

## Performance Assessment of Different Classification Techniques for Intrusion Detection

G. Kalyani<sup>1</sup>, A. Jaya Lakshmi<sup>2</sup>

<sup>1</sup>Associate Professor, Dept of CSE, DVR & Dr HS MIC College of Technology, Kanchikacherla, Krishna(dt),.

<sup>2</sup> Professor, Dept of CSE DVR & Dr HS MIC College of Technology, Kanchikacherla, Krishna (dt),.

---

**Abstract:** Intrusion detection is one of the major research problems in network security. It is the process of monitoring and analyzing the events occurring in a computer system in order to detect different security violations. The aim of this paper is to classify activities of a system into two major categories: normal and abnormal activities. In this paper we present the comparison of different classification techniques to detect and classify intrusions into normal and abnormal behaviours using WEKA tool. WEKA is open source software which consists of a collection of machine learning algorithms for Data mining tasks. The algorithms or methods tested are Naive Bayes, j48, OneR, PART and RBF Network Algorithm. The experiments and assessments of the proposed method were performed with NSL-KDD intrusion detection dataset. With a total data of 2747 rows and 42 columns will be used to test and compare performance and accuracy among the classification methods that are used.

**Key Words:** Intrusion detection, Classification, Machine Learning, WEKA, j48, PART.

---

### I. Introduction

Due to remarkable escalation in networked computer resources, a variety of network-based applications have been developed to provide services in many different areas, e.g., ecommerce services, entertainment etc. The increase in the number of networked machines has show the way to an increase in unauthorized activity, not only from outside attackers, but also from inside attackers, such as discontented employees and people abusing their privileges for personal gain [2]. An intrusion is defined as any set of events that compromise the integrity, confidentiality or availability of a resource [3, 4]. If a system is capable to assure that these three security tokens are fulfilled, it is considered as secure. Intrusion detection (ID) is an approach of security managing scheme for computers and networks. ID is the process of monitoring and analyzing the actions occurring in a computer system in order to detect signs of security problems [5]. There are two main strategies of ID [6]: misuse detection and anomaly detection. Misuse detection attempts to match patterns and signatures of already recognized attacks in the network traffic. A continuously updated database is usually used to accumulate the signatures of known attacks. It cannot identify a novel attack until trained for them. Anomaly detection attempts to recognize behaviour that does not conform to normal behaviour. This technique is based on the detection of traffic anomalies. The anomaly detection systems are adaptive in nature, they can deal with new attack but they cannot identify the specific type of attack.

Classification is one of the most commonly applied supervised data mining technique. Classification constructs a classification model based on data. The model can be used to classify a new record of data into one of the predefined classes based on the values of attributes. Depending on the information available on classes and the type of classification, solution approaches are distinguished such as decision trees, rule induction, neural networks, K-nearest neighbours, Bayesian methods, evolutionary algorithms etc.

The subsequent sections of this paper are organized as follows; Section 2 describes the data preparation and methods used to detect the anomalies, Section 3 deals with simulation results of different algorithms, Section 4 contains discussions on the results and finally section 5 draws conclusion.

### II. Methods

#### 2.1 Data Preparation

WEKA is a data mining system developed by the University of Waikato in New Zealand that implements data mining algorithms using the JAVA language. WEKA is a state of-the-art facility for developing machine learning (ML) techniques and their application to real-world data mining problems. It is a collection of machine learning algorithms for data mining tasks. The algorithms are applied directly to a dataset. WEKA implements algorithms for data pre processing, classification, regression, clustering and association rules; It also includes visualization tools. The new machine learning schemes can also be developed with this package. WEKA is open source software issued under General Public License [7].

The data used in this evaluation is the Intrusion Detection data [1]. It has a total of 42 dimensions and 2747 instances. For the purposes of training and testing, only 75% of the overall data is used for training and the rest is used for testing the accuracy of the classification of the selected classification methods.

## 2.2 Learning Algorithms

### 2.2.1 J48:

J48 is an open source Java implementation of the C4.5 algorithm in the WEKA data mining tool. C4.5 is a program that creates a decision tree based on a set of labelled input data. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier.

C4.5 builds decision trees from a set of training data using the concept of information entropy. The training data is a set  $S = s_1, s_2, \dots$  of already classified samples. Each sample  $s_i = x_1, x_2, \dots$  is a vector where  $x_1, x_2, \dots$  represent attributes of the sample. The training data is augmented with a vector  $C = c_1, c_2, \dots$  where  $c_1, c_2, \dots$  represent the class to which each sample belongs.

#### Pseudo code

1. Check for base cases
2. For each attribute  $a$ 
  1. Find the normalized information gain from splitting on  $a$
  3. Let  $a\_best$  be the attribute with the highest normalized information gain
  4. Create a decision *node* that splits on  $a\_best$
  5. Recurse on the sublists obtained by splitting on  $a\_best$ , and add those nodes as children of *node*[8]

### 2.2.2 OneR(One Rule ):

OneR is a simple and a very effective classification algorithm frequently used in machine learning applications. OneR generates a one-level decision tree. OneR is able to infer typically simple, yet accurate, classification rules from a set of instances. OneR is also able to handle missing values and numeric attributes showing adaptability despite simplicity. The OneR algorithm creates one rule for each attribute in the training data, and then selects the rule with the smallest error rate as its 'one rule'. To create a rule for an attribute, the most frequent class for each attribute value must be determined. The most frequent class is simply the class that appears most often for that attribute value.

#### Pseudo-code

1. For each attribute A,
  - i. For each value VA of the attribute, make a rule as follows: Count how often each class appears  
Find the most frequent class Cf  
Create a rule when A=VA  
Class attribute value = Cf
  - ii. Calculate the error rate of all rules
2. Chose the rule with the smallest error rate [9].

### 2.2.3 PART (Projective Adaptive Resonance Theory):

PART is an Instance-based learner using an entropic distance measure.

#### Pseudo-code

Initialization

Number  $m$  of nodes in F1 layer:=number of dimensions in the input vector. Number  $m$  of nodes in F layer:=expected maximum number of clusters that can be formed at each clustering level. Initialize parameters  $L, \rho, \sigma, \rho, h, \sigma, \alpha, \theta$ , and  $e$ [10].

1. Set  $\rho = \rho_0$ .
2. Repeat steps 3 – 7 until the stopping condition is satisfied.
3. Set all F2 nodes as being noncommitted.
4. For each input vector in dataset  $S$ , do steps 4.1-4.6.
  - 4.1. Compute  $h_{ij}$  for all F1 nodes  $v_i$  and committed F2 nodes  $v_j$ . If all F2 nodes are noncommitted, go to step 4.3.
  - 4.2. Compute  $T_j$  for all committed F2 nodes  $V_j$ .
  - 4.3. Select the winning F2 node  $V_J$ . If no F2 node can be selected, put the input data into outlier 0 & then continue to do step 4.
  - 4.4. If the winner is a committed node, compute  $r_J$ , otherwise go to step 4.6.
  - 4.5. If  $r_J \geq \rho$ , go to step 4.6, otherwise reset the winner  $V_J$  and go back to step 4.3.
  - 4.6. Set the winner  $V_J$  as the committed and update the bottom up and top-down weights for winner node  $V_J$ .

5. Repeat step 4 N times until stable clusters are formed (i.e. until the difference of output clusters as  $N^{\text{th}}$  and  $(N-1)^{\text{th}}$  time becomes sufficiently small)
6. For each cluster  $C_j$  in F2 layer, compute the associated dimension set  $D_j$ . Then, set  $S = C_j$  and set  $\rho = \rho + \rho_h$  (or  $\rho = |D_j| \rho_h$ ), go back to step 2.
7. For the outlier  $O$ , set  $S = 0$ , go back to step 2.

### 2.2.4 Naive Bayes:

A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple. Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In spite of their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations [11].

### 2.2.5 RBF Network

A Radial Basis Function (RBF) network is an artificial neural network that uses radial basis functions as activation functions. It is a linear combination of radial basis functions. They are used in function approximation, time series prediction, and control. Radial basis function (RBF) networks typically have three layers: an input layer, a hidden layer with a non-linear RBF activation function and a linear output layer. The output,  $\varphi: \mathbb{R}^n \rightarrow \mathbb{R}$ , of the network is thus

$$\varphi(\mathbf{x}) = \sum_{i=1}^N a_i \rho(\|\mathbf{x} - \mathbf{c}_i\|)$$

where  $N$  is the number of neurons in the hidden layer,  $\mathbf{c}_i$  is the center vector for neuron  $i$ , and  $a_i$  are the weights of the linear output neuron. In the basic form all inputs are connected to each hidden neuron. The norm is typically taken to be the Euclidean distance and the basis function is taken to be Gaussian [12].

RBF networks are universal approximators on a compact subset of  $\mathbb{R}^n$ . This means that a RBF network with enough hidden neurons can approximate any continuous function with arbitrary precision. The weights  $a_i$ ,  $\mathbf{c}_i$ , and  $\beta$  are determined in a manner that optimizes the fit between  $\varphi$  and the data.

## III. Simulation Results

To measure and examine the performance on the selected classification methods or algorithms namely Naive Bayes, Radial Basis Function (RBF) network, J48, PART and OneR we use the same experiment procedure as suggested by WEKA. The 75% data is used for training and the remaining is for testing purposes. In WEKA, all data is considered as instances and features in the data are known as attributes. The simulation results are partitioned into many sub parts for easier analysis and assessment. On the first part, correctly and incorrectly classified instances will be partitioned in percentage value and then mean absolute error, root mean squared error, True Positive (TP) rate and False Positive (FP) Rate will be in numeric value. We also show the relative absolute error and root relative squared error in percentage for references and assessment. The results of the simulation are shown in Tables 1 and 2 below. Table 1 mainly summarizes the result based on accuracy and time taken for each simulation and Table 2 shows the result based on error and TP, FP rates during the simulation.

Table 1: Statistics of algorithm wise simulation

Algorithm	Correctly classified instances %	Incorrectly classified instances %	Time taken (seconds)
J48	99.13	0.87	0.22
RBF Network	92.29	7.71	0.63
PART	98.98	1.02	0.15
OneR	96.51	3.49	0.07
Naive Bayes	89.96	10.04	0.08

Table 2: Simulation Errors

Algorithm	True Positive Rate	False Positive Rate	Mean Absolute Error	Root Mean Squared Error	Relative Absolute Error (%)	Root Relative Squared Error (%)
J48	0.991	0.009	0.0123	0.0923	2.4558	18.3765
RBF Network	0.923	0.07	0.125	0.2509	24.9807	49.9620
PART	0.99	0.01	0.0139	0.0954	2.7785	18.9970
OneR	0.965	0.36	0.0349	0.1869	6.9722	37.2153
Naive Bayes	0.90	0.10	0.1002	0.3085	20.0018	61.4276

Figures 1 and 2 are the graphical representations of the simulation result.

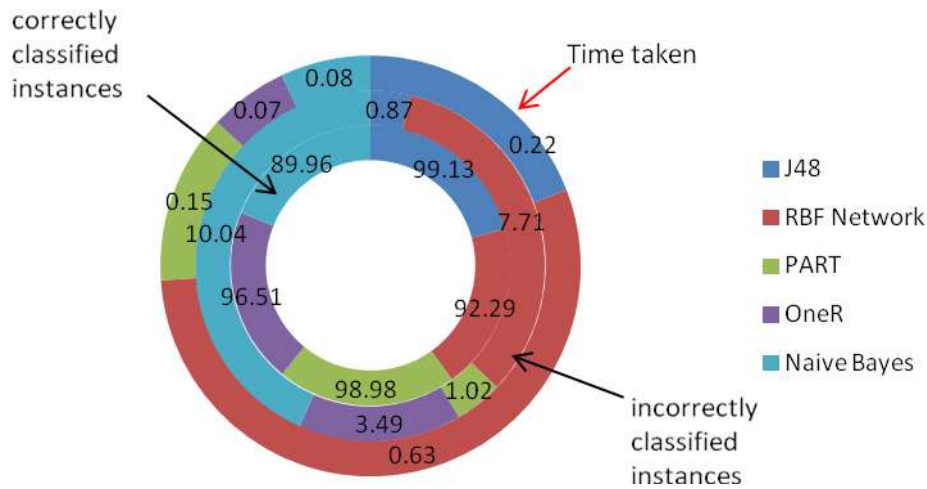


Figure 1: Graphical representation of Table 1

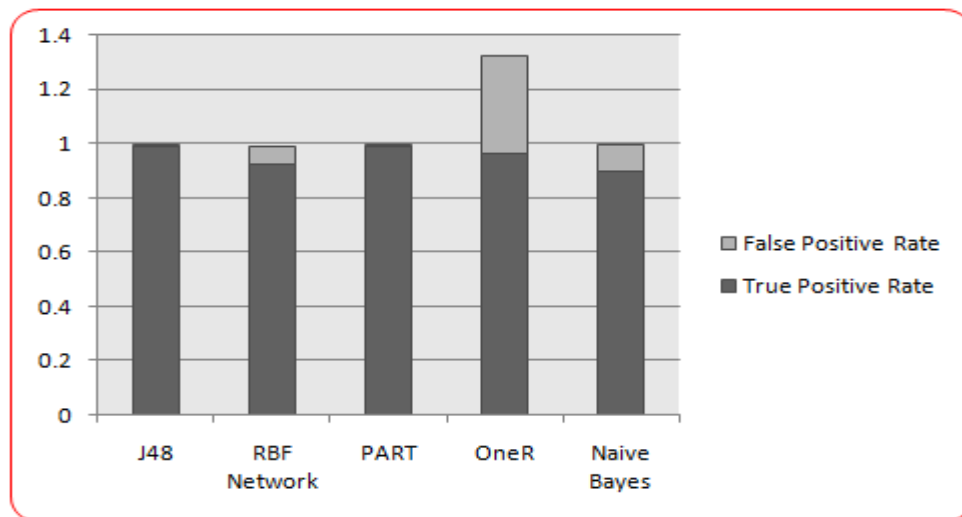


Figure 2: Graphical Representation of Table 2

#### IV. Discussions

Based on the above Figures 1, 2 and Table 1, it is clear that the highest accuracy is 99.13% and the lowest is 89.96%. The other algorithms yield an average accuracy of around 96%. Actually, the highest accuracy belongs to the J48 classifier, followed by PART with a percentage of 98.98% and subsequently OneR, RBF network and Naive Bayes classifier. Out of total 687 instances on an average 653 instances are correctly classified. . The total time required to build the model is also an important parameter in comparing the

classification algorithm. In this regard, from Figure 1, we can say that OneR requires the shortest time which is around 0.07 seconds compared to the others. RBF network requires the highest model building time which is around 0.63 seconds. The second on the list is Naive Bayes with 0.08 seconds.

From Figure 2, we can examine the differences of TP and FP rates resulting from simulation. In general, Positive indicates identified so True positive implies correctly identified and false positive implies incorrectly identified. The Highest TP rate belongs to J48 and PART classifiers and the lowest belongs to naïve Bayes classifier. Our analysis also implies a very commonly used indicator which is mean of absolute errors and root mean squared errors. Alternatively, the relative errors are also used. Since, we have two readings on the errors, average value will be considered. It is observed that the highest error is found in Naïve Bayes classifier with an average score of around 0.2 where the rest of the algorithms ranging averagely around 0.05-0.18. An algorithm which has a lower error rate will be preferred as it has more powerful classification capability and ability in terms of anomaly detection.

## V. Conclusion

Concluding our analysis, we have met our objective which is to evaluate and examine five selected classification algorithms based on WEKA. The best algorithm based on the intrusion detection data is PART classifier with an accuracy of 98.98% and the total time taken to build the model is at 0.15 seconds. PART classifier also has the lowest average error at 0.05 compared to others. These results suggest that among the machine learning algorithms tested, PART classifier is capable to significantly improve the conventional classification methods for use in anomaly detection in networks field.

## References:

- [1] [http://nsl.cs.unb.ca/NSL-KDD/KDDTrain+\\_20Percent.arff](http://nsl.cs.unb.ca/NSL-KDD/KDDTrain+_20Percent.arff)
- [2] Richard Power. 1999 CSI/FBI computer crime and security survey. Computer Security Journal Volume XV (2), 1999
- [3] Jian Pei Shambhu J. Upadhyaya Faisal Farooq Venugopal Govindaraju. Proceedings of the 20<sup>th</sup> International Conference on Data Engineering (ICDE'04) 1063-6382/04 \$ 20.00 © 2004 IEEE
- [4] Debar, H., Dacier, M., and Wespi, A., A Revised taxonomy for intrusion detection systems, Annales des Telecommunications, Vol. 55, No. 7-8, 361-378, 2000.
- [5] Bace, R. (2000). Intrusion Detection. Macmillan Technical Publishing
- [6] Markou, M. and Singh, S., Novelty Detection: A review, Part 1: Statistical Approaches, Signal Processing, 8(12), 2003, pp. 2481-2497.
- [7] WEKA at <http://www.cs.waikato.ac.nz/~ml/weka>
- [8] [Http://en.wikipedia.org/wiki/C4.5\\_algorithm](http://en.wikipedia.org/wiki/C4.5_algorithm)
- [9] [Http://www.saedsayad.com/oner.html](http://www.saedsayad.com/oner.html)
- [10] S.Vijayarani, M.Divya “ An efficient algorithm for Generating classification rules” International Journal of Computer Science & Technology issue4 october- December 2011.
- [11] [Http://en.wikipedia.org/wiki/Naive\\_Bayes\\_Classifier](http://en.wikipedia.org/wiki/Naive_Bayes_Classifier)
- [12] [Http://en.wikipedia.org/wiki/Radial\\_Basis\\_function\\_network](http://en.wikipedia.org/wiki/Radial_Basis_function_network)