Statistical Analysis of Network Security using Network Traffic Data Classification

*K.N. Pradhan[1], Dr. M. Siddppa[2], Dr.S.Kavitha[3]

[1]Research Scholar,Jain University,Department of Computer Science & Engineering,Bangalore

,[2] Professorand HOD, Department of Computer Science & Engineering , SSIT, TUMKUR
[3]Assistant Professor, Department of Computer Applications,Bangalore.
[1]pradhanaicte@gmail.com, [3]s.kavitha527@gmail.com

Abstract: Network security is essential for the modern era to protect the usability and integrity of the network and data. It manages the unauthorized access to the network. So it is significant to recognize and applied classification algorithm for escalating the performance in network security. In our research paper proposed the analysis of integrated network for the network traffic data. In this paper we evaluate statistical traffic characteristics with various algorithms for real-time network traffic classification. We found that ZeroR classifier is fast compared to other algorithms. In the tree classification Decision tree takes only o.o2 seconds to complete execution. It is proved that Fatherfirst is the best algorithm to network security in network signal classification. Our proposed method is used in integrated network security systems to classify the real network traffic datasets to select the best algorithm**.**The proposed method can be applied for real time traffic classification for the better network security issues.

Keywords: Network Security, Networks, Data Mining, Classification, Traffic.

1. INTRODUCTION

Network security is in demand as network attacks have amplified in number, IDS is playing important role. Due to large set of data is collected in network database, it is necessary to classify the statistical data of network traffic. The classification is to predict the category to which a particular traffic data belongs to. Data mining is essential in network security to find the various IP address, surveillance,cyber security,attacking buildings and destroying critical infrastructures such as power grids and telecommunication systems,to identify suspicious individuals and groups, capable of carrying out terrorist activities. In this paper we will focus on data mining for integrated network security.. Classification algorithms are used to cluster different traffic data. The dataset used is MAWI traffic data collected my MAWILab, which is updated daily to include new traffic from upcoming applications and anomalies. The weka tool is used to analyze and visualize packet traces.

2. LITERATURE SURVEY

In 2016 Christian Callegari[1] et al., has published a paper on performance comparison between two different histogram based anomaly detection methods using the Euclidean distance and the entropy to measure the deviation from the normal behaviour. In 2014,Chakchai et al used classification algorithms Decision Tree, Ripper Rule[2], Neural Networks, Naïve Bayes, k-Nearest-Neighbour, and Support Vector Machine using both KDD CUP dataset and recent HTTP BOTNET attacks for network intrusion systems. The author julisch[3] published a paper on representation of cross section of the various research effort on network related issues. The[4] proposed a method to design intrusion detection from suspicious URLs using optimal fuzzy logic system.

3. Data Set Information

The MAWI network traffic data is collected from Working Group Traffic Archieves has trans-Pacific packet traces from 1999-2003. The dataset collected with each day's statistic data of the protocol breakdown and the important biggest flows[Fig.2].

4. PREPROCESSING

Preprocessing is done using the Cartesian product algorithm shown in the diagram 1. Cartesian product is a filter for performing the Cartesian product of a set of nominal attributes for the MAWI dataset.

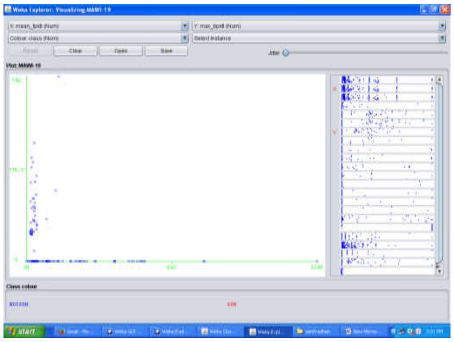


Figure 1: Preprocessing using Cartesian product

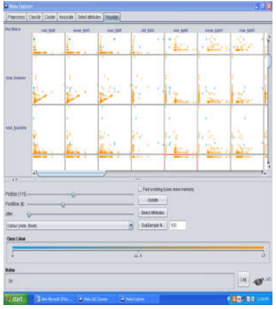


Figure 2: Attribute visualization

5.RESULTS AND DISCUSSIONS

5.1.CLASSIFICATION

Classification algorithms are applied for MAWI traffic dataset to find the best performance in accordance with the time complexity, error rate %, correctly classified instances and incorrectly classified instances. For all classifiers full training set is given with 20-fold cross validation.

Run information of all the classifiers are listed below.

5.1.1.ZEROR CLASSIFIER

The ZeroR algorithm is gives importance on the target and neglects all predictors. It predicts the best component class. It is useful for determining a baseline performance as a standard for other classification methods. Time taken to build model is 0 seconds for the traffic database.

5.1.2. BESTFIRST SEARCH ALGORITHM

BFS is convenient classifier algorithm. It uses greedy hill climbing augmented with a backtracking facility to finds the space of attribute subsets. It has three stages to classify the dataset:

- Begin with the empty set of attributes, proceed the searching in forward direction
- Begin with the complete set of attributes, proceed the searching in backward direction
- Else begin at any point, search in both directions.

Time taken to build model: 0.06 seconds

```
              TP Rate  FP Rate  Precision  Recall   F-Measure  MCC     ROC Area  PRC Area
Class
              1.000    0.000    1.000      1.000    1.000      0.000   ?         1.000     NOTSSH
              0.000    0.000    0.000      0.000    0.000      0.000   ?         ?         SSH
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      0.000   0.000     1.000
```

```
=== Confusion Matrix ===

  a   b   <-- classified as
 204   0 |   a = NOTSSH

   0     0 |   b = SSH
```

5.1.3. JRIP CLASSIFIER

The JRIP classifier is implemented by William W. Cohen as an optimized version of IREP. This class implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction known as RIPPER.Time taken to build model: 0.02 seconds.

```
=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
              1.000    0.000    1.000      1.000    1.000      0.000    ?         1.000     NOTSSH
              0.000    0.000    0.000      0.000    0.000      0.000    ?         ?         SSH
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      0.000    0.000     1.000

=== Confusion Matrix ===

   a   b   <-- classified as
 204   0 |   a = NOTSSH

   0     0 |   b = SSH
```

5.1.4. ONER CLASSIFIER

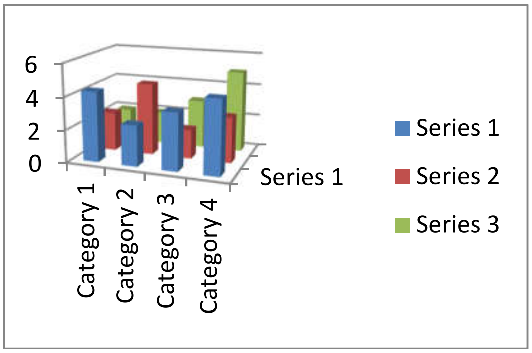
The One Rule classifier is one that generates one rule for each predictor in the data. After selecting the predictor,it selects the rule with the least error as its one rule[7]. The algorithm gives a rule for a predictor to build a frequency table for every interpreter against the target. Time taken to build model: 0.02 seconds

```
=== Detailed Accuracy By Class ===

              TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
              1.000    0.000    1.000      1.000    1.000      0.000    ?         1.000     NOTSSH
              0.000    0.000    0.000      0.000    0.000      0.000    ?         ?         SSH
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      0.000    0.000     1.000

=== Confusion Matrix ===

   a   b   <-- classified as
 204   0 |   a = NOTSSH
   0   0 |   b = SSH
```

Table shows the results of classifiers for the MAWI dataset and the details such as correctly classified, incorrectly classified and kappa statistic(Table 1).

Statistics	ZEROR CLASSIFIER	BESTFIRST SEARCH	JRIP CLASSIFIER	ONER CLASSIFIER
Correctly Classified Instances	100%	100%	100%	100%
Incorrectly Classified Instances	0%	0%	0%	0%
Kappa statistic	1	1	1	1
Mean absolute error	0.0051	0.0051	0%	0%
Root mean squared error	0.0051	0.0051	0%	0%
Relative absolute error	100%	100.5134 %	0%	0%
Root relative squared error	100%	100.5134 %	0%	0%
Total Number of Instances	204	204	204	204

Table 1: Evaluation of classifiers



Graph 1: Graph shows the result of classifiers

5.2. TREES

5.2.1. DECISION STUMP TREE

A decision stump is a machine learning model consisting of a one-level decision tree.That is, it is a decision tree with one internal node (the root) which is immediately connected to the terminal nodes (its leaves). A decision stump makes a prediction based on the value of just a single input feature. Sometimes they are also called 1-rules

Decision Stump
Time taken to build model: 0.02 seconds

5.2.2. M5P TREE

```
M5 pruned model tree:
(using smoothed linear models)

duration <= 82 : LM1 (138/0%)
duration >  82 :
|   total_fvolume <= 98.5 : LM2 (22/0%)
|   total_fvolume >  98.5 :
|   |   total_fvolume <= 360 :
|   |   |   max_fpktl <= 68.5 : LM3 (6/0%)
|   |   |   max_fpktl >  68.5 :
|   |   |   |   max_bpktl <= 207 :
|   |   |   |   |   duration <= 205223101 : LM4 (7/12.371%)
|   |   |   |   |   duration >  205223101 : LM5 (2/0%)
|   |   |   |   max_bpktl >  207 : LM6 (6/0%)
|   |   total_fvolume >  360 :
|   |   |   min_fpktl <= 120 : LM7 (14/36.46%)
|   |   |   min_fpktl >  120 : LM8 (9/14.287%)

LM num: 1
total_fpackets =
        0.0009 * min_fpktl
        - 0.0001 * mean_fpktl
        - 0.0015 * max_fpktl
        + 0.0023 * std_fpktl
        + 0.0003 * max_bpktl
        + 0 * duration
        + 0.0007 * total_fvolume
        + 0.9965

LM num: 2
total_fpackets =
        0.026 * min_fpktl
        - 0.0243 * mean_fpktl
        - 0.0104 * max_fpktl
        + 0.0287 * std_fpktl
        + 0.0013 * max_bpktl
        + 0 * duration
        + 0.004 * total_fvolume
        + 1.3158
```

LM num: 3
total_fpackets =
0.0304 * min_fpktl
- 0.033 * mean_fpktl
- 0.0103 * max_fpktl
+ 0.0305 * std_fpktl
+ 0.0014 * max_bpktl
+ 0 * duration
+ 0.005 * total_fvolume

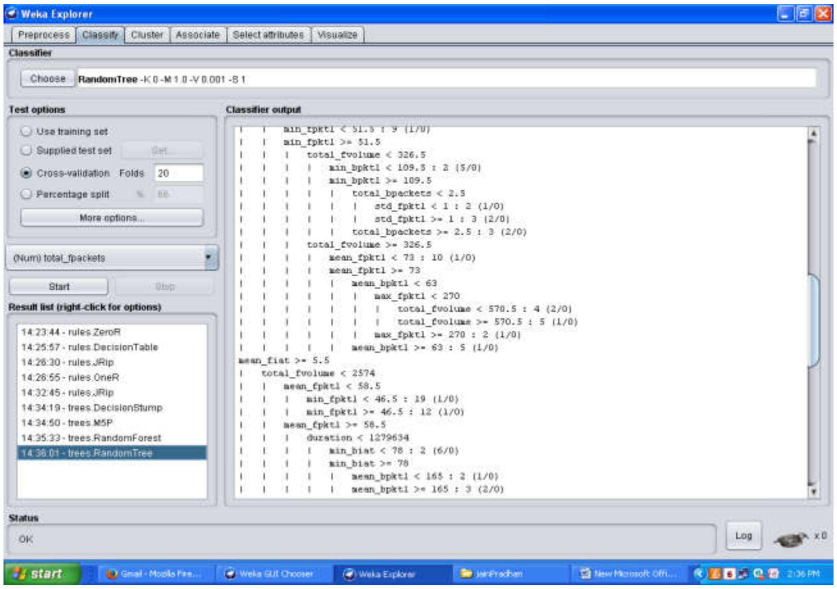
Number of Rules : 8

Time taken to build model: 0.16 seconds

5.2.3.RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.Random decision forests correct for decision trees' habit of overfitting to their training set.

Time taken to build model: 0.14 seconds



Classifier: RandomTree -K 0 -M 1.0 -V 0.001 -S 1

Test options:
○ Use training set
○ Supplied test set
● Cross-validation Folds 20
○ Percentage split % 66

(Num) total_fpackets

Result list (right-click for options)

```
14:23:44 - rules.ZeroR
14:25:57 - rules.DecisionTable
14:26:30 - rules.JRip
14:26:55 - rules.OneR
14:32:45 - rules.JRip
14:34:19 - trees.DecisionStump
14:34:50 - trees.M5P
14:35:33 - trees.RandomForest
14:36:01 - trees.RandomTree
```

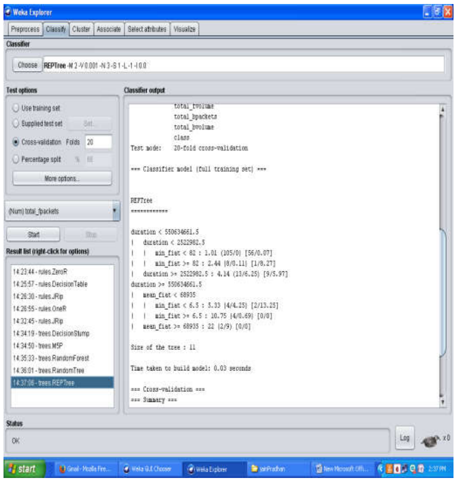
Classifier output

```
|   |   min_fpktl < 51.5 : 9 (1/0)
|   |   min_fpktl >= 51.5
|   |   |   total_fvolume < 326.5
|   |   |   |   min_bpktl < 109.5 : 2 (5/0)
|   |   |   |   min_bpktl >= 109.5
|   |   |   |   |   total_bpackets < 2.5
|   |   |   |   |   |   std_fpktl < 1 : 2 (1/0)
|   |   |   |   |   |   std_fpktl >= 1 : 3 (2/0)
|   |   |   |   |   total_bpackets >= 2.5 : 3 (2/0)
|   |   |   total_fvolume >= 326.5
|   |   |   |   mean_fpktl < 73 : 10 (1/0)
|   |   |   |   mean_fpktl >= 73
|   |   |   |   |   mean_bpktl < 63
|   |   |   |   |   |   max_fpktl < 270
|   |   |   |   |   |   |   total_fvolume < 570.5 : 4 (2/0)
|   |   |   |   |   |   |   total_fvolume >= 570.5 : 5 (1/0)
|   |   |   |   |   |   max_fpktl >= 270 : 2 (1/0)
|   |   |   |   |   mean_bpktl >= 63 : 5 (1/0)
mean_fiat >= 5.5
|   total_fvolume < 2574
|   |   mean_fpktl < 58.5
|   |   |   min_fpktl < 46.5 : 19 (1/0)
|   |   |   min_fpktl >= 46.5 : 12 (1/0)
|   |   mean_fpktl >= 58.5
|   |   |   duration < 1279634
|   |   |   |   min_biat < 78 : 2 (6/0)
|   |   |   |   min_biat >= 78
|   |   |   |   |   mean_bpktl < 165 : 2 (1/0)
|   |   |   |   |   mean_bpktl >= 165 : 3 (2/0)
```

Status
OK

Figure 3: Random Tree Algorithm

5.2.4.REPTREE ALGORITHM



REPTree -M 2 -V 0.001 -N 3 -S 1 -L -1 -I 0.0

○ Use training set
○ Supplied test set
● Cross-validation Folds 20
○ Percentage split % 66

(Num) total_fpackets

14:23:44 - rules.ZeroR
14:25:57 - rules.DecisionTable
14:26:30 - rules.JRip
14:26:55 - rules.OneR
14:32:45 - rules.JRip
14:34:19 - trees.DecisionStump
14:34:50 - trees.M5P
14:35:33 - trees.RandomForest
14:36:01 - trees.RandomTree
14:37:06 - trees.REPTree

```
              total_fvolume
              total_bpackets
              total_bvolume
              class
Test mode:    20-fold cross-validation

=== Classifier model (full training set) ===


REPTree
============

duration < 550634661.5
|   duration < 2522982.5
|   |   min_fiat < 82 : 1.01 (105/0) [56/0.07]
|   |   min_fiat >= 82 : 2.44 (8/0.11) [1/8.27]
|   duration >= 2522982.5 : 4.14 (13/6.25) [9/5.97]
duration >= 550634661.5
|   mean_fiat < 68935
|   |   min_fiat < 6.5 : 5.33 (4/4.25) [2/13.25]
|   |   min_fiat >= 6.5 : 10.75 (4/0.69) [0/0]
|   mean_fiat >= 68935 : 22 (2/9) [0/0]

Size of the tree : 11

Time taken to build model: 0.03 seconds

=== Cross-validation ===
=== Summary ===
```

OK

Figure 4. RepTree Algorithm

REPTree

============

```
duration < 550634661.5
|  duration < 2522982.5
|  |  min_fiat < 82 : 1.01 (105/0) [56/0.07]
|  |  min_fiat >= 82 : 2.44 (8/0.11) [1/8.27]
|  duration >= 2522982.5 : 4.14 (13/6.25) [9/5.97]
duration >= 550634661.5
|  mean_fiat < 68935
|  |  min_fiat < 6.5 : 5.33 (4/4.25) [2/13.25]
|  |  min_fiat >= 6.5 : 10.75 (4/0.69) [0/0]
|  mean_fiat >= 68935 : 22 (2/9) [0/0]

Size of the tree : 11
```

Time taken to build model: 0.03 seconds

Table shows the results of trees for the MAWI dataset and the details such as correlation correctly classified, incorrectly classified and kappa statistic(Table 2).

	DECISION STUMP	**M5P TREE**	**RANDOM FOREST**	**REPTREE**
Correlation coefficient	**0.5462**	**0.938**	**0.8176**	**0.6443**
Mean absolute error	**1.0768**	**0.3278**	**0.4371**	**0.8012**
Root mean squared error	**2.4144**	**0.9824**	**1.6795**	**2.2235**
Relative absolute error	**72.4101 %**	**22.0423 %**	**29.3955 %**	**53.8771 %**
Root relative squared error	**84.9462 %**	**34.5642 %**	**59.0909 %**	**78.2285 %**
Total Number of Instances	**204**	204	204	204

Table 2: Performance Evaluation of Tree algorithms

Table shows the evaluation for the MAWI dataset with details of time taken to build the model (Table 3).

Algorithm	**ZeroR**	**0**
	BredthFirstSearch	**0.06**
	JRIP	**0.02**
	OneR	**0.02**
Trees	**DecisionTree**	**0.02**
	M5P	**0.16**
	RandomForest	**0.14**
	RepTree	**0.03**
Cluster	**Canopy**	**0.03**
	Cobweb	**0.27**
	EM	**6.27**
	FathersFirst	**0**

Table 3: Comparison of data mining algorithms

CONCLUSION

Network security playing an important role for the current era. In the proposed method network traffic classification is done using classification algorithm and clustering algorithms. We found that ZeroR classifier is fast compared to other algorithms. In the tree classification Decision tree takes only o.o2 seconds to complete execution. It is proved that Fatherfirst is the best algorithm to network security in network signal classification. Our proposed method is used in integrated network security systems to classify the real network traffic datasets to select the best algorithm**.**

References

[1]. Statistical Network Anomaly Detection: An Experimental Study Christian Callegari(B), Stefano Giordano, and Michele Pagano, Springer International Publishing AG 2016, R. Doss et al. (Eds.): FNSS 2016, CCIS 670, pp. 12–25, 2016. DOI:10.1007/978-3-319-48021-3

[2]. An Evaluation of Data Mining Classification, Models for Network Intrusion Detection,:Kasidit Wijitsopon and Kanokmon Rujirakul, ISBN 978-1-4799-3724-0/14/$31.00 ©2014 IEEE

[3]. K. Julisch, "Data Mining for Intrusion Detection," Applications of Data Mining in Computer Security, Advances in Information Security, vol. 6, pp. 33–62, 2002.

[4]. I. Butun, S.D. Morgera, and R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," IEEE Communication Surveys & Tutorials, vol.16, no.1, pp. 266–282,2014.

[5]. Weka – Data Mining Machine Learning Software. Available at http://www.cs.waikato.ac.nz/ml/weka/]

[6]. http://www.fukuda-lab.org/mawilab/documentation.html

[7]. www.wikipedia.org/classificationrules/