# PERFORMANCE EVALUATION OF DECISION TREE CLASSIFIERS FOR RANKED FEATURES OF INTRUSION DETECTION

[1] *JAYSHRI R. PATEL*

**[1]Department of Computer Science,**
**Veer Narmad South Gujarat University,**
**Udhna Magdalla Road, Vesu, Surat, Gujarat, India.**
***jayshri.r@gmail.com***

***ABSTRACT:*** *Decision Trees are considered to be one of the most popular approaches for representing classifier for various disciplines such as statistics, machine learning and data mining. Classification of Intrusion detection, according to their features into either intrusive or non intrusive class is a widely studied problem. Decision trees are useful to detect intrusion from connection records. In this paper, we evaluate the performance of various decision tree classifiers for classifying intrusion detection data. The aim of this paper is to investigate the performance of various decision tree classifiers for ranked intrusion detection data. Information Gain is used to provide ranking to intrusion detection data. Decision tree classifiers evaluated are C4.5, CART, Random Forest and REP Tree.*

***KEYWORDS: Intrusion detection, Information Gain, Decision Tree, C4.5, CART (Classification and Regression Trees), Random Forest, REP (Reduced Error Pruning) Tree.***

JIKRIT

## 1. INTRODUCTION

Intrusion Detection is defined as a set of activities that attempt to distinguish the intrusive and normal activities. Intrusion Detection System (IDS) provides better security than the static defense mechanisms such as firewalls, software updates etc.

Intrusion detection is classified as host based or network based. A host based Intrusion Detection System will monitor resources such as system logs, file systems and disk resources; whereas a network based IDS monitors the data passing through the network.

The network intrusion detection has raw network traffic which should be summarized into higher-level objects such as connection records or audit record. The audit record capture various features of the network connections like duration, protocol type, source and destination bytes of a TCP connection. Feature selection reduces memory requirement and increases the speed of execution thereby increases the overall performance. Existing research shows that feature selection for intrusion detection increase effectiveness of the intrusion detection classification. Effective feature selection for intrusion detection identifies some of the important features for detecting anomalous network connections. In this work, the Information Gain feature selection method is used to provide ranking to the features of intrusion detection. First 15 ranked features and one class label i.e. 16 features among 42 features are selected for classification. Then, various Decision Tree classification algorithms namely C4.5, CART, Random Forest and REP Tree are applied to the ranked intrusion detection data.

## 2. RELATED WORK

In [1], Gary Stein et al. use Decision Tree classifier for Intrusion detection with GA based feature selection to improve the classification abilities of the decision tree classifier. They use a genetic algorithm to select a subset of input features for decision tree classifiers to increase the detection rate and decrease the false alarm rate in network intrusion detection. In [2], Juan Wang use C4.5 decision tree classification method to build an effective decision tree for intrusion detection. They build knowledge base of intrusion detection system from the rules which were generated from the decision tree. In [3], Mukkhmala et al. uses decision tree and Support Vector Machine (SVM) to model IDS. They compare the performance of SVM and Decision tree and found that Decision tree gives better overall performance than the SVM. In [4], Classification of intrusion detection is done based on various machine learning algorithms like J48, Naïve bayes, OneR and BayesNet. They find the Decision tree algorithm J48 most suitable with high positive rate and low false positive rate.

## 3. RANKING INTRUSION DETECTION DATA USING INFORMATION GAIN FEATURE SELECTION

Information gain measure is based on pioneering work by Claude Shannon on information theory, which studied the value or information content of messages. Let node N represents or hold the tuples of partition D. The attribute with the highest information gain is chosen as the splitting attribute for node N. This attribute minimizes the information needed to classify tuples in the resulting partitions and reflects the least randomness or impurity in these partitions. This approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple tree is found. [5]

Let pi be the probability that an arbitrary tuple in D belongs to class Ci, estimated by |Ci, D|/|D|. As discussed in the Han and Kamber [5], the expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i) \qquad (1)$$

Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} (|D_j| / |D|) \times Info(D_j) \qquad (2)$$

Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D) \qquad (3)$$

## 4. DECISION TREE CLASSIFIERS FOR INTRUSION DETECTION

Decision Tree is the learning of decision trees from class-labeled tuple. A decision tree is a flowchart-like tree structure, where each non-leaf node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node holds a class label. The top node is the root node. A decision tree can easily be converted to a set of classification rules. Many decision tree construction algorithms involve a two-step process [5] - Tree Construction and Tree pruning. During Tree construction a decision tree is constructed. The tree is partitioned till all the data items belong to the same class. In the Tree pruning step the constructed tree is prune back to prevent over fitting and to improve the accuracy.

Intrusion detection can be considered as classification problem where each connection record is identified as normal or intrusive based on some existing data. Classification for intrusion detection is an important challenge because it is very difficult to detect several new attacks, as the attackers are continuously changing their attack patterns. Various classification algorithms can be used for the classification of intrusion data such as Decision Tree, Naïve Bayes, OneR, Partial Decision Tree Algorithm and K-Nearest Neighbor algorithm. These algorithms provide the efficient results for intrusion detection data. As in [6], Decision tree algorithm and Partial Decision tree algorithm gives very efficient results for classification, the performance of various Decision Tree algorithms for intrusion detection can be evaluated. In the next section, I have discusses various Decision Tree Classifiers.

### C4.5

C4.5 algorithm is a successor of ID3 (Iterative Dichotomiser). C4.5 adopts a greedy approach in which decision trees are constructed in a top-down recursive divide-and –conquer manner. C4.5 handles continuous and discrete attributes. For handling continuous attributes, threshold is created and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it. The C4.5 works as follows: [7]

- Whenever a set of items (training set) is encountered, the algorithm identifies the attribute that discriminates the various instances most clearly. This is done using the standard equation of information gain;
- Among the possible values of this feature, if there is any value for which there is no ambiguity, i.e., for which the data instances falling within its category have the same value for the target variable, then that branch is terminated and the obtained target value is assigned to it;
- For all other cases, another set of attributes are looked at that gives the highest information gain;
- This is continued in the same manner until either a clear decision of the value of the target variable is reached with a combination of conditions on various independent variables/attributes, or running out of attributes;
- In the event of running out of attributes, or getting an ambiguous result from the available information, the branch is assigned a target value that the majority of the items under this branch possess.

JIKRIT

### CART

CART builds both classifications and regressions trees. The classification tree construction by CART is based on binary splitting of the attributes. It is also based on Hunt's model of decision tree construction and can be implemented serially. The CART decision tree is a binary recursive partitioning procedure capable of processing continuous and nominal attributes. CART uses regression analysis with the help of regression trees. The regression analysis feature is used in forecasting a dependent variable given a set of predictor variables over a given period of time.

### Random Forest

The Random Forest algorithm was developed by Leo Breiman. Random Forest, a meta-learner comprised of many individual trees, was designed to operate quickly over large datasets and more importantly to be diverse by using random samples to build each tree in the forest.

Construction of a tree:

- Randomly sample with replacement (bootstrap) the training set and select 2/3 of data to be used for tree construction; (inBag)
- Choose a random number of attributes from the in Bag data and select the one with the most information gain to comprise each node;
- Continue to work down the tree until no more nodes can be created due to information loss.
- Compute out-of-bag error estimates by running dataset through tree and measuring its correctness.

Diversity is obtained by randomly choosing attributes at each node of the tree and then using the attribute that provides the highest level of learning. The importance of this can not be overstated as the performance of the random forests algorithm is linked to the level of correlation between any two trees in the forest. The more the correlation increases, the lower the overall performance of the entire forest of trees. The way to vary the level of correlation between trees is by adjusting the number of random attributes to be selected when creating a split in each tree. Increasing this variable (m) will both increase the correlation of each tree and the strength of each tree. At some point the tree correlation and tree strength will complement each other providing the highest performance. In addition, increasing the number of trees will provide a more intelligent learner just as having a large diverse group will make intelligent decisions [7] [8].

**REP Tree**

REP Tree is a Fast decision tree learner. It builds a decision/regression tree using information gain/variance reduction and prunes it using reduced-error pruning. It only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces which is same as C4.5. [7]

IJKRIT

**5. EXPERIMENTS AND RESULTS**

The data for the experiments were prepared by the KDDCUP 1999 DARPA intrusion detection evaluation program by MIT Lincoln Laboratory. As given in [9], the data set contains 4 main attack categories namely Denial of Service (DoS), Remote to User (R2L), User to Root (U2R) and Probing. It includes total 24 different attacks types among the 4 main categories with 326026 instances. The original data set has 41 attributes for each connection record plus one class label. Examples of feature are protocol type, duration of each connection etc.

For performing the experiments I have used the open source package WEKA taken from [10]. Intrusion Detection Data with 24 different attack types are provided to the Information Gain Feature Selection Method. Using Information Gain Evaluation with Ranking search method for intrusion detection data, top 15 features among 41 features are selected for classification. This will generate Ranked list of features. I have select top 15 ranked features from the generated pre-processed dataset as shown in below fig. (1).

| NO. | Weightage | Name of the feature |
|---|---|---|
| 1 | 1.4384541 | SRC_BYTES |
| 2 | 1.3902543 | COUNT |
| 3 | 1.3552829 | SERVICE |
| 4 | 1.0977415 | SRV_COUNT |
| 5 | 1.0962104 | DST_HOST_SAME_SRC_PORT_RATE |
| 6 | 1.0159448 | PROTOCOL_TYPE |
| 7 | 0.9054554 | DST_HOST_SRV_COUNT |
| 8 | 0.8998206 | DST_HOST_DIFF_SRV_RATE |
| 9 | 0.8714134 | DST_HOST_SAME_SRV_RATE |
| 10 | 0.7844989 | DIFF_SRV_RATE |
| 11 | 0.7774032 | SAME_SRV_RATE |
| 12 | 0.7646002 | FLAG |
| 13 | 0.5820464 | DST_BYTES |
| 14 | 0.5663059 | DST_HOST_SERROR_RATE |
| 15 | 0.5449567 | SERROR_RATE |

Fig: 1 Top 15 ranked features using Information gain Evaluator and Ranker Search Method for Intrusion Detection Data

Now, various decision tree classifiers C4.5, CART, Random Forest and REP Tree are applied to the above 15 ranked features intrusion detection data. Using various decision tree classifiers, classification model is build.

The performance of the algorithms is evaluated based on the measures Accuracy, Learning Time (in seconds) and Size of the Tree.

As shown in the fig. (2), the learning time taken by the REP Tree is much less compare to other three algorithms. The size of the tree is smaller in the CART, but CART takes highest time to build the classifier.

As shown in the fig. (3) and (4), Random Forest identifies the highest number of correctly classified instances and less number of incorrectly classified instances. Random forest of 10 trees, each constructed while considering 5 random features. Random Forest gives Out of bag error is 0.046.

| | C4.5 | CART | Random Forest | REP Tree |
|---|---|---|---|---|
| Learning Time (in Seconds) | 58.31 | 711.99 | 89.64 | 14.55 |
| Size of the tree | 571 | 189 | - | 267 |

Fig. 2 Results of various Decision Tree Classifiers

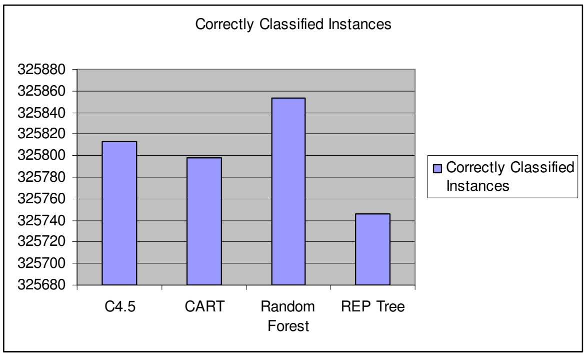

Fig. 3 Correctly Classified instances - Comparison of various Decision Tree Classifiers

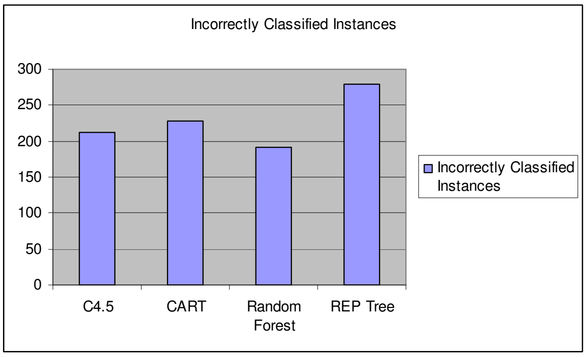

Fig. 4 Incorrectly classified instances - Comparison of various Decision Tree Classifiers

## 6. CONCLUSION

In this paper, performance of four selected decision tree classification algorithms for ranked intrusion detection data is evaluated and investigated. From the above experiment & result analysis it is very clear that the performance of Random Forest is better as it correctly identifies more number of instances than other. The learning time of REP Tree is very less compare to others but accuracy of REP Tree is very less. These results suggest that among the various decision tree classification algorithms tested, Random Forest and C4.5 decision tree classifier has the potential to significantly improve classification results for ranked intrusion detection.

## 7. REFERENCES

[1] Gary Stein, Bing Chen, Annie S. Wu, Kien A. Hua "Decision tree classifier for network intrusion detection with GA based feature selection" *ACM-SE 43: Proceedings of the 43rd annual Southeast regional conference*- Vol. 2, 136-141, March [2005].

[2] Juan Wang; Qiren Yang; Dasen Ren, "An Intrusion Detection Algorithm Based on Decision Tree Technology," Information Processing, APCIP 2009. *Asia-Pacific Conference,* vol.2, no., pp.333, 335, [2009]

[3] Mukkamala S., Sung A.H. and Abraham A., "Intrusion Detection Using Ensemble of Soft Computing Paradigms", *Third International Conference on Intelligent Systems Design and Applications, Springer Verlag Germany*, 239-248, [2003].

[4] Yogendra Kumar Jain and Upendra, "An efficient Intrusion Detection Based on Decision Tree Classifier Using Feature Reduction" *International Journal of Scientific and Research Publications*, Vol. 2, Issue 1, Jan. 2012, ISSN 2250-3153 [2012]

[5] Jiawei Han,Micheline Kamber, "*Data Mining: Concepts and Techniques*", 2nd Edition, Morgan Kaufmann [2006]

[6] J.R. Patel, "Classification of Relevant and Redundant Intrusion Detection Data Using Machine Learning Approaches*", Journal of Information, Knowledge and Research in Computer Science and Applications,* ISSN:0975-6728, Vol.2, Issue-2, Nov. 2012- Oct. 2013 pp. 103-105 [2012-13]

[7] Suban Ravichandran, Vijay Bhanu Srinivasan and Chandrasekaran Ramasamy, "Comparative Study on Decision Tree Techniques for Mobile Call Detail Record", *Journal of Communication and Computer 9*, pp. 1331-1335, [2012]

[8] N. Peter, "Enhancing random forest implementation in WEKA", in: Machine Learning Conference, [2005]

[9] KDD cup 99, http://kdd.ics.uci.edu/database/kddcup99/kddcup.data 10 percent.gz.

[10] http://www.cs.waikato.ac.nz/~ml/weka.

JIKRIT