A Comparative Study of Support Vector Machine and C4.5 Classifiers on Network Intrusion Data

Solomon Olalekan Akinola[1], Oladimeji Arowolo Abiola[2] and Monsur Olumide Sokunbi[3]
Department of Computer Science,
University of Ibadan, Ibadan,
Nigeria.
Email: [1]solom202@yahoo.co.uk, [2]oladimejiarowolo@yahoo.co.uk, [3]monsur.sokunbi@gmail.com

ABSTRACT

An Intrusion Detection System (IDS) is a device or software application that monitors events occurring on the network and analyses it for any kind of malicious activity that violates computer security policy. An intrusion detection system should be able to detect accurately and have minimal rates of false alarms. The accuracy and false positive rates are still a lingering problem in IDSs. This study was designed to employ the use of data mining techniques for improving IDSs in a computer network. Two classifiers, the support Vector machine (SVM) and Decision Tree, C4.5, were combined using a weighted majority rule voting technique in Waikato Environment for Knowledge Analysis (WEKA) tool environment. The technique combines the SVM and C4.5 classifiers, trains and tests them on 20 percent of the Network Security Laboratory - Knowledge Discovery (NSL-KDD) dataset. Information gain was used for pre-processing the data. The classifiers were also individually used to train and test the dataset. Performance evaluations were carried out on the individual and combined classifiers focusing mainly on the accuracy and the false positive rate. The study shows the C4.5 classifier performing better than the SVM classifier and the weighted majority rule ensemble. The C4.5 classifier gave a precision of 0.988 while SVM and weighted majority rule gave precisions of 0.972 and 0.974, respectively. The C4.5 also has a lower false positive rate than the other classifiers. The study shows that the decision tree C4.5 classifier is most effective than the statistically based Support Vector Machine classifier and the weighted majority rule ensemble. This result is in tandem with earlier results reported by researchers in the literature.

Keywords: *Intrusion detection system, Data mining, C4.5 Decision tree algorithm, Support vector machine*

African Journal of Computing & ICT Reference Format:

Solomon Olalekan Akinola, Oladimeji Arowolo Abiola and Monsur Olumide Sokunbi (2020), A Comparative Study of Support Vector Machine and C4.5 Classifiers on Network Intrusion Data, *Afr. J. Comp. & ICT*, Vol. 13, No. 1, pp. 43 – 57.

I. INTRODUCTION

The recent rise in the population of the Internet and personal computers has drastically increased the utilization rate of the internet. It is gradually changing people’s lives, and the majority of people study, communicate, shop and recreate on the internet. As the internet brings about convenience and real timeliness, there is however a consequent problem of information security; for instance: servers are attacked and paralyzed, inner data and information are stolen, etc. Intrusion detection begins with the designing of a secured computer network for data collection. Pattern-based software sensors monitor the network traffic and raise alarms when the traffic matches a saved pattern. After the alarms have been raised, the traffic data is sent to the security analysts who will then decide if the alarms represent a serious enough event to deserve a response. A response might be to shut down a part of the network, to phone the internet service provider associated with suspicious traffic, or to simply take note of unusual traffic for future reference. If the network is small and signatures are kept up to date, the human analyst solution to intrusion detection works well. But when organizations have a large, complex network, the human analysts quickly become overwhelmed by the number of alarms they need to review.

Intrusion Detection Systems appear like internet alarm and regulation device, to observe and investigate whether the internet attacks may occur, promptly send alarm before threats are caused by attacks, carry out appropriate countermeasures and reduce occurrence of bigger losses. Some technologies are based on pattern check, with low precision, but the pattern-base should be upgraded regularly. Such technologies do not possess enough detection capacity for unknown and renewed attack manners. Recently, many researches applied the technology of data mining and machine learning, which can analyse large data, and such technologies have better detection capacity for unknown attacks [1, 2, 3]. Though some research achievements have proven efficient, there is a lot of development potential.

This paper compares the intrusion detection performance of two data mining classification algorithms and their ensemble in terms of detection accuracy and other parameters (such as Kappa Statistic and Mean Absolute Error, MAE). The paper is organized as follows: existing related works are presented in Section 2 while the methodology adopted for the study is presented in Section 3. Results obtained are presented in Section 4 while Section 5 concludes the paper.

II. RELATED WORKS

Anderson [4] introduced a term audit trail which includes information for tracking down misuse and user behaviour. Basically, with the release of this paper, misuse detection concept gets introduced. His opinion provides a base for intrusion detection systems’ design and development.

Thereafter in 1983, SRI International’s Computer Science Lab, started working on a government project for intrusion detection systems development [5]. Their research goal was to explore misuse detection techniques to analyse audit trails. The first generation misuse components analysed system management facility (SMF) records from the IBM mainframe system. Later on, it started working on a rule-based expert system to detect known intrusions. This early research developed the very first prototype intrusion detection expert system (IDES) on the bases of audit trails.

Denning [6] proposes a model based on the real-time IDES ability to detect intrusion, break-ins and any kind of computer abuse. The model focused on detecting abnormal pattern or some kind of security violation of the system monitoring audit records. Based on the structure of the detection model, profiles have been created to represent the anomalous behaviour in the form of metrics and statistical models, which help to obtain information about the behaviour of audit records in future. This approach has become very useful for detecting intrusions that exploit a system. This approach provides an understanding of how intrusion is detected by basically monitoring it. But the major problem was to provide protection from any kind of exploits caused by vulnerability in a system. Fundamentally, the concept was to improve the system security by providing an extra layer of protection of IDES.

Alves-Foss [7] introduced a low cost approach based on clustering and multivariate analysis known as Network Analysis of Anomalous Traffic Events (NATE). This resolved the problem of those intrusion detection systems that were not able to handle high volume traffic and real time detection constraint. This solution was like any other light weighted approach with the quality feature of minimal network traffic measurement, limited attack scope and anomaly detection. NATE model performed on MIT Lincoln lab’s data. It consists of two phases of operation. In Phase-I, data collection and database creation was performed. In this phase, collected data were closely analysed for possible attack and assumed that only normal data was captured. But in reality, if the data Phase-I intrusion was treated as a normal then it would be big hindrance for further detection. Phase-II detects intrusion in real-time environment. This classification of normal and abnormal data was performed on the basis of cluster algorithm. This paper provides an idea about clustering, which was performed on the real time traffic so that easy and quick updating of the new features of the attacks in the database could be performed.

Barbara [8] proposes that rather than using the traditional method of detecting intrusion, a new approach would be introduced based on a data mining technique known as Bayesian analysis. This paper creates a test bed with ADAM (Audit Data Analysis and Mining), which helps to describe and study about various different types of data mining techniques in intrusion detection systems.

Shyu [9] introduces a statistical based intrusion detection system in which anomaly was treated as an outlier. The intrusion predictive model proposed was known as PCA (Principal Component Analysis). This model was constructed on the basis of distance between the occurrences of normal or anomaly behaviours. The proposed method was implemented on KDD Cup dataset 1999.

Hwang [10] introduces a new technique of generating Frequent Episode Rules (FER) of categorized internet traffic. This rule was helping to differentiate between various abnormal sequences (like Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP)) and normal traffic. Due to the increase in rule pruning the search space, time was reduced automatically. Therefore, it performed with better efficiency and higher detection rate. This proposed approach also showed better performance when implemented on the real time traffic.

Park [11] provides a survey on anomaly and hybrid (misuse and anomaly) detection system which was introduced in the past. In this paper, he also discussed the recent trends in anomaly detection and determined the various problem and challenges faced with detection. This provides a complete description about how a novel or zero-day attack detected by anomaly detection behaves and the basis of rule or knowledge base of known attacks detected by misuse detection. The paper includes a complete explanation about how traditional models differ from the existing models. So, to overcome the limitation of existing models while detecting intrusion, data mining techniques was introduced. There were so many different kinds of data mining techniques introduced to secure the network. The main idea of his proposed methods is to use data mining techniques to find out an accurate and robust model using system audit trail data.

Horng [12] proposed Support Vector Machine (SVM) - based intrusion detection system, which combines a hierarchical clustering algorithm (BIRCH) with SVM technique. Hierarchical clustering provides a simple feature selection procedure. It helps to reduce the training time for dataset and also improves the performance of SVM.

Lin [15] proposed a new novel approach called CANN (Cluster Centre and Nearest Neighbour). In this approach two distances were calculated and then summed up; one distance was between its cluster centre and each data sample. Second was K-NN distance known as nearest neighbour distance. It is a distance between the data and its nearest neighbour in the same cluster. Experimental results show that CANN does not perform better than K-NN. The limitation of this research method is that CANN didn’t give better performance in terms of detecting User to Root Attack (U2R) and Remote to Local Attack (R2L). This means that one-dimensional distance based feature selection performed less efficiently. So to remove this limitation, there is need to filter out the bad and noisy data which creates a problem with feature selection of a dataset.

Alaa and Amneh [3] compared C4.5 Decision Tree algorithm with Multi-layer Perceptron (MLP) and Support Vector Machine (SVM) for Network Intrusion Detection based Feature Analysis using NSL-KDD Data Set. They concluded that C4.5 has the best network intrusion classification performance among the three algorithms.

Oladele, *et.al.* [1] presents a survey of intrusion detection, its system and an improved classification algorithm seeking to remove the existing problem of data mining’s intrusion detection system. Information gain attribute evaluation technique (entropy) was used for performing feature selection (removal of redundant attributes) and feeding the filtered dataset into a multilayer perceptron algorithm for classification. The study concluded that feature selected dataset provides better results than the full dataset after classification is performed on both.

Oladejo, *et, al.,* [14] compare the performance of SVM and K-Nearest Neighbour Algorithm Using Microarray Data on Leukemia Cancer Dataset. They make use of two dimensionality reduction strategies, feature selection and feature extraction, to address the problems of highly correlated data. In this study, analysis of micro array data was carried out on Leukemia cancer dataset, with the end goal of finding the smallest quality subsets for precise tumor arrangement. Results from their study shows that SVM performed at 90% accuracy than the KNN algorithm, which had 81.67% accuracy.

Asiegbu, *et. al.* [15] studied Intrusion Detection System using Support Vector Machine and Infinite Latent Feature Selection Approach. The implementation was simulated using LibSVM 3.22 in a MATLAB integrated development environment and the KDD Cup' 99 dataset. The enhanced IDS achieved a detection accuracy of 99.88%, and 97.91% for normal network packets and attack network packets respectively, which is an improvement over the existing rate of 64.94% and 70% for detection accuracy of normal network packets and attack network packets respectively.

NSL-KDD is a data set suggested to solve some of the inherent problems of the KDD'99 data set. Although, this new version of the KDD dataset still suffers from some of the problems and may not be a perfect representative of existing real networks, because of the lack of public data sets for network based IDSs, it can be applied as an effective benchmark dataset to help researchers compare different intrusion detection methods. Furthermore, the number of records in the NSL-KDD train and test sets are reasonable. This advantage makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research work will be consistent and comparable [16].

Ji [17] reports that the NSL-KDD data set has the following advantages over the original KDD data set:

i. There are 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.
ii. The number of records in the train and test sets is reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable.
iii. It does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records.
iv. 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.

III. METHODOLOGY

The system is an ensemble based intrusion detection system to observe the accuracy of two individually strong classifiers. All step by step experiments were done by applying the selected classification algorithms on the Network Security Laboratory - Knowledge Discovery (NSL-KDD) dataset. Importantly, the building of experiment evaluation environment with major steps was done which involves the Environmental setup, Data pre-processing and choosing the data mining software.

The detailed tables of results having the performance of the selected classifiers were presented and also a table for comparison for their performances. The table for comparison holds the results of the performance of each individual base classifier against the performance of the ensemble method via voting.

3.1 Dataset

This study used 20 percent of the NSL-KDD dataset. The dataset was experimented on using two algorithms; the Support Vector Machine and C4.5 classifiers as individual classifiers and then combined using the weighted majority rule approach. The N-fold cross validation method is used, and the N value here is 10, which is the default value. This partitions the dataset, randomly, into 10 parts; 9 of the parts was used as the training dataset that trained each algorithm and the final partition was used for the testing.

Experiments and analysis were carried out for both the individual and combined classifiers, comparing their effectiveness using the performance measurements mentioned earlier, focusing more on accuracy and false positives which are the two major factors in the workings of an intrusion detection system

3.2 Feature Selection

While collecting real-time traffic, it includes some irrelevant, redundant, unreliable data, which is unrelated for knowledge discovery known as noisy data. Due to this noisy data, there is effect on the result of data mining during the classification of normal and abnormal traffic. Therefore, to improve the detection capabilities of classifiers, there is a need to pre-process the data. Pre-processing provides a feature to clean and normalize the data so that any kind of irrelevant data never affects the accuracy of classifiers. To remove the redundant data, there are various kinds of pre-processing filters which are available. In the present model, the information gain method was used to pre-process the data. It measures the amount of information in bits about the class prediction. What this means is that it measures the decrease in the weighted average impurity of the attributes compared with the impurity of the complete set of data items. Hence, the attributes with the largest information gain are considered to be most useful for classifying data items. This approach removes redundant or irrelevant features from the dataset to prevent decrease in classification accuracy and unnecessary increase in computational cost [18].

3.3 The Testing and Training Data

To build a competent model, it is necessary to use a separate dataset to build a model and then another dataset to test the accuracy of the model. This helps to avoid the problem of over-fitting, which occurs when either the model under or over generalises the relationship between the output class and the input attributes. Over generalising the relationship results in a model that is incapable of correctly classifying new data. For both C4.5 and Support Vector Machine algorithms, a 10-fold cross-validation method was used. The data was randomly split into 10 parts in which the class was represented in approximately the same percentages as in the full dataset. Each split was selected in turn and the learning scheme trained on the remaining nine-tenths; then the error rate was calculated on the selected set. At the end of the process, the procedure was executed a total of 10 times on the 10 different training sets built from the original training dataset. The 10 error estimates were also averaged to yield an overall error estimate.

The adopted system architecture for this study is shown in Figure 1. The NSL-KDD dataset was pre-processed. Information gain is used in the pre-processing to remove irrelevant attributes from the dataset. The data was then trained and tested using the N-fold cross validation. The weighted majority rule ensemble combining the Support Vector Machine and C4.5 algorithms was then used to classify the data. The results of the experiment was analysed and compared with the performance of the individual SVM and C4.5 classifiers.

The data mining software used was Waikato Environment for Knowledge Analysis (WEKA) tool and the algorithms employed via the voting method were the Support Vector Machine (SMO) and C 4.5 decision tree (named J48 in WEKA) as the base classifiers.

Weka processes data in the Attribute Relation File Format (ARFF). This file format consists of a list of instances and the attribute values for each instance. It is created by adding the dataset's name using the *@relation* tag, the attribute information using the *@attribute* and *@data* tags. A snippet of the intrusion dataset ARFF format is shown in Figure 2.

The experiments were carried out on a 64-bit Windows 10 Home operating system with 8GB of RAM and an intel core i5 CPU @ 2.20GHz. Figure 3 shows the pre-process tab of the Weka explorer interface. This is the first page seen after the dataset has been loaded into the tool. This interface gives an overview of the dataset contents.

IV. RESULTS

In this section, we show the output of the experiments, after which an analysis of the output is presented. Table 1 and Figure 5 show the statistical measures of the SVM, C4.5 and weighted majority rule classifiers while Figure 4 shows the results of the NSL-KDD dataset on the weighted majority rule ensemble

A Kappa statistic value of 0.944 indicates a high level of agreement between the Support Vector Machine classifier and the actual class value. The C4.5 classifier has a Kappa statistic of 0.976, which is higher than that of the SVM classifier. This indicates a higher level of agreement between the classifier and actual class values. The ensemble method using weighted majority rule to combine the SVM and C4.5 algorithms gave a Kappa statistic of 0.948, which is higher than that of the SVM, but lower than that of the C4.5 algorithm.

The mean absolute error value of the SVM classifier for our dataset was 0.028. This is low and it indicates that the algorithm performs well in the classification task. The mean absolute value of the C4.5 classifier was 0.017. That of the weighted majority rule was 0.023. This is lower than the SVM algorithm, but higher than the C4.5 algorithm. This shows that the C4.5 classifier performs better than the SVM classifier and the ensemble in this classification task.

The root mean square error value of the SVM classifier was 0.167 and it is higher than the 0.103 value of the C4.5 classifier. The root mean square error of the weighted majority rule at 0.117 is lower than that of the SVM classifier, but higher than that of the C4.5 classifier. This shows that the C4.5 algorithm performs better than both the SVM and the weighted majority rule. From definition, the root mean square value is higher than the mean absolute error value.

The relative absolute error value of the C4.5 classifier was 3.49% which is lower than the 5.61% and 4.55% of the

SVM and weighted majority rule respectively. The root relative squared error of the SVM classifier was 34%, the C4.5 had 21% and the weighted majority rule had 23%.

Tables 2 to 4 show the performance evaluations for the Support Vector Machine, C4.5 and Weighted Majority Rule respectively.

The True Positive (TP) rate is in short, a measure of correct classification. For the SVM classifier, the true positive value was 0.983 for the normal class and 0.960 for the anomaly class.

For the C4.5 classifier, the True Positive value was 0.990 for the normal class and 0.985 for the anomaly class. For the weighted majority rule, the True positive for the normal class was 0.987 and 0.959 for the anomaly class. The C4.5 classifier had higher True Positive values than the SVM and weighted majority rule classifiers. This implies that it predicts more classes correctly than either the SVM or weighted majority rule classifiers for this intrusion task.

The False Positive (FP) rate is the exact opposite of the True Positive rate as it represents incorrect classification. The False Positive values for the SVM classifier was 0.040 for the normal class and 0.017 for the anomaly class. The C4.5 classifier had a false positive rate of 0.015 for the normal class and 0.010 for the anomaly class. The weighted majority rule had a False Positive value of 0.041 for the normal class and 0.013 for the anomaly class.

The SVM classifier had a precision value of 0.965 for the normal class and 0.980 for the anomaly class. The C4.5 classifier had a precision value of 0.987 for the normal class and 0.989 for the anomaly class. The weighted majority rule had a precision of 0.965 for the normal class and 0.959 for the anomaly class. It is obvious that the C4.5 classifier had higher precision than the SVM and the weighted majority rule.

Recall is the proportion of actual positives or negatives which are predicted as positive or negative. The SVM classifier had a normal class recall statistic of 0.983 and an anomaly class value of 0.960. The C4.5 had a recall value of 0.990 for the normal class and 0.985 for the anomaly class. The weighted majority rule had a recall value of 0.987 for the normal class and 0.959 for the anomaly class. The recall statistic suggests again that the C4.5 classifier is a better classification algorithm for this task.

The confusion matrix is a table that allows us to assess the performance of the classifier. The columns of the table represent the instances of the normal and anomaly predicted classes. While the rows represent the actual normal and anomaly class values. Tables 5 to 7 show the confusion matrices obtained for the classifiers used in this study.

From the tables, the SVM classifier correctly classifies 6881 instances and misclassifies 119 instances. It also classifies 5936 instances of the anomaly class correctly but wrongly classifies 250 instances. The C4.5 classifier correctly classifies 6932 instances and misclassifies 68 instances. For the anomaly class, it correctly classifies 6095 instances but misclassifies 91 instances. The weighted majority rule correctly classifies 6911 instances and misclassifies 89 instances. For the anomaly class, it correctly classifies 5934 instances but misclassifies 252 instances.

The confusion matrices show that the C4.5 classifier performs much better than others in predicting both normal and anomaly instances. However, between the SVM and weighted majority rule, the latter performs better in predicting normal classes while the former performs better in predicting anomaly classes.

Table 8 shows the percentage accuracies of classifications for the classifiers. In all, the C4.5 classifier had the highest accuracy for the classification tasks.

V. CONCLUSION

Nowadays prevention of security breaches using the existing security technologies is unrealistic. As a result, intrusion detection is an important component in network security. Also, misuse detection technique cannot detect unknown attacks; so the anomaly detection technique is used to identify these attacks.

In this study, the intrusion detection performance of two data mining classification algorithms and their ensemble in WEKA (C4.5 Decision Tree and Support Vector machine, SVM) were compared. This work evidently shows that the C4.5 algorithm produces better results than the Support Vector Machine and weighted majority rule algorithms. The result is in tandem with the work of Alaa and Amneh [3] on their "Professional Comparison of C4.5, MLP, SVM for Network Intrusion Detection based Feature Analysis". They concluded that decision tree based C4.5 algorithm achieved the highest classification accuracy compared to other search techniques explored. This result would be useful for determining the choice of algorithm to use in an intrusion detection system.

REFERENCES

[1] T. O. Oladele, E. J. Adejoke, and A. Samuel., 'Performance Evaluation of Multilayer Perceptron Classifier on the KDD'99 Full and Reduced Dataset using WEKA Tool', *African Journal of Computing & ICT*, Vol. 9, Nos 1 & 2 (combined), pp. 22 – 53, 2016.

[2] A. A. Adeniran, and S. O. Akinola. .An Ensemble Data Mining Approach for Intrusion Detection in a Computer Network, *International Journal of Science and Engineering Investigations (IJSEI)* , Volume 6, Issue 62, pp. 73 – 77, 2017.

[3] F. S. Alaa and A. Amneh. A Professional Comparison of C4.5, MLP, SVM for Network Intrusion Detection based Feature Analysis, *Special Issue on Network Security Techniques, CNIR Journal*, ICGST LLC, Delaware, USA, pp. 15 – 29, 2015.

[4] J. Anderson. Computer security threat monitoring and surveillance. Technical report, James P. Anderson Company, Fort Washington, Pennsylvania, 1980.

[5] C. E. S. Endorf. Intrusion detection & prevention. New York: McGraw-Hill/Osborne, 2004.

[6] D. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, pp. 222-232, 1987.

[7] C. A. Alves-Foss. Nate: Network analysis of anomalous traffic events, a low-cost approach. *Proceedings of the 2001 workshop on New security paradigms*, 89-96, 2001.

[8] D. J. C. Barbara. Audit Data Analysis and Mining, ADAM: a testbed for exploring the use of data mining in intrusion detection. *ACM SIGMOD Record*, Volume 30, Issue 4, pp 15–24, 2001.

[9] M. L. S. C. Shyu. A novel anomaly detection scheme based on principal component classifier. Miami University Coral Gables FL Department of Electrical and Computer Engineering, 2003.

[10] M. Q. Hwang. Frequent episode rules for internet anomaly detection. Third IEEE *International Symposium on Network Computing and Applications* (NCA), pp. 161-168, 2004.

[11] A. P. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 3448-3470, 2007.

[12] M. S. Horng. A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert Systems with Applications*, 306-313, 2011.

[13] W. S. K. Lin. CANN: An intrusion detection system based on combining cluster centres and nearest neighbours. *Knowledge-Based Systems*, 13-21, 2015.

[14] A. K. Oladejo, T. O. Oladele and K. S. Yakub, Comparative Evaluation of Linear Support Vector Machine and K-Nearest Neighbour Algorithm using Microarray Data On Leukemia Cancer Dataset, *Afr. J. Comp. & ICT*, Vol.11, No.2, pp. 1 – 10, 2018.

[15] B. C. Asiegbu, C. O. Ikerionwu and N. C. Ajanwachuku , An Intrusion Detection System using Support Vector Machine and Infinite Latent Feature Selection Approach, *Afr. J. Comp. & ICT*, Vol.12, No. 3, pp. 74 – 84, 2019.

[16] R. Goel, A. Sardana and R. Joshi. Parallel Misuse and Anomaly Detection Model. *International Journal of Network Security*, 211-222, 2011.

[17] W. D. Ji. A MapReduce Implementation of C4.5 Decision Tree Algorithm. *International Journal of Database Theory and Application*, 49-60, 2014.

[18] L. Joseph and R. Sudha. Data Mining Based Intrusion Detection. *International Journal of Computer Science, Systems Engineering and Information Technology*, 4(1), 2011, pp. 81- 86, 2011.

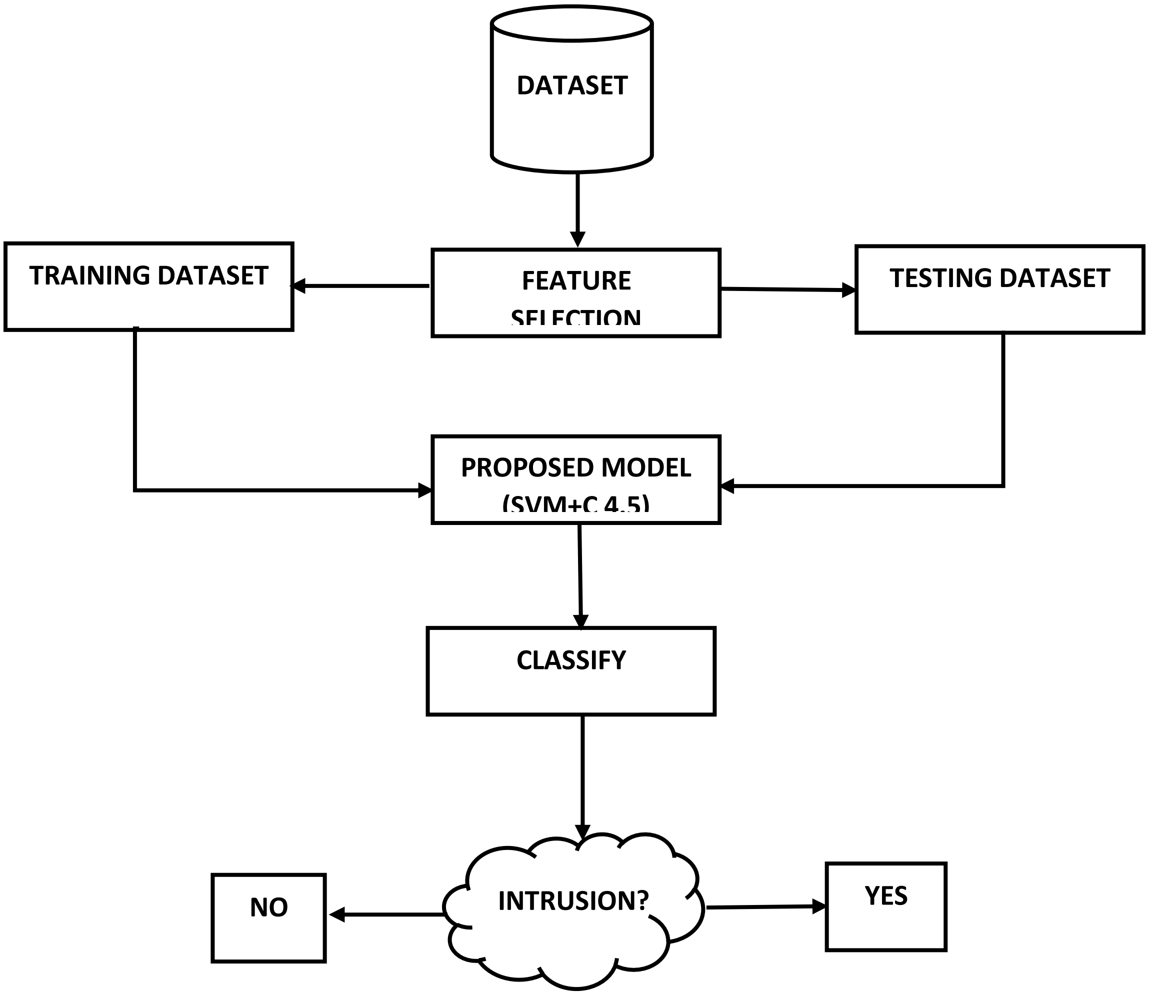
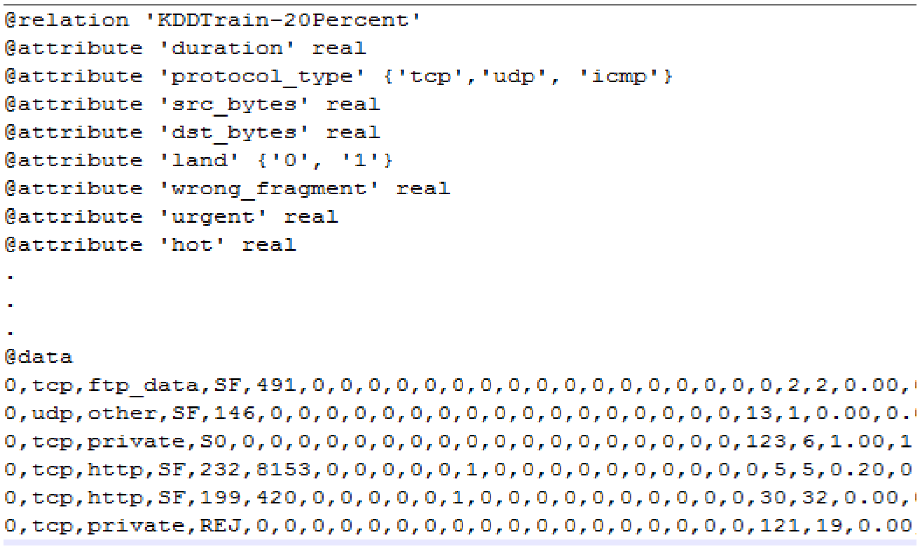
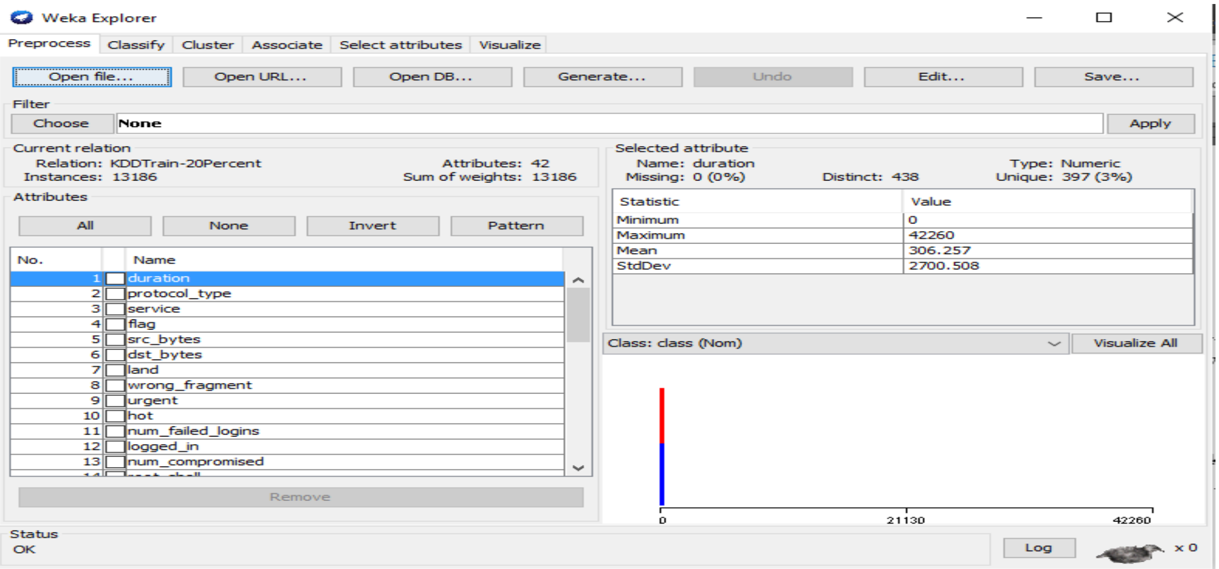


Figure 1: System Architecture



```
@relation 'KDDTrain-20Percent'
@attribute 'duration' real
@attribute 'protocol_type' {'tcp','udp', 'icmp'}
@attribute 'src_bytes' real
@attribute 'dst_bytes' real
@attribute 'land' {'0', '1'}
@attribute 'wrong_fragment' real
@attribute 'urgent' real
@attribute 'hot' real
.
.
.
@data
0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,
0,udp,other,SF,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0.00,0.
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1.00,1
0,tcp,http,SF,232,8153,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5,0.20,0
0,tcp,http,SF,199,420,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,30,32,0.00,
0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,121,19,0.00
```

Figure 2:*The ARFF input for Weka*



Filter
None

Current relation
Relation: KDDTrain-20Percent Attributes: 42
Instances: 13186 Sum of weights: 13186

Attributes

No.		Name
1	□	duration
2	□	protocol_type
3	□	service
4	□	flag
5	□	src_bytes
6	□	dst_bytes
7	□	land
8	□	wrong_fragment
9	□	urgent
10	□	hot
11	□	num_failed_logins
12	□	logged_in
13	□	num_compromised

Selected attribute
Name: duration Type: Numeric
Missing: 0 (0%) Distinct: 438 Unique: 397 (3%)

Statistic	Value
Minimum	0
Maximum	42260
Mean	306.257
StdDev	2700.508

Class: class (Nom)

Status
OK

Figure 3: *The Weka Explorer interface*

```
                    Number of Leaves  :   422

                    Size of the tree :          474

              Time taken to build model: 16.69 seconds

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

     Correctly Classified Instances        12845              97.4139 %
      Incorrectly Classified Instances       341               2.5861 %
                 Kappa statistic                    0.948
                Mean absolute error                  0.0227
               Root mean squared error                0.1169
               Relative absolute error             4.5532 %
              Root relative squared error          23.4291 %
             Coverage of cases (0.95 level)          99.4085 %
            Mean rel. region size (0.95 level)      51.5471 %
                Total Number of Instances           13186
                 === Detailed Accuracy By Class ===


 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
                               Class

   0.987    0.041    0.965      0.987    0.976     0.948    0.995     0.993     normal
```

```
                 0.959    0.013    0.985     0.959    0.972     0.948    0.995     0.995     anomaly

Weighted Avg.    0.974    0.028    0.974     0.974    0.974    0.948   0.995     0.994


                              === Confusion Matrix ===

                                  a    b   <-- classified as

                                6911   89 |    a = normal

                               252 5934 |    b = anomaly
```

Figure 4: *Results of the NSL-KDD dataset on the weighted majority rule ensemble*

Table 1: *Statistical measures for individual and ensemble classifiers*

STATISTICAL MEASURES	SUPPORT VECTOR MACHINE	C4.5	WEIGHTED MAJORITY RULE (SVM+C4.5)
Kappa Statistic	0.944	0.976	0.948
Mean Absolute Error	0.028	0.017	0.023
Root Mean Squared Error	0.167	0.103	0.117
Relative Absolute Error	5.618%	3.488%	4.553%
Root Relative Squared Error	33.521%	20.684%	23.429%

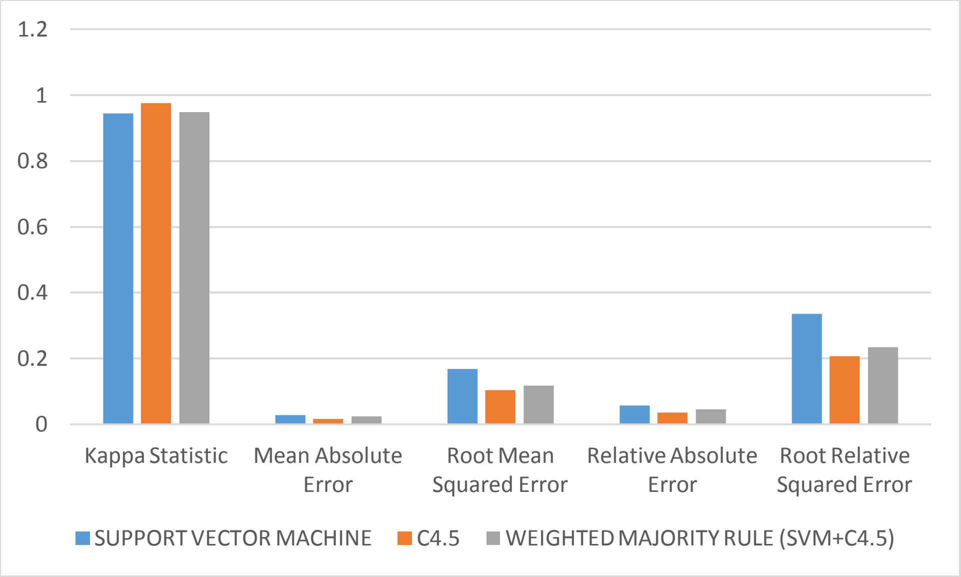


Figure 5: *Bar chart showing the performance measures of the classifier types*

Table 2: *Performance Evaluation for the Support Vector Machine classifier*

	TP RATE	FP RATE	PRECISON	RECALL
NORMAL	0.983	0.040	0.965	0.983
ANOMALY	0.960	0.017	0.980	0.960
WEIGHTED AVERGAE	0.972	0.029	0.972	0.972

Table 3: *Performance Evaluation for the C4.5 classifier*

	TP RATE	FP RATE	PRECISON	RECALL
NORMAL	0.990	0.015	0.987	0.990
ANOMALY	0.985	0.010	0.989	0.985
WEIGHTED AVERGAE	0.988	0.012	0.988	0.988

Table 4: *Performance Evaluation for the Weighted Majority Rule*

	TP RATE	FP RATE	PRECISON	RECALL
NORMAL	0.987	0.041	0.965	0.987
ANOMALY	0.959	0.013	0.985	0.959
WEIGHTED AVERGAE	0.974	0.028	0.974	0.974

Table 5: *Confusion matrix for SVM classifier*

	PREDICTED CLASS	
	NORMAL	ANOMALY
NORMAL	6881	119
ANOMALY	250	5936

Table 6: *Confusion matrix for C4.5 classifier*

	PREDICTED CLASS	
	NORMAL	ANOMALY
NORMAL	6932	68
ANOMALY	91	6095

Table 7: *Confusion matrix for weighted majority rule*

	PREDICTED CLASS	
	NORMAL	ANOMALY
NORMAL	6911	89
ANOMALY	252	5934

Table 8: *Percentage Accuracies of Classifications.*

	SVM (%)	C4.5 (%)	Weighted Majority Rule (%)
Correct Classification	96.201	98.794	97.414
Incorrect Classification	2.798	1.206	2.586