AN ENSEMBLE APPROACH BASED ON DECISION TREE AND BAYESIAN NETWORK FOR INTRUSION DETECTION

[1] Balogun A. O., [1] Balogun A. M., [1] Sadiku P. O., [2] Amusa L. B.

[1] Department of Computer Science, University of Ilorin, Ilorin, Nigeria
[2] Department of Statistics, University of Ilorin, Ilorin, Nigeria

Corresponding author: BALOGUN A. O., balogun.ao1@unilorin.edu.ng, bharlow058@gmail.com

ABSTRACT: *This paper presents an overview of intrusion detection and a hybrid classification algorithm based on ensemble method (stacking) which uses decision tree (J48) and Bayesian network as base classifiers and functional tree algorithm as the meta-learner. The data set is passed through the decision tree and node Bayesian network for classification. The meta-learner (Functional tree classifier) will then select the value of the base classifier that has the higher accuracy based on majority voting. The key idea here is to always pick the value with higher accuracy since both base classifier (decision tree and Bayesian network) will always classify all instances. A performance evaluation was performed using a 10-fold cross validation technique on the individual base classifiers (decision tree and Bayesian network) and the ensemble classifier (DT-BN) using the KDD Cup 1999 dataset on WEKA tool. Experimental results show that the hybrid classifier (DT-BN) gives the best result in terms of accuracy and efficiency compared with the individual base classifiers (decision tree and BN). The decision tree gave a result of (99.9974% for DoS, 100% for Normal, 98.8069% for probing, 97.6021% for U2R and 73.0769% for R2L), the Bayesian network (99.6410% for DoS, 100% for Normal, 97.1756% for probing, 97.0693% for U2R and 69.2308% for R2L),while the ensemble method gave a result of (99.9977% for DoS, 100% for Normal, 98.8069% for probing, 97.6909% for U2R and 73.0769% for R2L).*

KEYWORDS*: Network security, Intrusion detection system, classifiers, Bayesian network, Functional tree Decision Tree, meta-learner.*

1.0 INTRODUCTION

Traditional protection techniques such as user authentication, data encryption, avoiding programming errors and firewalls are used as the first line of defense for computer security. If a password is weak and is compromised, user authentication cannot prevent unauthorized use, firewalls are vulnerable to errors in configuration and suspect to ambiguous or undefined security policies ([Sum97]). They are generally unable to protect against malicious mobile code, insider attacks and unsecured modems. Programming errors cannot be avoided as the complexity of the system and applications of ware is evolving rapidly leaving behind some exploitable weaknesses. Consequently, computer systems are likely to remain unsecured for the foreseeable future. Therefore, intrusion detection is required as an additional wall for protecting systems despite the prevention techniques. Intrusion detection is useful not only in detecting successful intrusions, but also in monitoring attempts to break security, which provides important information for timely countermeasures ([H+90; S+96]). Intrusion detection is classified into two types: misuse intrusion detection and anomaly intrusion detection.

The goal of intrusion detection is to detect security violations in information systems. Intrusion detection is a passive approach to security as it monitors information systems and raises alarms when security violations are detected. Examples of security violations include the abuse of privileges or the use of attacks to exploit software or protocol vulnerabilities. Traditionally, intrusion detection techniques are classified into two broad categories: misuse detection and anomaly detection ([Mou97]). Misuse detection works by searching for the traces or patterns of well-known attacks. Clearly, only known attacks that leave characteristic traces can be detected that way. Anomaly detection, on the other hand, uses a model of normal user or system behavior and ages significant deviations from this model as potentially malicious. This model of normal user or system behavior is commonly known as the user or system profile. Strength of anomaly detection is its ability to detect previously unknown attacks. Additionally, intrusion detection systems (IDSs) are categorized according to the kind of input information they analyze. This leads to the distinction between host-based and network-based IDSs. Host-based IDSs analyze host-bound audit sources such as operating system audit trails, system logs, or application logs. Network-based IDSs analyze network packets that are captured on a network.

Data mining has attracted a great deal of attention in the information industry and in society as a whole in recent years, due to the wide availability of huge

amounts of data and the imminent need for turning such data into useful information and knowledge ([HK00]). The information and knowledge gained can be used for applications ranging from market analysis, fraud detection, and customer retention, to production control and science exploration. Data mining can be viewed as a result of the natural evolution of information technology. The most commonly accepted definition of "data mining" is the discovery of "model" for data. A "model" however, can be one of several things. Statisticians were the first to use the term "data mining". Originally, "data mining" or "data dredging" was a derogatory term referring to attempts to extract information that was not supported by the data. Today, "data mining" has taken on a positive meaning. Now, statisticians view data mining as the construction of a statistical model, that is, an underlying distribution from which the visible data is drawn. There are some who regard data mining as synonymous with machine learning. There is no question that some data mining appropriately uses algorithms from machine learning. Machine-learning practitioners use the data as a training set, to train an algorithm of one of the many types used by machine-learning practitioners, such as Bayes nets, Support Vector Machines, decision trees, hidden Markov models, and many others. There are situations where using data in this way makes sense. The typical case where machine learning is a good approach is when we have little idea of what we are looking for in the data. On the other hand, machine learning has not proved successful in situations where we can describe the goals of the mining more directly.

In recent years, a growing number of research projects have applied data mining to intrusion detection. However, the approach used in this study is to carry a comparative study on selected data mining algorithms used in intrusion detection system. The intention of this study is to give the reader a broad overview of the data mining algorithms in intrusion detection system. Due to the increasing incidents of cyber-attacks, building effective and efficient intrusion detection systems are important for protecting and detecting such attacks, and yet it remains an elusive goal and a challenge ([M+05]). Many of recent researches of IDS have proposed anomaly detection to detect novel attacks ([PP07; H+13; BJ15; B+15; M+16]). Many of these approaches resulted in high detection rate and accuracy ([A+13]). However, majority of them encounter high false alarm rates ([M+04]). As the result of falsely classification of normal connections as attack, authentic users cannot access to the network ([M+04]). Therefore, IDS research area is in desperate need of focusing on false alarm to properly identify such intrusions and further enhancement of the algorithms used in IDS ([A+13]). Also, quite a number of ensemble methods have also been developed to this effect but the problem of accurate classification still lingers ([GC12]). All these serve as the main motivation for this research.

2.0 RELATED WORKS

The main idea of ensemble methodology is to combine a set of models, each of which solves the same original task, in order to obtain a better composite global model, with more accurate and reliable estimates or decisions than can be obtained from using a single model. The idea of building a predictive model by integrating multiple models has been under investigation for a long time ([Lio09]). Also, the author further opined that the way of combining the classifiers may be divided into two main groups: simple multiple classifier combinations and meta-combiners. The simple combining methods are best suited for problems where the individual classifiers perform the same task and have comparable success. However, such combiners are more vulnerable to outliers and to unevenly performing classifiers. On the other hand, the meta-combiners are theoretically more powerful but are susceptible to all the problems associated with the added learning (such as over-fitting, long training time).

According to Giovanni and Elder ([GE10]), there is a way to improve model accuracy that is easier and more powerful than judicious algorithm selection: one can gather models into ensembles. Building an ensemble consists of two steps which are constructing varied models and combining their estimates. One may generate component models by, for instance, varying case weights, data values, guidance parameters, variable subsets, or partitions of the input space. Combination can be accomplished by voting, but it is primarily done through model estimate weights, with gating and advisor perceptron as special cases.

Hui Zhao ([Hui13]) reveals that intrusion detection data often have some characteristics such as nonlinearity, higher dimension, much redundancy and noise, and partial continuous-attribute. The author presented a new ensemble algorithm to improve intrusion detection precision. Firstly, it generates multiple training subsets in difference by using bootstrap technology. Then using neighborhood rough sets with different radiuses to make attribute reduction in these subsets, obtained the training subsets with greater difference, while Particle Swarm Optimization is used to optimize parameters of support vector machine in order to get

base classifiers with greater difference and higher precision. Finally, the above base classifiers were inter-grinded by weighted synthesis method.

Chaurasia and Jain ([CJ14]) proposed an ensemble classifier technique for intrusion detection by demonstrating ensemble of different classifiers for increasing the accuracy. The authors used K-NN and NN classifiers, besetting the two classifiers for misclassification data to enhance the detection rate, thereby evaluated the performances of each classifier individually and also the performance of the bagging of multiple classifiers on the KDD cup'99 dataset. Bagging provided better results for evaluation of KDD cup'99 for 5 classes (normal, dos, probe, u2r and r2l).

Intrusion detection systems based on the human immunological system have been proposed in Esponda et al. ([E+04]), Hofmeyr and Forrest ([HF99]). Forrest et al. ([F+04]) proposed a formal frame work for anomaly detection in computer systems, inspired by the characteristics of the natural immune system. Hofmeyr and Forrest ([HF99]) applied the concepts derived from natural immune system to design and test an artificial immune system to detect network intrusion. They specifically mentioned 4 important characteristics of natural immune system that they think define immunity. They are diversity, distributed nature, error tolerance and dynamic nature. They designed the detector analogous to the T and B-Lymphocytes that are found in the human immunological system.

Govindarajan and Chandrasekaran ([GC12]) presented new hybrid classification method using classifiers in a heterogeneous environment using arcing classifier and their performances are analyzed in terms of accuracy. A Classifier ensemble is designed using a Radial Basis Function (RBF) and Support Vector Machine (SVM). Here, modified training sets are formed by resampling from original training set; classifiers constructed using these training sets and then combined by voting. Also, the authors exposed that Breiman introduced Arcing ('Adaptive Resampling and Combining') as a generalization of Bagging and Boosting. In Arcing, as Breiman puts it, "modified training sets are formed by resampling from the original training set, classifiers constructed using these training sets and then combined by voting. Arcing is a more complex procedure. Again, multiple classifiers are constructed and vote for classes. But the construction is sequential, with the construction of the (k+1)st classifier depending on the performance of the k previously constructed classifiers. At the start of each construction, there is a probability distribution {p(n)} on the cases in the training set. A training set T' is constructed by sampling N times from this distribution. Then the probabilities are updated depending on how the cases in T are classified by C(x,T'). A factor $\beta>1$is defined which depends on the misclassification rate. If the nth case in T is misclassified by C(x,T'), then put weight $\beta p(n)$ on that case. Otherwise define the weight to be p(n). Now divide each weight by the sum of the weights to get the updated probabilities for the next round of sampling. After a fixed number of classifiers have been constructed, voting is done for the class.

3.0 METHODOLOGY

The proposed system is an ensemble based intrusion detection system aimed for providing a better security on a computer or an arbitrary network. All step-by-step experiments were done by applying the selected classification algorithms on the KDD 99 dataset. More so, the selection of comprehensive sets of classifier algorithms was chosen for the ensemble method, which included some distinct but widely used classifier algorithms so as to cover classification algorithms from Naïve Bayes, decision tree and artificial neural networks. However, the preprocessing stage will involve fragmenting the KDD 99 dataset into various categories of attacks, performing feature selection using principal component analysis technique which is an advanced feature selection algorithm. The data mining software used for carrying out this research is "WEKA" – (Waikato Environment for Knowledge Analysis) tool and the algorithms that were ensemble via stacking method of ensemble are Bayesian Network (BN) and Decision tree (J48) as the base classifiers while Functional Tree (FT) serves as the meta-learner. However, detailed tables of results having the performance of the selected classifiers will be presented and also a table for comparison of their performances. The table for comparison will hold the results of the performance of each individual base classifiers against the performance of the Ensemble method via stacking which dataset's features will have being filtered using the principal component analysis technique.

3.1 PRINCIPAL COMPONENT ANALYSIS FOR FEATURE SELECTION

Principal component is a multivariate statistics method and its basic idea is to seek a projection that best represents the data in a least-square sense. It a linear transform technology that seek directions in feature space that represents the data in a sum-squared error sense ([GZZ08]). Principal component analysis (PCA) has been widely applied in data mining to investigate data structure. In PCA, new orthogonal variables (latent variables or principal

components) are obtained by maximizing variance of the data. The number of the latent variables (factors) is much lower than the number of original variables, so that the data can be visualized in a low-dimensional PC space. While PCA greatly reduces the dimensionality of the space, it does not reduce the number of the original variables, as it uses all the original variables to generate the new latent variables (principal components). For interpretation purposes or future investigations, it would often be very useful to reduce the number of variables. Feature (variable) selection can be achieved either by choosing Informative variables or discarding redundant variables. Several methods exist and most of them perform feature reduction using stepwise forward and/or backward techniques.

3.2 BAYESIAN NETWORK ALGORITHM

The Bayesian Network (BN) is a powerful knowledge representation and reasoning algorithm under conditions of uncertainty. A Bayesian network B = (N, A, Θ) is a Directed Acyclic Graph (DAG) (N, A) where each node n ∈ N represents a domain variable (e.g. a data set attribute or variable), and each arc a ∈ A between nodes represents a probabilistic dependency among the variables, quantified using a conditional probability distribution (CP table) $\Theta_i \in \Theta$ for each node n_i. A BN can be used to compute the conditional probability of one node, given values assigned to the other nodes. Many Bayesian network structure learning algorithms have been developed. These algorithms generally fall into two groups, search and scoring based algorithms and dependency analysis based algorithms. Although some of these algorithms can give good results on some benchmark data sets, there are still several problems such as node ordering requirement, lack of efficiency and lack of publicly available learning tools (Neapolitan, 1990). In order to resolve these problems, two types of algorithms have been developed in the area of Bayesian network structure learning. Type 1 deals with a special case where the node ordering is given, which requires O (N^2) Conditional Independence (CI) tests and is correct given that the underlying model is DAG faithful. Type 2 deals with the general case and requires O (N^4) CI tests and is correct given that the underlying model is monotone DAG faithful.

3.3 DECISION TREE (J48) ALGORITHM

Decision trees are a way of representing a series of rules that lead to a class or value. A decision tree is a tree structure consisting of internal and external nodes connected by branches. An internal node is a decision making unit that evaluates a decision function to determine which child node to visit next. The external node, on the other hand, has no child nodes and is associated with a label or value that characterizes the given data that leads to its being visited. However, many decision tree construction algorithms involve a two – step process. First, a very large decision tree is grown. Then, to reduce large size and over-fitting the data, in the second step, the given tree is pruned. The pruned decision tree that is used for classification purposes is called the classification tree. J48 as a type of decision tree builds the decision tree from labeled training data set using information gain and it examines the same that results from choosing an attribute for splitting the data. To make the decision the attribute with highest normalized information gain is used. Then the algorithm recurs on smaller subsets. The splitting procedure stops if all instances in a subset belong to the same class. Then the leaf node is created in a decision tree telling to choose that class.

3.4 FUNCTIONAL TREE ALGORITHM

FT combines a standard univariate decision tree, such as C4.5, with linear functions of the attributes by means of linear regressions. While a univariate decision tree uses simple value tests on single attributes in a node, FT can use linear combinations of different attributes in a node or in a leaf. In the constructive phase a function is built and mapped to new attributes. A model is built using the constructor function. This is done using only the examples that fall at this node. Later, the model is mapped to new attributes. The constructor function should be a classifier or a regresssor depending on the type of the problem. In the former the number of new attributes is equal to the number of classes, in the latter the constructor function is mapped to one new attribute. Each new attribute is computed as the value predicted by the constructed function for each example. In the classification setting, each new attribute value is the probability that the example belongs to one class given by the constructed model. The merit of each new attribute is evaluated using the merit-function of the univariate tree, and in competition with the original attributes.

3.5. PROPOSED SYSTEM ARCHITECTURE

The figure 3.1 shows the proposed architecture for detecting and classifying attacks.

The dataset used was 10% of KDD99 which is the mostly widely used dataset, containing 42 features (with label) and 420,00 instances. This dataset is

being feed into the ensemble method and the individual classifiers 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 individual classifier (Decision tree and Bayesian network) and the ensemble method (DT-BN) for detecting and classifying intrusions.

Feature selection layer provided the removal of redundant and not important attributes in the dataset. Feature selection is used in order to decrease the dimensionality of a dataset and increase its accuracy and performance of the classifiers. Principal component analysis will be used for the feature selection.

Result analysis layer provide the performance evaluation process for the base classifiers and the ensemble method when being feed with the dataset as input.

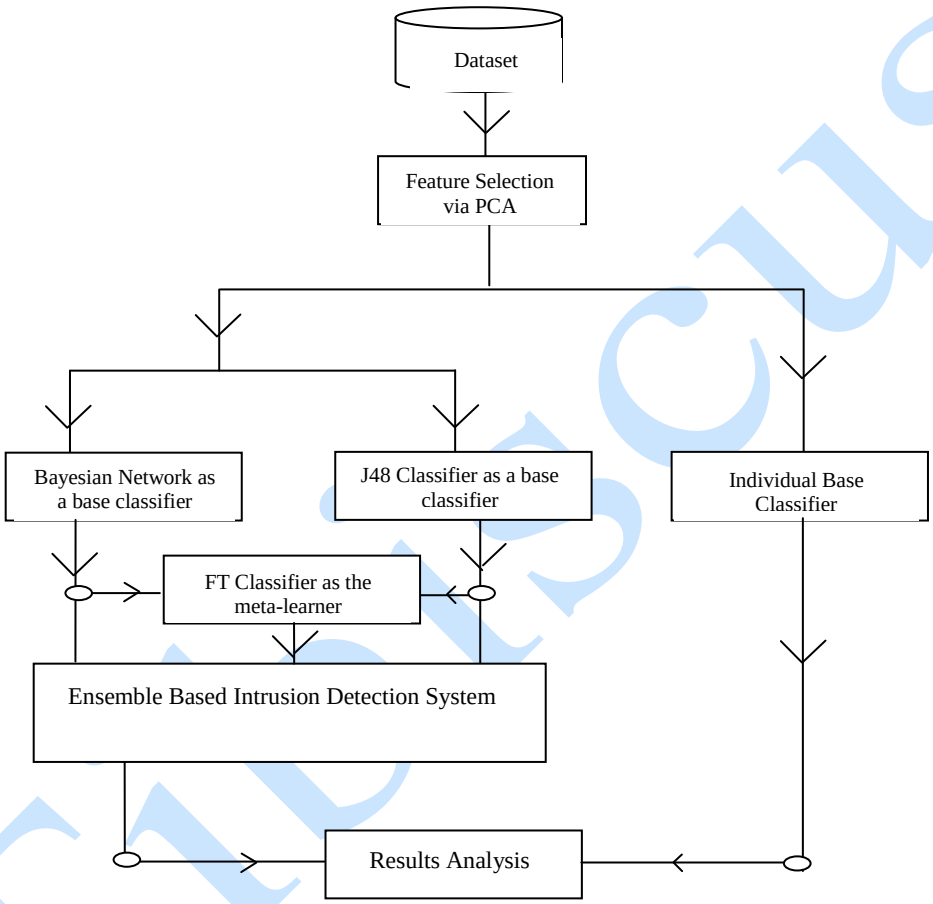

Figure 3.1: Ensemble System Architecture

3.6 EVALUATION SETUP

The experiments were carried out on a HP probook 6470b laptop with the following configurations Intel(R) Core(TM)i5-3230M, CPU 2.60GHz, 6GB RAM (5.55 GB usable), 64-bit operating system whose platform is Microsoft Windows7 Professional (Service Pack 1). The latest Weka – an oen source machine learning package was used for setting up the experimental and evaluation environment (Weka 3.6.11). Weka is a software that holds machine learning algorithms for data mining tasks containing tools for visualization, data preprocessing, regression, classification, association rules, and clustering.

3.7 PERFORMANCE COMPARISON

The performance of the ensemble method on each dataset i.e. the full (containing all the features and the reduced dataset, will be evaluated and measured via the following parameters: incorrectly classified instances (%),correctly classified instances (%),root mean squared error, relative absolute error, kappa statistics, root relative squared error and measured via the following parameters: TP (True Positive) rate, FP (False Positive) rate, Precision, Recall, F-Measure and TT (Training Time of the algorithm on each dataset), and AA (Average Accuracy = Total correctly classified instances/Total instances).

4.0 ANALYSIS OF RESULTS

The feature extracted from the original KDD '99 cup dataset based on principal component analysis technique were fed as input to the base classifiers, that is, Bayesian Network and Decision tree (J48) algorithms, and the training and testing of the stacking ensemble method was done using 10-fold cross validation technique. The results of the base classifiers which is the level-0 model were supplied as input to the meta-learner (FT algorithm) that is, the level-1 model so that the meta-model can combine the inputs and make the final prediction.

The Tables I, II, III, IV, V, VI, VII, and VIII displays the performance of the ensemble method (DT-BN) and the individual classifiers (Decision tree and Bayesian Network) based on the two distinct dataset mentioned earlier, and the table V is derived from all the previous tables.

From these results, it can be concluded that the ensemble method performance was greater than the performance of a base classifier or was at least as the performance of a base classifier.

Table I: Performance evaluation of Bayesian network on the datasets

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
CORRECTLY CLASSIFIED INSTANCES (%)	99.641	100	97.1756	97.0693	69.2308
INCORRECTLY CLASSIFIED INSTANCES (%)	0.359	0	2.8244	2.9307	30.7692
KAPPA STATISTICS	0.9913	1	0.9593	0.8428	0.4519
MEAN ABSOLUTE ERROR	0.0003	0	0.0032	0.0027	0.0395
ROOT MEAN SQUARED ERROR	0.0135	0	0.0436	0.0442	0.1387
RELATIVE ABSOLUTE ERROR	0.9369	50.0068	5.3331	16.0851	63.3818
ROOT RELATIVE SQUARED ERROR	10.0702	50.0068	25.1595	50.4075	83.278

Table II: Performance measurement of Bayesian network on the datasets

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
TP RATE	0.996	1	0.972	0.971	0.692
FP RATE	0	0	0.007	0.034	0.24
PRECISION	1	1	0.984	0.982	0.644
RECALL	0.996	1	0.972	0.971	0.692
F-MEASURE	0.998	1	0.979	0.975	0.666
ROC AREA	1	0	0.999	0.995	0.834

Table III: Performance evaluation of Decision tree (J48) on the datasets

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
CORRECTLY CLASSIFIED INSTANCES (%)	99.9977	100	98.8069	97.6909	73.0769
INCORRECTLY CLASSIFIED INSTANCES (%)	0.0023	0	1.1931	2.3091	26.9231
KAPPA STATISTICS	0.9999	1	0.9827	0.8682	0.5695
MEAN ABSOLUTE ERROR	0	0	0.0012	0.0023	0.0274
ROOT MEAN SQUARED ERROR	0.0014	0	0.0313	0.0414	0.1519
RELATIVE ABSOLUTE ERROR	0.0058	0	2.0313	13.4825	44.0217
ROOT RELATIVE SQUARED ERROR	1.0619	0	18.0609	47.1529	91.2052

Table IV: Performance measurement of Decision tree (J48) on the datasets

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
TP RATE	1	1	0.988	0.977	0.731
FP RATE	0	0	0.004	0.069	0.115
PRECISION	1	1	0.988	0.98	0.776
RECALL	1	1	0.988	0.977	0.731
F-MEASURE	1	1	0.988	0.978	0.747
ROC AREA	1	0	0.993	0.932	0.774

Table V: Performance measurement of Ensemble methods (DT-BN) on the datasets

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
CORRECTLY CLASSIFIED INSTANCES (%)	99.9974	100	98.8069	97.6021	73.0769
INCORRECTLY CLASSIFIED INSTANCES (%)	0.0026	0	1.1931	2.3979	26.9231
KAPPA STATISTICS	0.9999	1	0.9827	0.8611	0.5156
MEAN ABSOLUTE ERROR	0	0	0.0014	0.0024	0.0287
ROOT MEAN SQUARED ERROR	0.0014	0	0.0307	0.041	0.1352
RELATIVE ABSOLUTE ERROR	0.0074	0	2.3893	14.1827	46.1206
ROOT RELATIVE SQUARED ERROR	1.0656	0	17.7293	46.6988	81.1894

Table VI: Performance measurement of Decision tree (J48) on the datasets

PARAMETERS	DOS	NORMAL	PROBING	R2L	U2R
TP RATE	1	1	0.998	0.976	0.731
FP RATE	0	0	0.004	0.094	0.232
PRECISION	1	1	0.988	0.974	0.697
RECALL	1	1	0.988	0.976	0.731
F-MEASURE	1	1	0.988	0.975	0.706
ROC AREA	1	0	0.995	0.971	0.851

Table VII: Accuracy of the Ensemble method (DT-BN) and the Individual classifier on the datasets

CLASSIFIER	ATTACK TYPES				
	DOS	NORMAL	PROBING	R2L	U2R
BAYESIAN NETWORK	99.641	100	97.1756	97.0693	69.2308
J48	99.9974	100	98.8069	97.6021	73.0769
ENSEMBLE	99.9977	100	98.8069	97.6909	73.0769

Table VIII: False positive rate of the Ensemble method (DT-BN) and the individual classifiers

CLASSIFIER	ATTACK TYPES				
	DOS	NORMAL	PROBING	R2L	U2R
BN	0	0	0.007	0.034	0.24
J48	0	0	0.004	0.69	0.115
ENSEMBLE	0	0	0.004	0.094	0.232

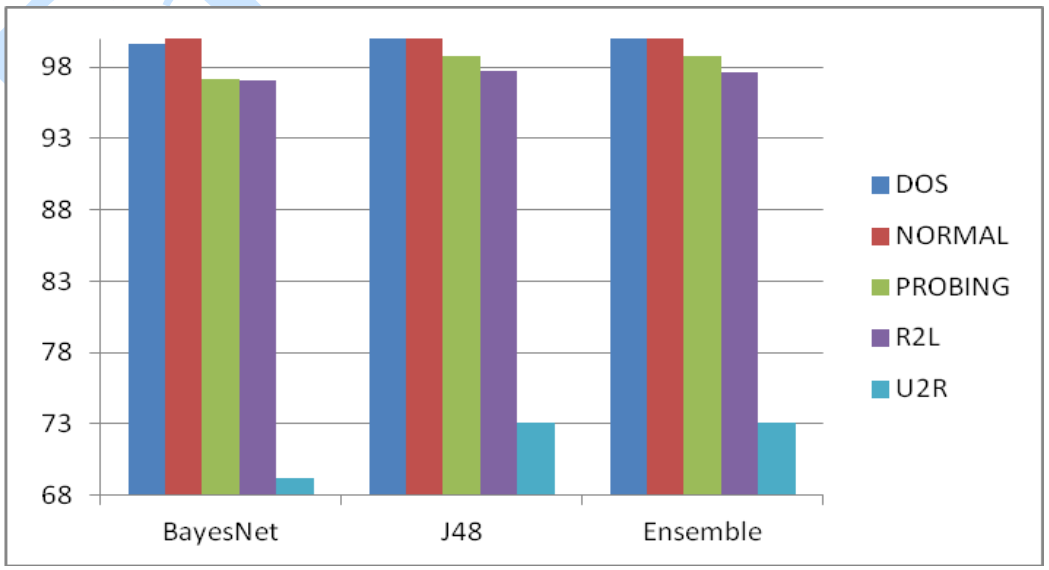

Figure 4.1: Graphical representation of accuracies of various models

The above chart graphically depicts the accuracy of the correctly classified instance of attack performed by each algorithm.

5.0 CONCLUSIONS AND FUTURE WORKS

Deducing fact from the analysis, the empirical result showed that the ensemble method accompanied with principal component analysis technique for feature selection performed better that each individual base classifiers in the classification of various attack. The overall performance of the ensemble algorithm based on stacking has the best result in most cases or at least as the performance of each base classifier on some dataset, though the input fed into the individual base classifiers and even the ensemble method was preprocessed.

From this research, the ensemble method approach to intrusion detection is a better way of developing an intrusion detection system as it combines different machine learning algorithms together, each one complimenting the other(s). Based on the ensemble method used which allows combination of algorithm, the researcher recommends for future work combination of more than two classification algorithm. More so, The researcher also recommend that this research can be furthered by using other feature selection techniques and also various feature selection techniques can be combined for the purpose of reducing the dimensionality of the dataset (which will be relative to the dataset used) before carrying out the classification process. Also, the authors recommend that this research work can be implemented by developing software using the algorithms for the purpose of detecting intrusion.

REFERENCES

[AB13] **B. Alexandre, G. Björn** - *Ensemble of Decision Trees for Network Intrusion Detection Systems*. IJAS, Vol 6 No 1&2. 2013.

[AIA13] **A. Ajayi, S. A. Idowu, A. Anyaehie** - *Comparative study of selected data mining algorithms used for intrusion detection*. International Journal of Soft computing Engineering (IJSCE), 2013.

[BJ15] **A. O. Balogun, R. G. Jimoh** - *Anomaly Intrusion Detection Using An Hybrid Of Decision Tree And K-Nearest Neighbor.* Journal of Advances in Scientific Research & Applications (JASRA). 2(1): 67-74. 2015.

[BL97] **M. J. A. Berry, G. Lino** - *Data Mining Techniques.* John Wiley and Sons, Inc. 1997.

[B+15] **A. O. Balogun, A. M. Balogun, V. E. Adeyemo, P. O. Sadiku** - *A Network Intrusion Detection System: Enhanced Classification via Clustering Model*. Computing, Information System Development Informatics & Allied Research Journals. 6(4):53-58. 2015.

[EG08] **J. M. Eitel, K. T. Giri** - *A Comparative Study Of Data Mining Algorithms For Network Intrusion Detection In The Presence Of Poor Quality Data*. ICIQ-03, 2008.

[EFH04] **F. Esponda, S. Forrest, P. Helman** - *A formal framework for positive and negative detection*. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 2004; 34(1).

[F+96] **U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy** - *Advances in Knowledge Discovery and Data Mining.* AAAI Press/MIT Press.1996.

[Gov14] **M. Govindarajan** - *Hybrid Intrusion Detection Using Ensemble of Classification Methods*. International Journal of Computer Network and Information Security, 2014, 2, 45-53.

[GC12] **M. Govindarajan, R. M. Chandrasekaran** - *Intrusion Detection using an Ensemble of Classification Methods*. WCECS 2012 Vol. I,.2012.

[GJ10] **S. Giovanna, F. E. John** - *Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions*. Chicago: Morgan and Claypool Publishers.2010.

[GZZ08] **Y. Gu, B. Zhou, J. Zhao** - *PCA-ICA Ensembled Intrusion Detection System by Pareto-Optimal Optimization*. Information Technology Journal, 7: 510-515, 2008.

[Hui13] **Z. Hui** - *Intrusion Detection Ensemble Algorithm based on Bagging and Neighborhood Rough Set.* IJSIA, Vol. 7. No.5 (2013), pp.193 – 204, 2013.

[HF99] **S. A. Hofmeyr, S. Forrest** - *Immunity by design: an artificial immune system*. In: Proceedings GECCO Conference, 1999.

[HK00] **J. Han, M. Kamber** - *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publisher, 2000.

[HMY13] **V. M. Hashemi, Z. Muda, W. Yassin** - *Improving Intrusion Detection Using Genetic Algorithm.* Information Technology Journal, 12: 2167-2173, 2013.

[H+90] **R. Heady, G. Luger, A. Maccabe, M. Servilla** - *The architecture of a network level intrusion detection system*. Technical Report, Department of Computer Science, University of New Mexico.

[IEM11] **H. W. Ian, F. Eibe, A. H. Mark** - *Data Mining: Practical Machine Learning Tools and Techniques (3rd edition)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.

[JMS13] **V. Jaiganesh, S. Mangayarkarasi, P. Sumathi** - *Intrusion Detection Systems: A Survey and Analysis of Classification Techniques*. IJARCCE, Vol. 2, Issue 4, 2013.

[KJ12] **G. Krzystof, N. Jankowski** - *Feature selection with Decision Tree Criterion*. 2012.

[KRR11] **E. Kesavalu, V. N. Reddy, P. G. Rajulu** - *A Study of Intrusion Detection in Data Mining*. Proceedings of the World Congress on Engineering 2011 Vol IIIWCE 2011, July 6-8, 2011, London, UK.

[Lio09] **R. Lior** - *Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography*. Computational Statistics & Data Analysis 53: 12. 4046-4072, 2009.

[Mou97] **A. Mounji** - *Languages and Tools for Rule-Based Distributed Intrusion Detection*. PhD thesis, Faculties Universitaires Notre-Dame dela Paix Namur (Belgium), 1997.

[MCA08] **Y. Ma, D. Choi, S. Ata** - *Application of Data Mining to Network Intrusion Detection: Classifier Selection Model*, APNOMS 2008 LNCS 5297, pp. 399–408, 2008.

[MSA05] **S. Mukkamala, A. H. Sung, A. Abraham** - *Intrusion Detection Using Ensemble of Soft computing Paradigms.* Journal of Networks and Computer applications, 28: 167-18, 2005. doi:10.1016/j.jnca.2004.01.003.

[M+04] **S. Mukkamala,, A. H. Sung, A. Abraham, V. Ramos** - *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].

[M+16] **M. A. Mabayoje, A. O. Balogun, A. O. Ameen, V. E. Adeyemo** - *Influence of Feature Selection On Multi-Layer Perceptron Classifier for Intrusion Detection System*. Computing, Information System Development Informatics & Allied Research Journals. Vol 7 No 4. Pp 87-94.2016.

[NWY02] **S. Noel, D. Wijesekera, C. Youman** - *Modern Intrusion Detection, Data Mining, and Degrees of Attack Guilt*, In D. Barbarà and S. Jajodia (eds.), Applications of Data Mining in Computer Security, Kluwer Academic Publishers, Boston, MA, 2002, pp. 2-25.

[PP07] **A. Patcha, J. Park** - *Computer Networks*, The International Journal of Computer and Telecommunications Networking Volume 51 Issue 12, August, 2007 Pages 3448-3470.

[PSV13] **H. Patel, B. Sarkhedi, H. Vaghamshi** - *Intrusion Detection in Data Mining with Classification Algorithm*, IJAREEIE, Vol. 2, Issue7, July 2013.

[Sum97] **R. C. Summers** - *Secure computing: threats and safeguards.* NewYork: McGraw-Hill; 1997.

[Sun96] **A. Sundaram** - *An introduction to intrusion detection*. ACM CrossRoads 1996; 2(4).1996.

[SA14] **C. Shalinee, J. Amrag** - *Ensemble Neural Network and K-NN Classifiers for Intrusion Detection*. IJCSIT, Vol. 5(2), 2014.

[SAJ04] **C. Srilatha, A. Ajith, P. T. Johnson** - *Feature Deduction and Ensemble Design of Intrusion Detection Systems*. Computer & Security, 2004.

[SRS09] **T. Subbulakshmi, A. Ramamoorthi, S. M. Shalinie** - *Ensemble Design for Intrusion Detection Systems*. IJCSIT, Vol 1, No 1, August 2009.

[YBZ08] **G. Yu, Z. Bo, L. Zhao** - *PCA-ICA Ensemble Intrusion Detection System by Pareto-Optimal Optimization*. ITJ 7(3): 510-515, 2008.

Tibiscus