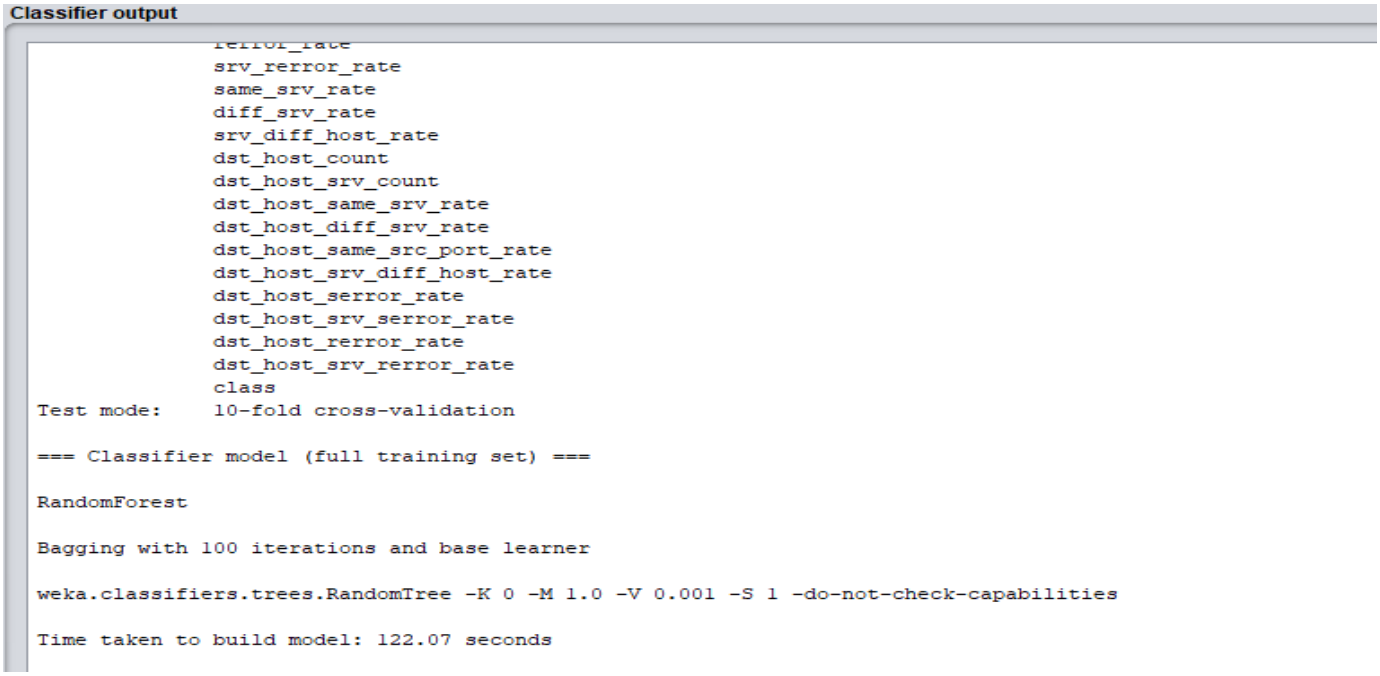
Classifier output

```
Time taken to build model: 40.56 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      125698               99.7817 %
Incorrectly Classified Instances       275                0.2183 %
Kappa statistic                          0.9956
Mean absolute error                      0.0033
Root mean squared error                  0.0457
Relative absolute error                  0.6548 %
Root relative squared error              9.1672 %
Total Number of Instances           125973

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.002    0.998      0.998    0.998      0.996    0.999     0.998     normal
                 0.998    0.002    0.998      0.998    0.998      0.996    0.999     0.998     anomaly
Weighted Avg.    0.998    0.002    0.998      0.998    0.998      0.996    0.999     0.998

=== Confusion Matrix ===

     a     b   <-- classified as
 67200   143 |     a = normal
   132 58498 |     b = anomaly
```

Figure 5: Output of J48 Classifier

C. Random Forest Classifier

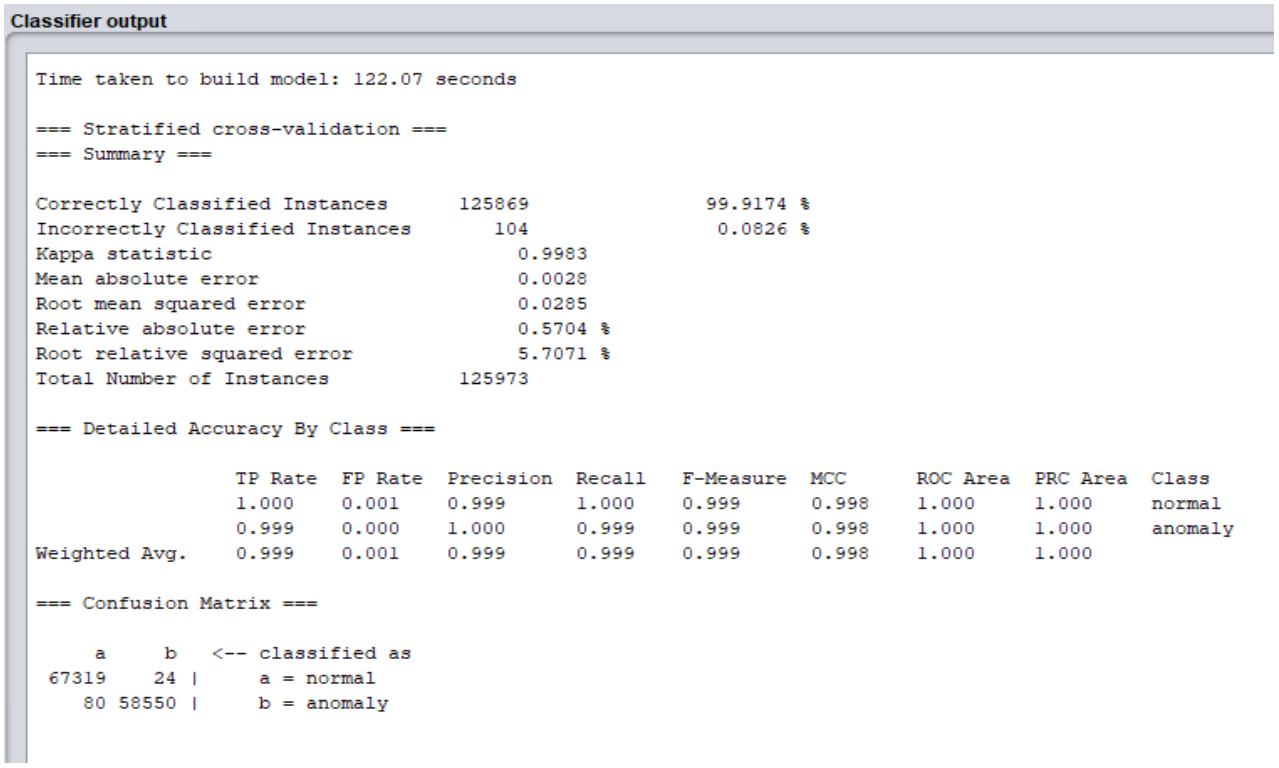
Similarly, the figure 6&7 shows the output of the Random Forest Classifier with 125869 Correctly Classified Instances (99.9%) and 104 Incorrectly Classified Instances (0.082%). This also suggest a very good classification performance.



Classifier output

```
                rerror_rate
                srv_rerror_rate
                same_srv_rate
                diff_srv_rate
                srv_diff_host_rate
                dst_host_count
                dst_host_srv_count
                dst_host_same_srv_rate
                dst_host_diff_srv_rate
                dst_host_same_src_port_rate
                dst_host_srv_diff_host_rate
                dst_host_serror_rate
                dst_host_srv_serror_rate
                dst_host_rerror_rate
                dst_host_srv_rerror_rate
                class
Test mode:      10-fold cross-validation

=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 122.07 seconds
```

Figure 6: Output of Random Forest Classifier



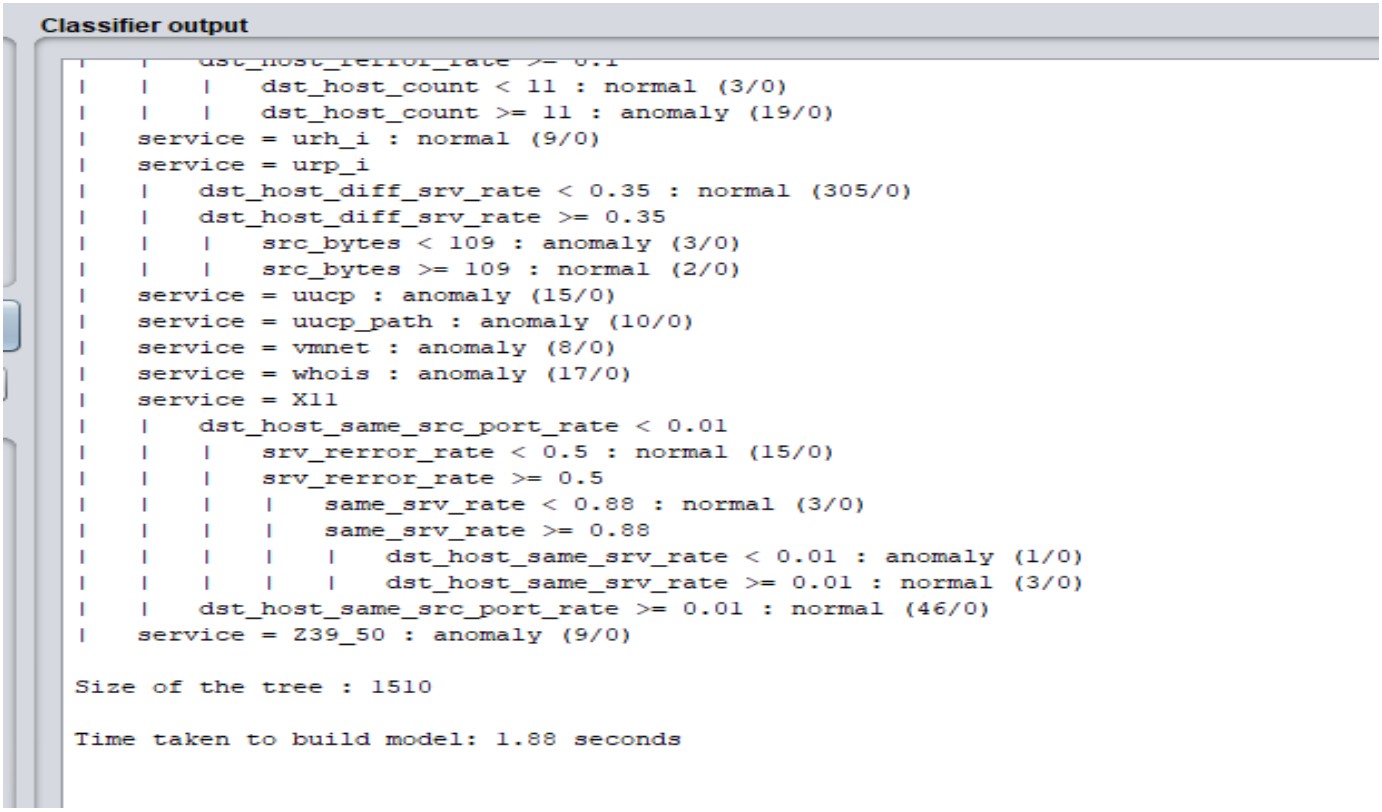
Classifier output

```
Time taken to build model: 122.07 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      125869               99.9174 %
Incorrectly Classified Instances       104                0.0826 %
Kappa statistic                          0.9983
Mean absolute error                      0.0028
Root mean squared error                  0.0285
Relative absolute error                  0.5704 %
Root relative squared error              5.7071 %
Total Number of Instances           125973

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.001    0.999      1.000    0.999      0.998    1.000     1.000     normal
                 0.999    0.000    1.000      0.999    0.999      0.998    1.000     1.000     anomaly
Weighted Avg.    0.999    0.001    0.999      0.999    0.999      0.998    1.000     1.000

=== Confusion Matrix ===

     a     b   <-- classified as
 67319    24 |     a = normal
    80 58550 |     b = anomaly
```

Figure 7: Output of Random Forest Classifier

D. Random Tree Classifier

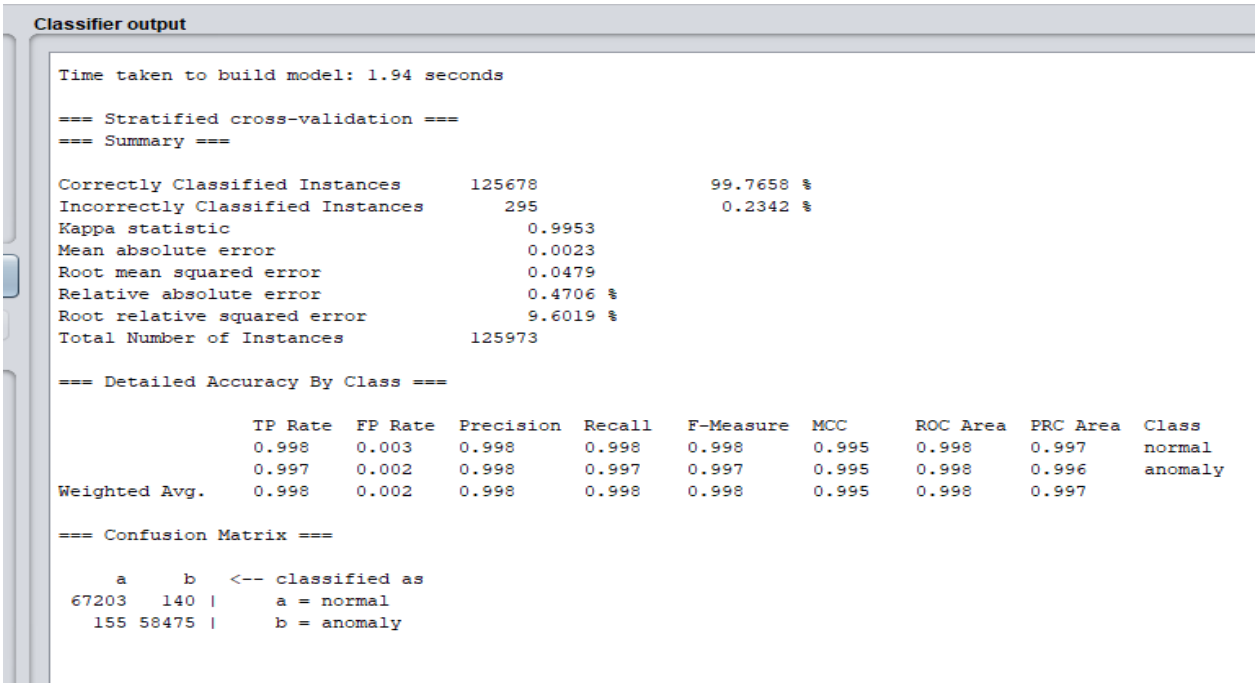
Furthermore, the figure 8&9 shows the output of the Random Tree Classifier with 125678 Correctly Classified Instances (99.7%) and 295 Incorrectly Classified Instances (0.23%). This algorithm also demonstrated competitive performance.



Classifier output

```
|   |   |   dst_host_count < 11 : normal (3/0)
|   |   |   dst_host_count >= 11 : anomaly (19/0)
|   service = urh_i : normal (9/0)
|   service = urp_i
|   |   dst_host_diff_srv_rate < 0.35 : normal (305/0)
|   |   dst_host_diff_srv_rate >= 0.35
|   |   |   src_bytes < 109 : anomaly (3/0)
|   |   |   src_bytes >= 109 : normal (2/0)
|   service = uucp : anomaly (15/0)
|   service = uucp_path : anomaly (10/0)
|   service = vmnet : anomaly (8/0)
|   service = whois : anomaly (17/0)
|   service = X11
|   |   dst_host_same_src_port_rate < 0.01
|   |   |   srv_rerror_rate < 0.5 : normal (15/0)
|   |   |   srv_rerror_rate >= 0.5
|   |   |   |   same_srv_rate < 0.88 : normal (3/0)
|   |   |   |   same_srv_rate >= 0.88
|   |   |   |   |   dst_host_same_srv_rate < 0.01 : anomaly (1/0)
|   |   |   |   |   dst_host_same_srv_rate >= 0.01 : normal (3/0)
|   |   dst_host_same_src_port_rate >= 0.01 : normal (46/0)
|   service = Z39_50 : anomaly (9/0)

Size of the tree : 1510

Time taken to build model: 1.88 seconds
```

Figure 8: Output of Random Tree Classifier



```
Classifier output

Time taken to build model: 1.94 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      125678               99.7658 %
Incorrectly Classified Instances       295                0.2342 %
Kappa statistic                          0.9953
Mean absolute error                      0.0023
Root mean squared error                  0.0479
Relative absolute error                  0.4706 %
Root relative squared error              9.6019 %
Total Number of Instances           125973

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.003    0.998      0.998    0.998      0.995    0.998     0.997     normal
                 0.997    0.002    0.998      0.997    0.997      0.995    0.998     0.996     anomaly
Weighted Avg.    0.998    0.002    0.998      0.998    0.998      0.995    0.998     0.997

=== Confusion Matrix ===

     a     b   <-- classified as
 67203   140 |     a = normal
   155 58475 |     b = anomaly
```

Figure 9: Output of Random Tree Classifier

From the figures or results above, the four-machine learning algorithm performed a classification technique against the classes of attacks and it shows that the Random Forest algorithm has the highest precision in classifying the attacks in the class label followed by J48 amongst other classifiers in the experiment. Also, the results show that Random Forest algorithm has the highest detection accuracy (Recall) followed by Random Tree algorithm.

Finally, Random Tree classifier outperform the other classifier in carrying out F-Measure in the experiment.

Table 3 Percentage of Weighted Average of The Four Classifiers

Evaluation Metrics	Bayes Net (%)	J48 (%)	Random Forest (%)	Random Tree (%)
Precision	97.3	99.8	99.9	99.8
Recall	97.2	99.8	99.9	99.8
F-Measure	97.2	99.8	99.9	99.8

The table 3 above described the percentages of the weighted average of the machine learning classifiers that were used to perform the experiment.

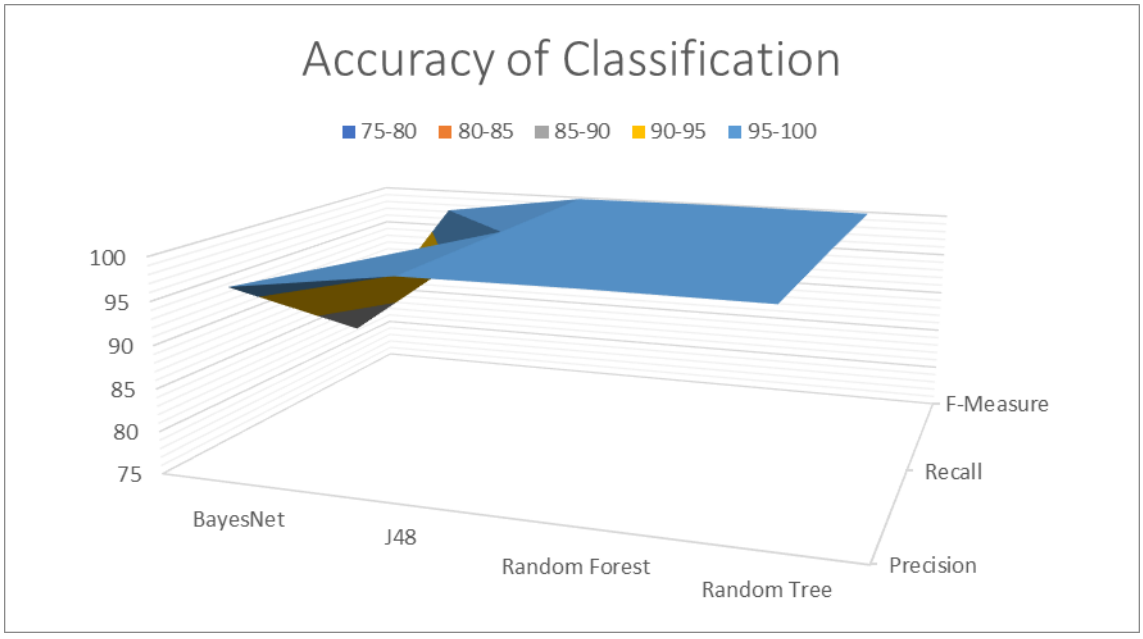


Figure 10: Accuracy of Classification of Four (4) Machine Learning Algorithm

The graph in figure 4.6 is generated from Table 3; the Y-axis denotes the percentage of accuracy while the X-axis represents the Machine Learning Classifiers. The graph was plotted in other to obtain the percentage of accuracy in the four (4) classifiers. The comparison shows that Random Forest and Random Tree algorithms outperform the other algorithms in their level of precision and F-measure as they are above 99% and 98% respectively, while the Random Forest outperforms the others by its detection rate. However, the Random Forest and Random Tree algorithms are more efficient in performing classification exercise on the Test datasets

5. CONCLUSION AND FUTURE DIRECTION

There has been a great need over the years for a machine learning based intrusion detection system due to the widespread proliferation of computer networks which has resulted in the increase of attacks on information system. These attacks are used for illegally gaining access to information, misuse of information or to reduce the availability of information to authorized users. These attacks are increasing at a staggering rate and so is their complexity. There is therefore, need for complete protection of public and private organizational computing resources which is driving the attention of people towards intrusion detection system. Our proposed system is able to efficiently protect the network system against intrusions at the point of entry and therefore save a lot of public and private organizations a lot of problems. The proposed system will be of great use or importance to all organizations especially network-based systems. The work performed in this research provides a basis for future research of hybrid intrusion detection systems. An area of future direction is to increase the number of datasets especially the NSL-KDD data set (NSL-KDD).

REFERENCES

1. Wu, W. Li, R. Xie, G. An, J. Bai, Y. Zhou, J. Li, K. (2019). A Survey of Intrusion Detection for In-Vehicle Networks, *IEEE Transactions on Intelligent Transportation Systems* PP (1) (2019) 1–15. doi:10.1109/TITS.2019.2908074J. U. Duncombe. **Infrared navigation—Part I: An assessment of feasibility**, *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34-39, Jan. 1959.
2. Thing, V. L. L. & Wu, J. (2016). Autonomous Vehicle Security: A Taxonomy of Attacks and Defences, in: Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data, iThings-GreenCom-CPSCom-Smart Data 2016, 2017, pp. 164–170. doi:10.1109/iThings-GreenCom-CPSCom-SmartData.2016. 52.
3. Omer, M. (2019). Implementing security techniques to lower the probability of IoT- devices getting hacked (2019).
4. Choudhary, S. & Kesswani, N. (2019). Analysis of KDD-Cup'99, NSL-KDD and UNSW- NB15 Datasets using Deep Learning in IoT, Procedia Computer Science 167 (2019) (2020) 1561–1573. doi:10.1016/j.procs.2020.03.367. URL https://doi.org/10.1016/j.procs.2020.03.36
5. Anup K. G. Aaron, S. & Michael, S. (1999). Learning Program Behavior Profiles for Intrusion Detection.. In Workshop on Intrusion Detection and Network Monitoring, Vol. 51462. 1–13.
6. Radford, B.J., Apolonio, L.M., Trias, A.J. & Simpson, J.A. (2018) Network traffic anomaly detection using recurrent neural networks. CoRR abs/1803.10769 H. Poor. ***An Introduction to Signal Detection and Estimation***; New York: Springer-Verlag, 1985, ch. 4.
7. Chiroma, H., Abdulhamid, S.M., Ya'aGital, A., Usman, A.M and Maigari, T.U (2011) "Academic Community Cyber Cafes: A Perpetration Point for Cyber Crimes in Nigeria". International Journal of Information Sciences and Computer Engineering, Vol. 2, No.2 pp. 7-13
8. Olusola, M., Samson, O., Semiu, A and Yinka, A (2013) "Impact of Cyber Crimes on Nigerian Economy".The International Journal of Engineering and Science (IJES). Vol. 12, Issue 4. Pp. 45-51..
9. The Guardian News paper (2013).
10. Yasir Hamid, M. Sugumaran & Ludovic Journaux (2017). Machine Learning Techniques for Intrusion Detection: A Comparative Analysis. DOI: http://dx.doi.org/10.1145/2980258.2980378
11. Khan, S. (2008). Ethem Alpaydin. Introduction to Machine Learning (Adaptive Computation and Machine Learning Series). The MIT Press. *Natural Language Engineering*, vol. 14, no. 01, pp. 133–137.