

IMPROVED PERFORMANCE OF INTRUSION DETECTION SYSTEM USING FEATURE REDUCTION AND J48 DECISION TREE CLASSIFICATION ALGORITHM

O. C. Abikoye¹, A. O. Balogun², A. K. Olarewaju³ & A. O. Bajeh⁴

Department of Computer Science, University of Ilorin,
Ilorin, Nigeria.

¹kemi_adeoye@yahoo.com, ²bharlow058@gmail.com,

³larykcomputer@yahoo.com, ⁴bajeh_amos@yahoo.com

ABSTRACT

Due to the obvious importance of accuracy in the performance of intrusion detection system, in addition to the algorithms used there is an increasing need for more activities to be carried out, aiming for improved accuracy and reduced real time used in detection. This paper investigates the use of filtered dataset on the performance of J48 Decision Tree classifier in its classification of a connection as either normal or an attack. The reduced dataset is based on using Gain Ratio attribute evaluation technique (entropy) for performing feature selection (removal of redundant attributes) and feeding the filtered dataset into a J48 Decision Tree algorithm for classification. A 10-fold cross validation technique was used for the performance evaluation of the J48 Decision Tree classifier on the KDD cup 1999 dataset and simulated in WEKA tool. The results showed J48 decision tree algorithm performed better in terms of accuracy and false positive report on the reduced dataset than the full dataset (Probing full dataset: 97.8%, Probing reduced dataset: 99.5%, U2R full dataset: 75%, reduced dataset: 76.9%, R2L full dataset: 98.0%, reduced dataset: 98.3%).

KEY WORDS: *IDS, Feature reduction, data mining, classification, decision tree, gain ratio.*

1. INTRODUCTION

Due to the fast growing of the network, security has become the fundamental issue of computer technology. So the main task of the technology expert is to provide a secure data confidentiality, data integrity and data availability (Hemant, Sarkhedi & Vaghamshi, 2013). As network of computers has become an important part of the society, the security of such network is of great importance as failure to do so may lead to a dangerous effect, such as information theft. Information gotten from a private network, for instance governmental information, can be used against the progress of the society which can lead to pandemonium and panic in the society at large.

Intrusion detection and prevention is a developing field as it tends to demand maximum attention nowadays due to the prevalent activities of hacker. Network attacks have increased in number and severity over the past few years, thus intrusion detection system (IDS) is increasingly becoming a critical component to secure the network (Nguyen & Choi, 2008). Hence, failure to secure the network has led to various cyber-attacks, infection of (networked) computers by viruses, information theft and leakage, unauthorized access to classified information and misuse of information (Mukkamala, Sung, & Abraham, 2005). Despite the use of different types of antiviruses and firewalls in the access and information protection, there are still cases of leakage, theft and intrusion which always cause a devastating effect. Several studies have worked on improvement of intrusion detection techniques (Patcha & Park, 2007; Mukkamala, Sung, Abraham, & Ramos, 2004; Hashemi, Muda & Yassin, 2013; Balogun & Jimoh, 2015; Balogun, Balogun, Adeyemo, & Sadiku, 2015).

Data Mining refers to the process of extracting hidden, previously unknown and useful information from large databases (D'silva & Vora, 2013). Classification is a data mining technique that creates a step-by-step guide for how to determine the output of a new data instance. The tree it creates is exactly that: a tree whereby each node in the tree represents a spot where a decision must be made based on the input, and we move to

the next node and the next until we reach a leaf that tells you the predicted output (Karim & Rahman, 2013).

Classification is one of data mining functionalities. It finds a model or function that separates classes or data concepts in order to predict the classes of an unknown object. The data analysis task is classification, where a model or classifier is constructed to predict class (categorical) labels, for example such as “safe” or “risky” for the loan application data. These categories can be represented by discrete values, where the ordering among values has no meaning. Due to the fact that the class labels of training data is already known, it is also called supervised learning. Classification consist two processes: (1) training and (2) testing. The first process, training, builds a classification model by analysing training data containing class labels while the second process, testing, examines a classifier (using testing data) for accuracy (in which case the test data contains the class labels) or its ability to classify unknown objects (records) for prediction (Mabayoje, Balogun, Salihu, & Oladipupo, 2015) .

The majority of real-world classification problems require supervised learning where the underlying class probabilities and class-conditional probabilities are unknown, and each instance is associated with a class label. In real-world situations, relevant features are often unknown a priori. Therefore, many candidate features are introduced to better represent the domain. Unfortunately many of these are either partially or completely irrelevant/ redundant to the target concept. A relevant feature is neither irrelevant nor redundant to the target concept; an irrelevant feature does not affect the target concept in any way, and a redundant feature does not add anything new to the target concept (John, Kohavi, & Pfleger, 1994).

In many applications, the size of a dataset is so large that learning might not work as well before removing these unwanted features. Reducing the number of irrelevant/redundant features drastically reduces the running time of a learning algorithm and yields a more general concept. This helps

in getting a better insight into the underlying concept of a real-world classification. This paper intends to investigate how the performance of IDS can be improved by the usage of J48 Decision Tree algorithm for detection of intrusion on three types of attacks (Normal, Probing and U2R attacks) extracted from KDD'99 (Knowledge Discovery Database) dataset using gain ratio for feature reduction.

2. RELATED WORKS

According to Balogun et al.(2015), from their analysis, they revealed that feature selection using J48 (decision tree) algorithm, improved the performance of both K-means and Expectation Maximization (EM) algorithm on the datasets, though EM didn't perform well on the U2R and DOS datasets as compared to K-means. Their experiment recorded that there were general improvements of both algorithms on the dataset. The performances of the algorithms were well above average in terms of accuracy (correctly classified instances). Evaluating the results of the models on KDD dataset revealed EM algorithm is better than its K-means counterpart though not in all respect. Nevertheless, the results of both models on various categories of attack are credible and differ slightly in some cases. However, they strongly recommended further research be carried out on the effect of feature selection techniques in classification problems.

In the experiment carried out by Mabayoje, Akintola, Balogun, and Ayilara (2015), it was discovered noticed an improvement in the performance of the decision tree classifier in some categories of attack, that is, remote to local (98.31% for reduced dataset over 98% for full data set) and user to root(76.92% for reduced dataset over 75% for full dataset), in the case of denial of service and Normal categories, both methods gave same result(100% for both full and reduced data sets) and also there was some demeaning results in the category of probing attack (97.78% for reduced dataset over 99.49% for full dataset). They concluded that, feature selection can help improve the accuracy of classifiers.

Balogun, and Jimoh (2015) presented an overview of intrusion detection and a hybrid classification algorithm based on decision tree and K Nearest neighbour. The data set is first passed through the decision tree and node information is generated. Node information is determined according to the rules generated by the decision tree. This node information (as an additional attribute) along with the original set of attributes is passed through the KNN to obtain the final output. The key idea presented is to investigate whether the node information provided by the decision tree will improve the performance of the KNN. A performance evaluation was performed using a 10-fold cross validation technique on the individual base classifiers (decision tree and KNN) and the proposed hybrid classifier (DT-KNN) using the KDD Cup 1999 dataset on WEKA tool. Experimental results showed that the hybrid classifier (DT-KNN) gives the best result in terms of accuracy and efficiency compared with the individual base classifiers (decision tree and KNN).

Heba, Sherif and Mohamed (2012) designed a multi-layer intrusion detection model, aimed at improving the detection and classification rate accuracy and also achieving high efficiency. They made use of the Naïve Bayes, Multilayer Perceptron neural network (MLP), and C4.5 decision tree; gain ratio was used to best select features for getting high intrusion detection performance. The results they presented indicated that the proposed model achieved higher classification rate accuracy, and less false alarm rate than Naïve Bayes and MLP. They also pointed out that Gain Ratio enhanced the accuracy of User to Root (U2R) and Root to Local (R2L) for the three machine learning techniques significantly. The classification rate of MLP was high when using the whole 41 features in Denial of service (Dos) and Probe layers.

Purva and Priti (2013) developed application software for detecting intrusion through the usage of Multilayer Perceptron (MLP) algorithm based on Back Propagation. They proposed a system that not only detect attacks but also classify them in 6 groups with the accuracy of approximately 83% with the two hidden layers for the neural networks.

And also see how live detection of Internet Control Message Protocol (ICMP) attacks using Snort IDS.

Aida, Ahmed and Tamer (2010) in their work stated that an IDS's responsibility is to detect suspicious or unaccepted system and network activity and to alert a systems administrator to this activity. They evaluated the performance of nine NNs based classifiers, based on a selected group of features. They reveal in their result that; the Multilayer perceptron (MLPs) based classifier had about 99.63% true positive thereby having the best result.

Nguyen and Choi (2008), conducted a research on ten classification algorithms used for intrusion detection and evaluated their performances using the KDD99 dataset. Based on the attacks category, they chose the best algorithms and proposed a two classifier algorithm selection models.

Yogendra and Upendra (2012), studied and analyzed few data mining classification algorithms (Naïve Bayes, OneR, BayesNet, and Decision tree J48) in order to detect intrusions and thereafter compare their relative performances. They pointed out that J48 decision tree out-performed other three algorithms.

Moradi and Zulkernine (2010) proposed a neural network approach to intrusion detection; they used Multilayer Perceptron for detecting intrusion based on an off-line analysis approach. They focused on solving a problem of multiclass in which the type of attack is also detected by the neural network aside from classification of records in one of the two general classes - normal and attack.

Adsul, Danke, Jagdale, Chaudlhari, and Jadhav (2014) in their work, made a new approach of intrusion detection system based on artificial neural networks. They utilized Multilayer Perceptron (MLP) for detecting intrusion, and the designed system detect the attacks and classify them in six groups with the two hidden layers of neurons in the neural network. Their work is similar to that of (Devikrishna & Ramakrishna, 2014).

3. METHODOLOGY

The appropriate dataset attack type to be used will be selected and then check if the dataset selected is a full dataset or reduced dataset, if the dataset is full dataset then we train and test J48 Decision Tree with the full dataset using 10 fold cross validation, then if otherwise we perform feature reduction using Gain ratio to reduce the features into 11 attributes including 1label attribute, after that we train and test the J48 Decision Tree with the reduced dataset using 10 fold cross validation.

J48 (C4.5) builds decision trees from a set of training data, using the concept of information entropy. The training data is a set $S = S_1, S_2, S_3 \dots$ of already classified samples. Each sample S_i consists of a p -dimensional vector $(X_{1,i}, X_{2,i}, \dots, X_{p,i})$ where X_j represent attributes or features of the sample where $\{0 > i \geq j; i, j > 0\}$, as well as the class in which S_i falls.

At each node of the tree, the decision tree chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision, the J48 algorithm then recurs on the smaller sub lists.

The algorithm is shown below:

Input: training sample set T , the collection of candidate attribute list

Output: a decision tree

1. Create a root node N
2. IF T belongs to the same category C , then return N as a leaf node, and mark it as class C ;
3. IF attribute list is empty or the remainder samples of T is less than a given value, then return N as a leaf node, and mark it as the category which appears most frequently in attribute list, for each attribute, calculate its information gain ratio;
4. Suppose test attribute is the testing attribute of N , the test attribute = the attribute which has the highest information gain ratio in attribute list;

5. If testing attribute is continuous, then find its division threshold;
6. For each new leaf node grown by node N
 - {
 - Suppose T' is the sample subset corresponding to the leaf node.
 - If T' has only a decision category, then mark the leaf node as this category.
 - Else continue to implement J48_Tree (T' , T' _attributelist)
 - }
7. Calculate the classification error rate of each node, and then prune the tree.

The pseudo code for J48_Decision Tree decision tree algorithm (C4.5) is given below;

Algorithm Decision Tree (D, A, T)

If D contains only training examples of the same class c_j **C then**

Make T a leaf node labelled with c_j ,

elseif $A = 0$ **then**

Make T a leaf node labeled with c_j , which is the most frequent class in D

else // D contains examples belonging to a mixture of classes.

// we select a single attribute to partition D into subsets so that each subset is purer

$P_0 = \text{impurityEval} - 1(D)$;

For each attribute $A_i \{A_1, A_2, A_3, \dots, A_k\}$ **do**

$P_i = \text{impurityEval} - 2(A_i, D)$

End

```

        Select  $A_g\{A_1, A_2, A_3, \dots, A_k\}$  that gives the biggest
        impurity reduction, computed using  $P_0 - P_i$ 

    If  $P_0 - P_g < \text{threshold}$  then //  $A_g$  does not significantly reduce
    impurity  $P_0$ 

    Make  $T$  a leaf node labeled with  $c_j$  the most frequent class in  $D$ 

    else //  $A_g$  is able to reduce impurity  $p_0$ 

    Make  $T$  a decision node on  $A_g$ ,

    Let the possible values of  $A_g$  be  $V_1, V_2, V_3, \dots, V_m$ 

    Partition  $D$  into  $m$  disjoint subsets  $D_1, D_2, D_3, \dots, D_m$  based on the
     $m$  values of  $A_g$ 

    For each  $D_j$  in  $\{D_1, D_2, D_3, \dots, D_m\}$  do
        If  $D_j \neq \text{then}$ 
            Create a branch (edge) node  $T_j$  for  $V_j$  as a child node of  $T$ ;
        Decision Tree  $\{D_j, A - \{A_g\}, T_j\}$  //  $A_g$  is removed
        end
    end
end
end
end

```

Fig 3.1: Pseudo code for the (J48) Decision Tree Algorithm

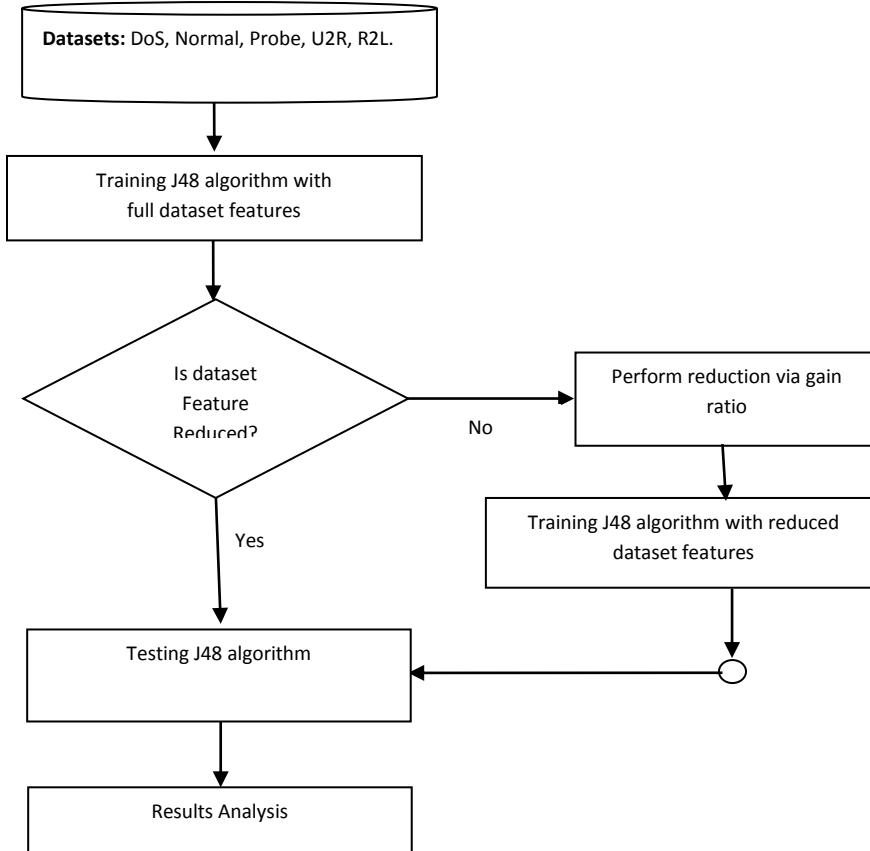


Fig 3.2: System activity diagram

Figure 3.2 shows the system activity diagram for detecting and classifying attacks.

The dataset used is 10% of KDD99 which is the mostly widely used dataset containing 42 features (with label). This dataset is being feed into the J48 classifier for training and testing.

The training and testing layer made used of cross validation technique (10 folds) which divided the dataset into 10 segments in which 9 segments are used for training and the last one for testing

The classifier layer involved the usage of decision tree algorithm for detecting and classifying intrusion.

Feature selection layer provided the removal of redundant and not important attributes in the dataset; gain ratio is used in this study. Feature selection is used in order to decrease the dimensionality of a dataset and perhaps may increase the accuracy and performance of the J48 decision tree classifier.

Result analysis layer provides the performance evaluation process for the decision tree classifier when being feed with all features as input and also when being feed with the reduced dataset.

3.1 Evaluation Set-Up

The experiments were carried out on a 64-bit Windows 7 operating system with 4GB of RAM and a Pentium (R) Dual -core CPU at 2.20GHz per core using the WEKA tool, developed by the University of Waikato via JAVA programming language, WEKA is a data mining system implementing various machine learning algorithms. For the purpose of this study the datasets that will be used are the U2R, DoS, NORMAL, Probing and R2L dataset in the KDD'99 dataset. To access the effectiveness of the algorithm, it was trained and tested using the 10% KDD '99 dataset with a 10-fold cross validation in Weka Environment. This method divides the dataset into 10 subsets, one of the 10 subsets is used as the test set while the remaining k-1 is used as training set, then the performance statistics are calculated across all the 10 subset which in turn provides a good clue of how sound the classifier works. The performance of the algorithm was evaluated on the feature reduced dataset and the original dataset.

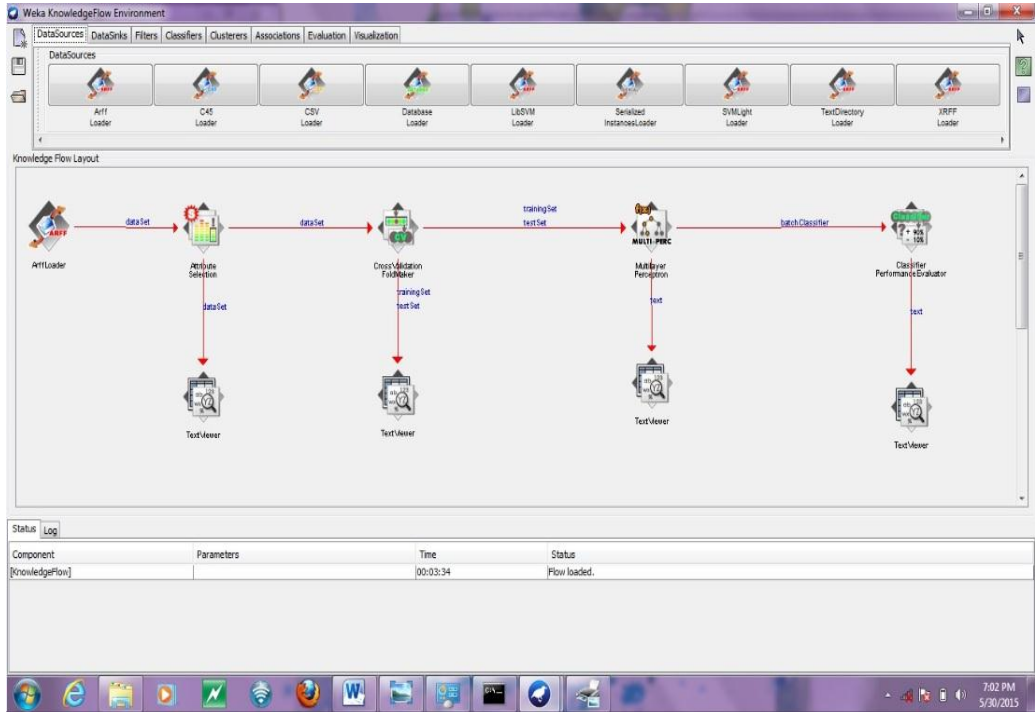


Fig 3.3: Knowledge Flow Representation of System architecture using Weka.

4. RESULTS AND DISCUSSION

This section presents the experimental result of feature selection from the full dataset based on Normal, DoS, Probe, U2R, R2L. Tables (I, II, III, IV and V) below show the summary of the Performance measurement of J48 Decision Tree Classifier on DOS, Normal, Probing R2L and U2R (full dataset) and also a detailed explanation of the performances

Table I: Performance evaluation of J48 Decision Tree Classifier on the full dataset

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
CORRECTLY CLASSIFIED INSTANCES (%)	100	100	99.4887	98.0462	75
INCORRECTLY CLASSIFIED INSTANCES (%)	0	0	0.5113	1.9538	25
KAPPA STATISTICS	1	1	0.9926	0.8874	0.5
MEAN ABSOLUTE ERROR	0	0	0.005	0.0018	0.0224
ROOT MEAN SQUARED ERROR	0	0	0.0205	0.0367	0.1294
RELATIVE ABSOLUTE ERROR	0	0	0.8252	10.9148	35.888
ROOT RELATIVE SQUARED ERROR	0	0	11.839	41.8572	77.7255

Table II: Performance measurement of J48 Decision Tree Classifier on the full dataset

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
TP RATE	1	1	0.995	0.98	0.75
FP RATE	0	0	0.002	0.043	0.092
PRECISION	1	1	0.995	0.976	0.781
RECALL	1	1	0.995	0.98	0.75
F-MEASURE	1	1	0.995	0.978	0.761
ROC AREA	1	0	0.998	0.978	0.853
Training Time	18.33secs	0.02secs	0.17secs	0.09secs	0secs

Table III: Performance evaluation of J48 Decision Tree Classifier on the reduced dataset

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
CORRECTLY CLASSIFIED INSTANCES (%)	100	100	97.7843	98.3126	76.9231
INCORRECTLY CLASSIFIED INSTANCES (%)	0	0	2.2157	1.6874	23.0769
KAPPA STATISTICS	1	1	0.9679	0.9014	0.5815
MEAN ABSOLUTE ERROR	0	0	0.0024	0.0021	0.0237
ROOT MEAN SQUARED ERROR	0	0	0.0362	0.0357	0.1216
RELATIVE ABSOLUTE ERROR	0	0	3.9425	12.3839	38.059
ROOT RELATIVE SQUARED ERROR	0	0	20.8699	40.7015	73.0542

Table IV: Performance measurement of J48 on the reduced dataset

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
TP RATE	1	1	0.978	0.983	0.769
FP RATE	0	0	0.007	0.06	0.201
PRECISION	1	1	0.977	0.981	0.753
RECALL	1	1	0.978	0.983	0.769
F-MEASURE	1	1	0.978	0.981	0.75
ROC AREA	1	0	0.996	0.949	0.791
Training Time	1.73secs	0.03secs	0.06secs	0.02secs	0secs

Table V: Accuracy of J48 and time taken to build models on the two datasets

CLASSIFIER	ATTACK TYPES				
	DO S	NORM AL	PROBING	R2L	U2R
J48 DT(reduced dataset) %	100	100	97.7843	98.3126	76.9231
Time taken to build model (s)	1.48	0.02	0.11	0.03	0.001
J48 DT (full dataset) %	100	100	99.4887	98.0462	75.00
Time taken to build model (s)	12.8 2	0.05	0.25	0.09	0.2

The results presented which are summarized in the Table V above, shows that we have 100% accuracy for full dataset and reduced dataset on DoS and Normal while Probing full dataset is 97.8%, Probing reduced dataset is 99.5%, U2R full dataset is 75%, reduced dataset is 76.9%, R2L full dataset is 98.0%, reduced dataset is 98.3%. It can be seen that dataset features reduction has a great influence in the accuracy of the performance evaluation and measurement of J48 decision tree classifier.

5. SUMMARY AND CONCLUSION

The field of intrusion detection has found a lot of researchers working tremendously on improving its standard. Various algorithms have being implemented; several hybridized algorithms have been developed for the purpose of complimenting the merits and demerits of both algorithms, not

forgetting that each algorithm possess separate knowledge and problem solving technique. Thus for the improvement of the performance of an algorithm, another algorithm can be used in conjunction with it for performing its functions.

Thus, this research work is based on performing feature selection using information gain ratio technique on the given dataset (KDD cup'99), and classifying the dataset using J48 Decision Tree algorithm (a supervised learning algorithm). The results are evaluated based on training time, detection rate (accuracy), false alarm rate, kappa statistics, ROC curve, correctly classified instances (%), incorrectly classified instances (%), TP rate, FP rate, precision, recall, f-measure, mean absolute error, root mean squared error, relative absolute error, root relative squared error . From the results, it can be concluded that there is a positive improvement in the classification of various attacks by the J48 Decision Tree algorithm except for Probing classification which shows a demeaning result but an obvious difference is seen in the User to Root classification and the effect of feature selection is obviously seen as it improved the performance of the classification algorithm (Decision Tree).

REFERENCES

- Adsul, A. P., Danke, P., Jagdale, M., Chaudhari, K., & Jadhav, S. (2014), "Attacks Classification in Network Intrusion Detection System Using ANN", *IJAIEEM*, Vol. 3, Issue 4, April 2014
- Aida, O. A., Ahmed, S., & Tamer R. (2010), "Multilayer perceptrons networks for an Intelligent Adaptive intrusion detection system", *IJCSNS*, Vol. 10, No. 2, February 2010.
- Balogun, A. O., Balogun, A. M., Adeyemo, V. E., & Sadiku, P. O. (2015): A Network Intrusion Detection System: Enhanced Classification via Clustering Model. *Computing, Information System Development Informatics & Allied Research Journals*. 6(4):53-58.
- Balogun, A. O., & Jimoh, R.G. (2015): Anomaly Intrusion Detection Using a Hybrid of Decision Tree and K-Nearest Neighbor. *Journal of Advances in Scientific Research & Applications (JASRA)*. 2(1): 67-74.

- Balogun, A. O., Mabayoje M.A., Salihu, S. & Arinze, S.A. (2015): Enhanced Classification Via Clustering Using Decision Tree for Feature Selection. *International Journal of Applied Information Systems (IJ AIS)*. 9(6):11-16.
- Devikrishna, K. S., & Ramakrishna, B. B. (2013): "An Artificial Neural Network based Intrusion Detection System and Classification of Attacks" Vol. 3, Issue 4, Jul-Aug 2013.
- D'silva, M., & Vora, D. (2013) "Comparative Study of Data Mining Techniques to Enhance Intrusion Detection". *IJERA*, Vol. 3, Issue 1, January – February 2013.
- Hashemi, V. M., Muda, Z., & Yassin, W. (2013). "Improving intrusion detection using genetic algorithm". *Inform. Technol. J.*, 12: 2167-2173.
- Heba, E. I., Sherif, M. B., & Mohamed A. S. (2012) "Adaptive Layered Approach using Machine Learning Techniques with Gain Ratio for Intrusion Detection Systems". *International Journal of Computer Applications IJCA*, Volume 56 – No.7, October 2012.
- Hemant, P., Sarkhedi, B., & Vaghamshi, H. (2013) "Intrusion Detection in Data Mining with Classification Algorithm". *IJAREEIE*, Vol. 2, Issue7, July 2013.
- John, G. H., Kohavi, R. & Pflieger, K. (1994). "Irrelevant features and the subset selection problem". In: Proceedings of the Eleventh International Conference on Machine Learning, 1994.
- Karim, M., & Rahman, R. M. (2013). "Decision Tree and Naïve Bayes Algorithm for Classification and Generation of Actionable Knowledge for Direct Marketing". *Journal of Software Engineering and Applications*. Vol.6, No.4, DOI:10.4236/jsea.2013.64025
- KDD cup 1999 Computer network intrusion detection. <http://www.kdd.org/kdd-cup-1999-network-intrusion-detection>.
- Mabayoje M.A., Akintola, A. G., Balogun, A. O & Ayilara, O. (2015): Gain Ratio and Decision Tree Classifier for Intrusion Detection. *International Journal of Computer Applications (IJCA)*. 126(1):56-59
- Mabayoje, M. A., Balogun, A. O., Salihu, S. A., & Oladipupo, K. R. (2015): Comparative Analysis of Selected Supervised Classification Algorithms. *African Journal of Computing & ICTs*. 8(3); 35-42.

- Moradi, M. and Zulkernine, M (2004). Neural Network based System for Intrusion Detection and Classification of Attacks”, *Proceedings of IEEE International Conference on Advances in Intelligent Systems – Theory and Applications*, Luxembourg, Vol. 148, pp. 1-6, 2004.
- Mukkamala, S., Sung, A. H., & Abraham, A (2005). “Intrusion Detection Using Ensemble of Soft computing Paradigms”. *Journal of Networks and Computer applications*, 28: 167-182. doi:10.1016/j.jnca.2004.01.003
- Mukkamala, S., Sung, A.H., Abraham, A., & Ramos, V. (2004). “Intrusion detection systems using adaptive regression splines”. In: SerucaI, Filipe, J., Hammoudi, S., Cordeiro, J., editors. *Proceedings of the 6th international conference on enterprise information systems, ICEIS’04, vol.3, Portugal. 2004b. p.26–33*[ISBN: 972-8865-007].
- Nguyen, H. A., & Choi, D. (2008), “Application of Data Mining to Network Intrusion Detection: Classifier Selection Model”, *APNOMS ’08 Proceedings of the 11th Asia-Pacific Symposium on Network Operations and Management: Challenges for Next Generation Network Operation and Service Management*, pp. 399 – 408.
- Patcha, A. & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51: 3448-3470
- Patel, R., Amit T., Amit G. (2012) “A Survey and Comparative Analysis of Data Mining Techniques for Network Intrusion Detection Systems”. *IJSCE, Volume-2, Issue-1, March 2012*.
- Purva, A., & Priti, S. (2013), “Detecting and Classifying Attacks in Network Intrusion Detection System Using Multi-layer Perceptron Based on Artificial Neural Network” *IJARCSSE, Vol. 3, Issue 6, June 2013*.
- Quinlan, J. R., (1986) *Induction of Decision Trees*, Machine Learning 1: pp.81-106, *Kluwer Academic Publishers, Boston*.
- Yogendra K. J. & Upendra (2012), “An efficient Intrusion Detection Based on Decision Tree Classifier Using Feature Reduction”. *IJSRP, Vol. 2, Issue 1, January 2012*.