Intrusion Detection of DOS Attacks using Feature Selection and Binning

---

IS734 - Data Analytics for Cybersecurity
Dr. Vandana Janeja
Spring 2016

By
Ashwin Chengappa
Geethika Mudunuri
Naveen Nandipi

Abstract:

Feature selection and binning can be done is several different ways. In our paper, after referring to two related works, we propose a new method to feature selection that involves a combination of both the methods used in two papers various levels of selecting features and maximizing on the efficiency of attack detection with the least amount of features. Through our method, we were able to attain a better detection rate for detecting DOS attacks in the 1999 KDD Cup dataset. More specifically, we were able to achieve high precision rate and lower error rates on the most occurring classes such as Normal, Neptune and Smurf that comprise 98% of the training data.

Table of Contents:

1. Introduction

In this day and age, the use of computers and networks has increased tremendously due to which cybercrime has also been on a rise. To defend against such cyber attacks and computer viruses, several security techniques have been researched in the past decade like firewalls, anomaly and intrusion detection (Nguyen & Choi). IT security industry is estimated to be worth $77 billion and is growing rapidly at over 8% annually (NASSCOM SETS-UP). According to the latest report by NASSCOM, **rising cyber security threats** are expected to create $35 billion revenue opportunity in the coming years. This is an issue that has been increasing exponentially in the past few years. These days it has become very difficult to move data around (be it confidential company information or personal identification information like account details or financial information) without giving a thought of when where and who might steal the information. Big organisations today are always worried about the security of their data in their day to day activities. Healthcare sector, which did not face major attack problems during early 2000, has become one of the biggest victims of cyber threats and attacks in recent years. For example, interruptions to a hospital's system in terms of denial of service can create a numerous life or death situations.

In our research, we will focus on the 1999 KDD Cup dataset which is a simplified and smaller version of the 1998 DARPA dataset. We will focus on the various types of DOS attacks; back, land, neptune, pod, smurf, and teardrop. The feature sets we derived from previous case studies will be our starting point to analyze detection rates. Then, we will come up with our own feature list for detecting DOS attacks that will be useful for future research.

1.1. Motivation

It is critical for big organizations to have uninterrupted access to information at all times to run the business and by the DOS attacks it has become impossible. This shows us how important the cyber security industry is and also how important it is to show continuous improvement. In the recent years there have been a lot of interesting research on Knowledge discovery. Our motivation comes from its importance and criticality of the matter. Our ultimate aim of this project is to show how certain Denial Of Service (DOS) attacks could be detected at a better rate than previously done. We try to achieve this by studying previous papers and coming up with our own sort of approach in a way where we could use the best possible feature set and compare them to show better detection rate..

To sustain uninterrupted business, service and save a lot of money. A recent survey on estimating the cost of a successful DDoS (source of attack is more than one) attack gave the following results in terms of cost per hour.

- 36% of respondents -> $5,000 ~ $19,999.
- 15% of respondents -> less than $5,000.

- 17% of respondents -> $20,000 ~ $59,999.
- 17% of respondents -> $60,000 ~ $99,999.
- 15% of respondents -> over $100,000.

Considering that approximately 49% of the attacks last between 6 hours to 24 hours, the average cost is estimated at roughly $500,000. However, security companies say some attacks can result in much higher costs. Our motivation comes from the above explanation where we try to reduce various efforts and increase the rate of detection.

By implementing a more crisp and exact feature list we expect the result to show different detection rates and compare each of the approaches used to come up with the one that produces a better detection rate.

Other things that will impact the our approach are:

1) **Reduced time**

   By implementing an exact feature set that will only impact the result and getting rid of any unwanted feature which might add to runtime and detection time, we would be able to produce a result in lesser time.

2) **Reduced cost**

   While this project aims at producing better detection rates between different approaches we would be able to save cost and effort on research on DOS attack detection.

3) **Improved detection rate**

   By making use of different feature sets and building one of our own set we aim to produce a better detection rate as compared to the other feature set used in the analysis.

4) **Ability to implement the same approach and bring same results on different platforms.**

   This feature set should not only produce this result in ;Weka but also be able to give similar results in other applications for instance Knime, Spark and so on.

1.2. Challenges

Among the various challenges we faced, these were the most prevalent:

- *Reducing the feature set by over 50% :* The primary aim of this project is to reduce the dimensionality of the data. Reducing the no. of features used by the two related papers to come up with a more effective and result driven approach is one of the challenges.
- *CFS Subset Evaluation for feature selection:* Applying CFS Subset Evaluation on the already reduced feature set and further apply the subset evaluation on each DOS attack type to assign a rank for each feature based on its relevance to each attack type and

finally summarising a new feature set containing 9 attributes is one of the biggest challenges.

- *Removing of certain features (relevant for less occuring attack class) as irrelevant to reduce the error rate:* Here certain relevant features that were only specific to small attack types like land and pod were ignored since their addition resulted in the increased error rate on much bigger attack types such as Smurf and Back.

2. Background of the Data

In 1998 DARPA Intrusion Detection Evaluation Program was managed by MIT Lincoln Labs. The objective of this program was to come up with a set of realistic attacks and embed them in normal data to evaluate the false alarm rate and detection rates, and then improve systems to correct the detected weaknesses (Devaraju & Ramakrishnan). An intrusion detection system is used to detect unauthorized intrusions into a computer system or network. These systems will generate alerts based on the following behaviors in a confusion matrix:

True positive (TP): The amount of attack detected when it is actually attack;

True negative (TN): The amount of normal detected when it is actually normal;

False positive (FP):The amount of attack detected when it is actually normal called as false alarm;

False negative (FN): The amount of normal detected when it is actually attack, namely the attacks which can be detected by intrusion detection system.

		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	**TP** True Positive	**FP** False Positive
	negatives	**FN** False Negative	**TN** True Negative

The existing intrusion detection systems in the military at that time produced a large number of false alarms while trying to find a certain percentage of intrusions.

2.1. Data Description

For our research, we will be focusing on the 1999 KDD Cup dataset which is a simplified and smaller version of the 1998 DARPA dataset. The KDD dataset has different types of attacks:

Type of Attack	Description	Example
Denial of Service	deny legitimate requests to a system	Flood
Remote-to-Local (R2L)	unauthorized access from a remote machine	Guessing passwords
User-to-Root (U2R)	unauthorized access to local superuser (root) privileges	Various buffer overflow attacks
Probing	surveillance and other probing	Port scanning

In our research, we will be focusing on all the DOS attacks: back, land, neptune, pod, smurf, and teardrop.

Back - An attacker submits requests URL's containing many front slashes. As the server tries to process these requests it will slow down and becomes unable to process other requests.

Land - Occurs when an attacker sends a spoofed SYN packet in which the source address is the same as the destination address.

Neptune - Also known as SYN flood, is whenever an attacker sends syn requests to a targeted system in an attempt to consume enough server resources to make the system become unresponsive to legitimate/actual traffic.

Pod - Ping of Death is an attack that affects many older operating systems. Some systems will react in an unpredictable way when receiving oversized IP packets such as crashing, freezing or rebooting.

Smurf - A system is flooded with spoofed ping messages which creates high computer network traffic on the victim's network making it unresponsive.

Teardrop - Involves sending fragmented packets to a target machine. Since the machine receiving such packets cannot reassemble them due to a bug in TCP/IP fragmentation reassembly, the packets overlap one another, crashing the target network device.

We focused on two feature sets that we derived from previous case studies and analyzed their detection rate. Based on those features, we created our own feature list for detecting DOS attacks.

The training dataset consists of 494,021 records as follows:

Class Type	No. of Records	% of Occurrence
Normal	97,277	19.69
DOS	391, 458	79.24
Probe	4,107	0.83
R2L	1.126	0.23
U2R	52	0.01
Total	494,020	100.00

3. Related Work

Class Label	Feature List 1	Feature List 2
Normal	1, 6, 12, 15, 16, 17, 18, 19, 31, 32, 37	1, 3, 5, 6, 10, 12, 16, 19, 23, 24, 26, 28, 30, 31, 32, 33, 34, 35, 36, 37, 38, 40, 41
DOS Attacks	2, 3, 4, 5, 23, 24, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 38, 39, 40, 41	3, 23, 29, 30, 32, 34, 35

Feature List 1:
Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features - Adetunmbi A.Olusola., Adeola S.Oladele. and Daramola O.Abosede

In this paper, Adetunmbi, Adeola and Daramola present the relevance of each feature in 1999 KDD Cup intrusion detection dataset to the detection of each class. They used the rough set degree of dependency and dependency ratio of each class to determine the most discriminating features for each class. Based on the work done by them, it is evident that every study involves removing the features that are not relevant as one of their first steps to narrow down the analysis. It also follows removing features that are less relevant to further improve the analysis. In their

work, features 20 and 21 (outbound_command_count and hot_login) were disregarded since they did not help even a bit in the intrusion detection approach. Further, they also delete features 13, 15, 17, 22, and 40 (no_of_compromised_conditions, su_attempted, no_of_file_creation_operations, guest_login, dest_host_rerror_rate respectively) as features that are less relevant.

Feature List 2:
Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets - H. Günes Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood

In this paper, Gunes uses information gain and entropy approach to measure the most significant features for any given class. The features used in the paper were grouped into four categories:

Basic Features: Basic features can be derived from packet headers without inspecting the payload.

Content Features: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts

Time-based Traffic Features: These features are designed to capture properties that mature over a 2 second temporal window. One example of such a feature would be the number of connections to the same host over the 2 second interval

Host-based Traffic Features: Utilize a historical window estimated over the number of connections – in this case 100 – instead of time. Host based features are therefore designed to assess attacks, which span intervals longer than 2 seconds

According to this paper the list of features that are the most important for the "normal" class are features 1, 6, 12, 15, 16, 17, 18, 19, 31, 32, 37. The features 2, 3, 5, 23, 24, 27, 28, 36, 40, 41 are the most relevant features for Smurf attacks and 4, 25, 26, 29, 30, 33, 34, 35, 38, 39 for Neptune attacks.

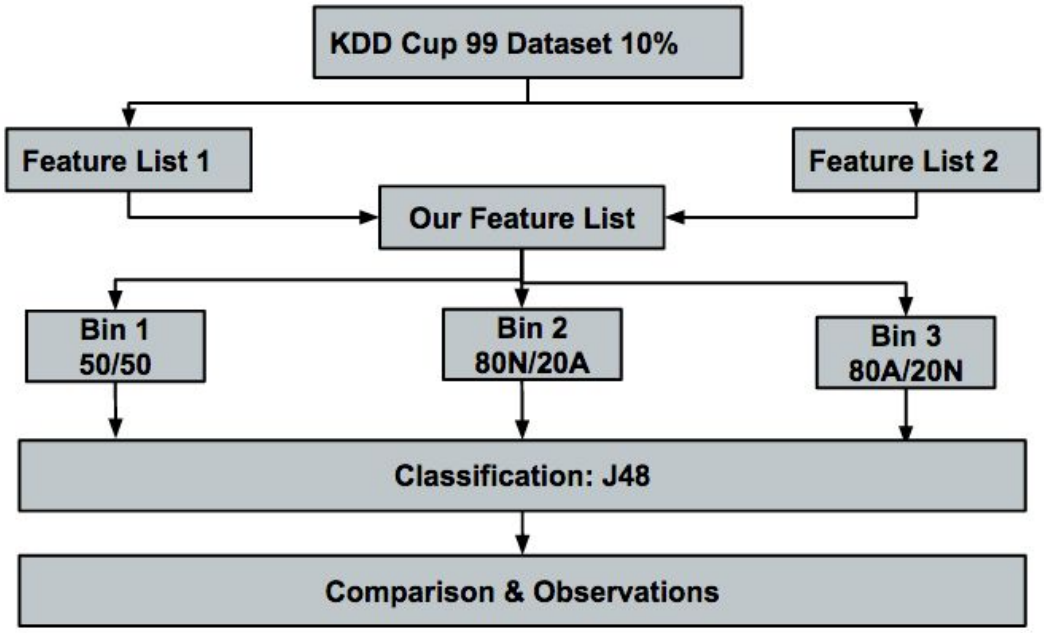
4. Methodology

The most relevant features for detecting DOS attacks were derived from the feature list obtained from the papers referred. We arrived at 24 common features from both the papers. CFSsubsetEval was applied on top of these common 24 features, which gave a subset of 9 attributes. The feature selection for classification in our project was done in two steps.

Firstly, CFSsubsetEval feature selection was used against each DoS attack. We got feature lists of different size and attributes for each attack. Since CFSsubsetEval does not allow ranking of

attribute, we had to manually rank attributes based on their occurrence. For instance, src_bytes was common in 5 out of 6 attacks and it was ranked high. Dst_bytes was occurring in only one attack so it was ranked least. Second, top 9 features that impact classifier were selected from the ranked list for classification.The reason behind using CFSsubsetEval is, it is proven it works well on small datasets in one of our reference paper.

4.1. Block Diagram



4.2. Preprocessing

- Attribute names were taken from the KDDCup official website.
- Redundant values in the data were removed using MS-excel(size of the data dropped to less than half of the original size).
- Attributes apart from the derived feature list are removed.
- The values in different features were in different range. So Normalisation was done on the data using **Normalization** filter in weka.
- To make the tree more readable, numeric values were transformed to nominal using **NumericToNominal** filter in weka.
- Data obtained was then used to create bins. The data was split into the following three bins using **SQL developer** tool.
  (i) Data with equal amounts of normal and attack datasets.
  (ii) Datasets with 80:20 percent normal and attack data respectively.
  (iii) Datasets with 80:20 percent attack and normal data respectively.

4.3. Classification

Classification is a data mining and machine learning technique used to predict group membership for data instances. [1] Here, we have used "**J48 Decision Tree**" algorithm using WEKA as the software tool. J48 is a java implementation of C4.5 algorithm. It makes J48 an efficient algorithm because it can handle both numeric and nominal values equally, it can handle missing attribute values and pruning trees after creation. J48 is proven efficient even in our experimentation. Please refer **additional work** section for details.

Examples of CFSSubSetEval feature relevance for each attack type:

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 2 FLAG):
        Information Gain Ranking Filter

Ranked attributes:
 0.69889    7 DST_HOST_SERROR_RATE
 0.69647    8 DST_HOST_SRV_SERROR_RATE
 0.03534    1 SERVICE
 0.01396    4 COUNT
 0.0017     5 SAME_SRV_RATE
 0          6 DST_HOST_DIFF_SRV_RATE
 0          3 SRC_BYTES
 0          9 ATTACK_TYPE

Selected attributes: 7,8,1,4,5,6,3,9 : 8
```

Eval on Neptune type

```
=== Attribute Selection on all input data ===

earch Method:
        Greedy Stepwise (forwards).
        Start set: no attributes
        Merit of best subset found:    0

ttribute Subset Evaluator (supervised, Class (
        CFS Subset Evaluator
        Including locally predictive attributes

elected attributes: 1 : 1
                    SERVICE
```

Eval on Land type

5. Tests and Results

After running the J48 classification method on the three binned data sets below is the result comparison of the results:

Binned DataSets	DataSet-1 (50-50)	DataSet-2 (80 normal 20 attack)	DataSet-3 (20 normal 80 attack)
Classification method	J48	J48	J48
Total Instances	103454	100745	54564
Correctly Classified Instances	103445 (99.9913%)	100736 (99.9911%)	54546 (99.967%)
Incorrectly Classified Instances	9 (0.0087%)	9 (0.0089%)	18 (0.033%)
Root mean squared error	0.0055	0.005	0.0097
Relative absolute error	0.03%	0.04%	0.11%
Root relative squared error	1.83%	2.27%	4.07%

Fig 1: Results comparison of all three datasets

a) DataSet 1 (Contains 50% Normal and 50% attack class types)

Data Set 1 has evenly distributed data between Normal and attacks types and produces a better result as compared to the other two datasets. We observe 99.9913% of correctly classified instances out of ~100,000 records. Only 9 instances were incorrectly classified. Dataset 1 managed to achieve a low root mean squared error (0.0055) and relative absolute error (0.03%) followed root relative squared error of 1.83% . These results while compared with dataset 2 and dataset 3 show that better results can be achieved with a balanced data distribution.

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

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        normal.
                 1         0          1         1         1          1        neptune.
                 0.895     0          1         0.895     0.944      0.953    land.
                 1         0          1         1         1          1        back.
                 0.99      0          1         0.99      0.995      0.996    pod.
                 1         0          1         1         1          1        smurf.
                 1         0          0.999     1         0.999      1        teardrop.
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
 48888     0     0     0     0     0     1 |     a = normal.
     4 51816     0     0     0     0     0 |     b = neptune.
     0     2    17     0     0     0     0 |     c = land.
     0     0     0   962     0     0     0 |     d = back.
     2     0     0     0   203     0     0 |     e = pod.
     0     0     0     0     0   641     0 |     f = smurf.
     0     0     0     0     0     0   918 |     g = teardrop.
```

Fig 2. Detailed accuracy report (dataset 1)

The above detailed accuracy report shows how the most occurring class types Normal, Neptune, Smurf and Back produce a precision and recall of 1 while the smaller attack types like land, pod and teardrop produce errors meaning, these classes are responsible for the incorrectly classified instances shown above. Since the largely occurring classes are correctly classified, we could infer this dataset goes on to produce a good result of classifying the normal classes from attack classes.

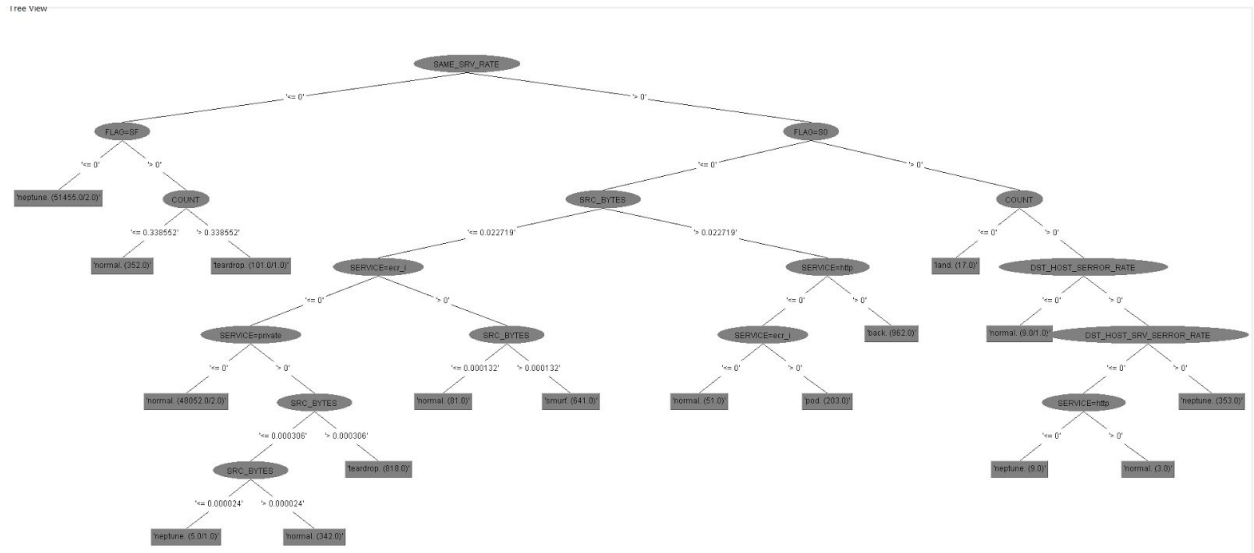


Fig 3. Decision tree (dataset1)

The classification method produces a viewable and understandable tree. The user friendly tree is a result from converting the Nominal values (Status and flag) to binary using *NominalToBinary* selection which creates a range between 0 and 1 for the specified attributes and creates the tree with better view and understandability.

b) DataSet 2 (Contains 80% Normal and 20% attack class type)

Dataset 2 is a combination of 80% Normal type and 20% attack type data. We created this dataset to see how the system behaves when we have majority Normal class and a mere 20% attack type. The results (fig 1) is similar to dataset 1 where it classifies 99.9911% of the instances correctly and 0.0089% of the instances incorrectly however, it produces a much higher error rate than dataset 1 with relative absolute error of 0.04% and root relative squared error of 2.27%. This shows how the system goes on to overtrain the Normal class and fails to identify attack types.

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

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        normal.
                 1         0          1         1         1          1        neptune.
                 1         0          1         1         1          1        smurf.
                 1         0          1         1         1          1        teardrop.
                 0.99      0          1         0.99      0.995      0.993    pod.
                 0.895     0          0.895     0.895     0.895      0.947    land.
                 1         0          1         1         1          1        back.
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
 79998     0     0     0     0     2     0 |     a = normal.
     3 17997     0     0     0     0     0 |     b = neptune.
     0     0   641     0     0     0     0 |     c = smurf.
     0     0     0   918     0     0     0 |     d = teardrop.
     2     0     0     0   203     0     0 |     e = pod.
     1     1     0     0     0    17     0 |     f = land.
     0     0     0     0     0     0   962 |     g = back.
```

Fig 4 . Detailed accuracy report (dataset 2)

The detailed accuracy report is similar to dataset 1 with the most occurring classes producing a good precision and no error rate. The confusion matrix shows how the result produces a True positive rate of 1 for big attack types like Neptune and Smurf while the False measure is coming from small attack types such as pod and land.

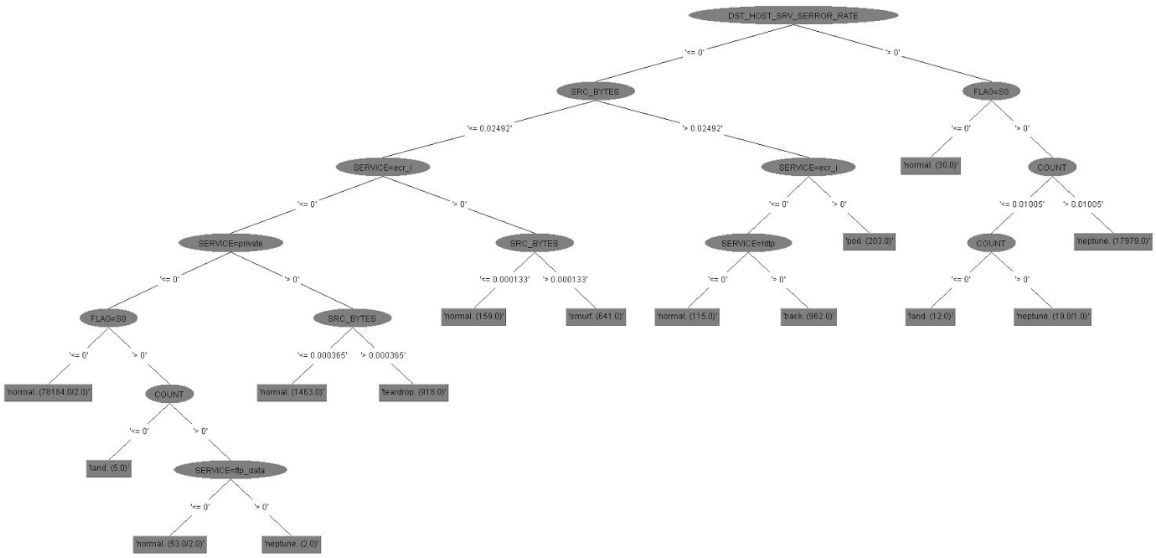


Fig 5. Decision tree (dataset 2)

The decision tree in fig 5 shows how a viewable and understandable tree can be achieved by converting the Nominal values from status and flag to binary. This decision tree can be used by a novice to analyse the distribution of classes and how each attribute affects the decision making.

c) DataSet 3 (Contains 20% Normal and 80% attack class type)

Lastly, dataset 3 stands out from the previous two datasets in terms of the results it gives to the difference in values it creates due to the uneven distribution of class types in the training data. Looking at fig 1 (in the comparison above) we can clearly see the drop in the detection rate of attacks types by the classification method J48. This is due to the fact that we have 80% data comprising of attacks types and just 20% data containing Normal class type. Some of the important observations in the results of dataset 3 are:

i) Correctly classified instances drops from 99.99 in the previous two datasets to 99.96 in this dataset.
ii) Incorrectly classified instances increases by 4 times as compared to the previous two datasets taking into consideration the size of the dataset being half of the previous two.
iii) What is even more is that even the Normal data seem to be incorrectly classified due to the imbalance in the data distribution (see Fig 6 Detailed accuracy by class (dataset 3) below).

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

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.997     0          1         0.997     0.998      0.999    back.
                 0.895     0          0.85      0.895     0.872      0.912    land.
                 1         0          1         1         1          1        neptune.
                 0.999     0          0.999     0.999     0.999      0.999    normal.
                 0.985     0          1         0.985     0.993      0.997    pod.
                 1         0          1         1         1          1        smurf.
                 0.998     0          0.997     0.998     0.997      0.999    teardrop.
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
   965     0     0     3     0     0     0 |     a = back.
     0    17     1     1     0     0     0 |     b = land.
     0     0 40898     0     0     0     2 |     c = neptune.
     0     3     2 10906     0     0     1 |     d = normal.
     0     0     0     3   203     0     0 |     e = pod.
     0     0     0     0     0   641     0 |     f = smurf.
     0     0     2     0     0     0   916 |     g = teardrop.
```

Fig 6: Detailed accuracy by class (dataset 3)

In the above fig it is evident that only Neptune and Smurf continue to produce a high precision and low error rate while the rest of the classes go on to produce error rates and true positive rates below 1.

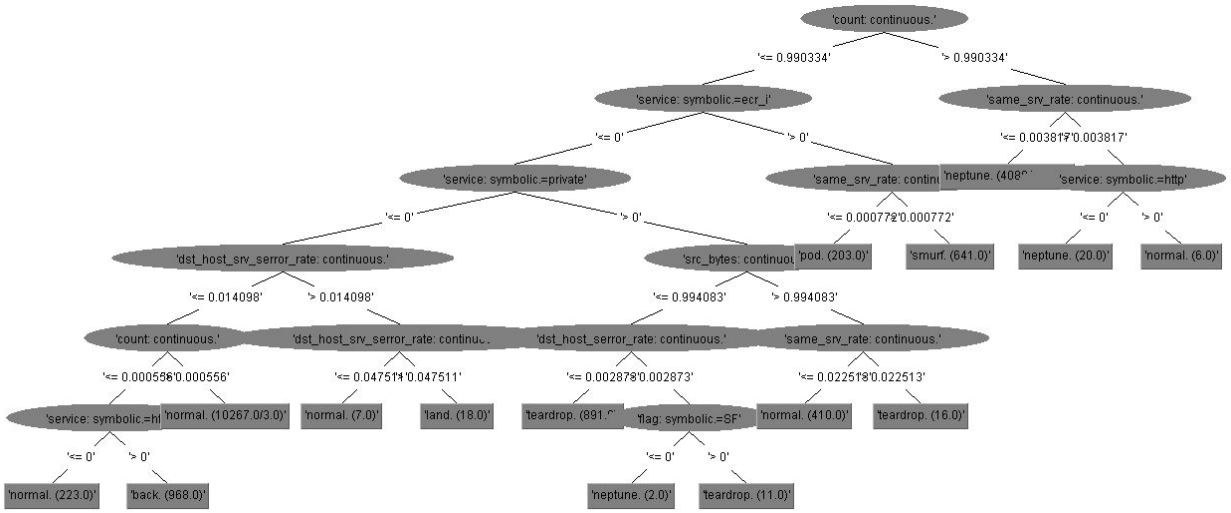


Fig 7: Decision tree (dataset 3)

6. Insights Gained

Some of the insights gained from this project after examining the results and comparing them were:

1) It is possible to achieve high precision and lower error rates by reducing the dimensionality of the dataset by half as compared to the related two paper works.
2) This feature set could be used as a base to improve the error rate and precision rate of smaller attack types such as land and teardrop to further improve the detection rate of DOS attacks in the future research.
3) By applying two different feature relevance ranking method (InfoGainAttributeEval and CFSSubsetEval) with one applied over the other we are able to achieve a much stronger feature set and produce better result set. To explain this better, by applying CFS Subset Evaluation on each DOS attack type than applying on the entire set of DOS attack types, we achieve a more relevant feature set. (Explained in Classification section)
4) With Normal, Neptune, Smurf and Back classes comprising of 98% of the KDDCUP data, it is possible achieve a precision of 1 and zero error rate by concentrating on the above said attack types. (shown in the detailed accuracy reports above)

7. Conclusion

Our results show that Normal, Smurf and Neptune classes comprising of 98% of the training data, their feature relevance makes a big impact and by achieving a high precision rate and lower error rate on these classes it is possible for machine learning algorithms to achieve a high detection rate percentage on DOS attack types while reducing the dimensionality of the data by 50%.

Observations:

1. DOS attack types, namely pod, teardrop, and land, tend to mostly fall under incorrectly classified instances while neptune, smurf, and back have a better accuracy by class.
2. If we remove neptune and smurf and back…will pod teardrop and land do better? It is observed that running J48 classification method, removing neptune, smurf and back attacks, and just retaining pod, teardrop and land the system is able to produce a better precision rate and classify more instances correctly.
3. By creating an imbalance in the data distribution of attack and normal types, Incorrectly classified instances in dataset 3 increases by 4 times as compared to the previous two datasets taking into consideration the size of the dataset being half of the previous two.
4. What is even more is that even the Normal data seem to be incorrectly classified due to the imbalance in the data distribution (see Fig 6 Detailed accuracy by class).

Additional work (Work done on KDDCUP prior to creating the above mentioned 3 datasets):

As explained in the presentation, we initially binned the 10% data set into five different bins to compare their individual behaviour on each classification type namely,

a) J48 decision tree
b) Naive Bayes Tree (NBTree)

And below is the result comparison we made:

Binned DataSets	DataSet-1		DataSet-2		DataSet-3		DataSet-4		DataSet-5	
Classification method	J48	NBT	J48	NBT	J48	NBT	J48	NBT	J48	NBT
Total instances	58102		66492		57427		38881		50536	
Correctly Classified Instances	58096 (99.9897%)	58096 (99.9897%)	66487 (99.9925%)	66485 (99.9895%)	57423 (99.993%)	57417 (99.9826%)	38877 (99.9897%)	38877 (99.9897%)	50528 (99.9842%)	50531 (99.9901%)
ncorrectly Classified Instance	6 (0.0103%)	6 (0.0103%)	5 (0.0075%)	7 (0.0105%)	4 (0.007%)	10 (0.0174%)	4 (0.0103%)	4 (0.0103%)	8 (0.0158%)	5 (0.0099%)
Root mean squared error	0.0055	0.005	0.005	0.0061	0.0045	0.0065	0.0054	0.0053	0.0067	0.0051
Relative absolute error	0.05%	0.06%	0.02%	0.04%	0.02%	0.06%	0.03%	0.05%	0.06%	0.04%
Root relative squared error	2.62%	2.42%	1.74%	2.16%	1.70%	2.41%	2.00%	1.59%	2.49%	1.88%
Unknown instances	73	73	NA	NA	NA	NA	NA	NA	NA	NA

Data sets 1 - 5 run on J48:

```
Time taken to build model: 7.37 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       58096               99.9897 %
Incorrectly Classified Instances         6                0.0103 %
Kappa statistic                          0.9997
Mean absolute error                      0
Root mean squared error                  0.0055
Relative absolute error                  0.0478 %
Root relative squared error              2.6239 %
Total Number of Instances            58102
Ignored Class Unknown Instances             73

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        neptune.
                 1         0          1         1         1          1        normal.
                 1         0          1         1         1          1        smurf.
                 1         0          1         1         1          1        back.
                 0         0          0         0         0          0.49     land.
                 1         0          1         1         1          1        pod.
                 1         0          0.976     1         0.988      1        teardrop.
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===
```

Dataset 1 (J48)

```
Time taken to build model: 14.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       66487               99.9925 %
Incorrectly Classified Instances         5                0.0075 %
Kappa statistic                          0.9998
Mean absolute error                      0
Root mean squared error                  0.005
Relative absolute error                  0.0222 %
Root relative squared error              1.7422 %
Total Number of Instances            66492

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        neptune.
                 1         0          1         1         1          1        normal.
                 1         0          1         1         1          1        smurf.
                 1         0          1         1         1          1        back.
                 1         0          1         1         1          1        pod.
                 1         0          0.952     1         0.976      1        teardrop.
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f   <-- classified as
 26417     0     0     0     0     0 |     a = neptune.
     3 39706     0     0     0     2 |     b = normal.
     0     0   128     0     0     0 |     c = smurf.
     0     0     0   192     0     0 |     d = back.
     0     0     0     0     4     0 |     e = pod.
     0     0     0     0     0    40 |     f = teardrop.
```

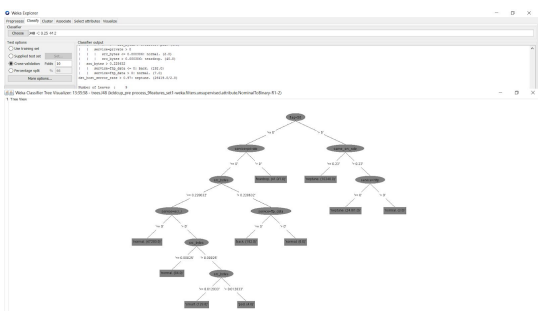
Dataset 2 (J48)

```
Time taken to build model: 11.86 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       57423               99.993  %
Incorrectly Classified Instances         4                0.007  %
Kappa statistic                          0.9999
Mean absolute error                      0
Root mean squared error                  0.0045
Relative absolute error                  0.0235 %
Root relative squared error              1.6962 %
Total Number of Instances            57427
Ignored Class Unknown Instances            982

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        neptune.
                 1         0          1         1         1          1        normal.
                 1         0          1         1         1          1        smurf.
                 1         0          1         1         1          1        back.
                 0         0          0         0         0          0.26     land.
                 1         0          1         1         1          1        pod.
                 0.975     0          1         0.975     0.987      0.987    teardrop.
Weighted Avg.    1         0          1         1         1          1
```

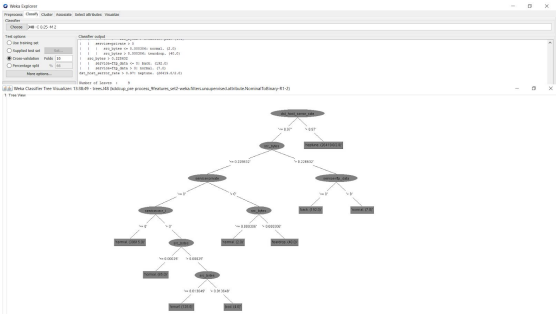
Dataset 3 (J48)

```
Time taken to build model: 8.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       38877               99.9897 %
Incorrectly Classified Instances         4                0.0103 %
Kappa statistic                          0.9998
Mean absolute error                      0
Root mean squared error                  0.0054
Relative absolute error                  0.0262 %
Root relative squared error              2.0049 %
Total Number of Instances            38881

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        neptune.
                 1         0          1         1         1          1        normal.
                 1         0          1         1         1          1        smurf.
                 1         0          1         1         1          1        back.
                 1         0          1         1         1          1        land.
                 1         0          1         1         1          1        pod.
                 1         0          1         1         1          1        teardrop.
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===
```

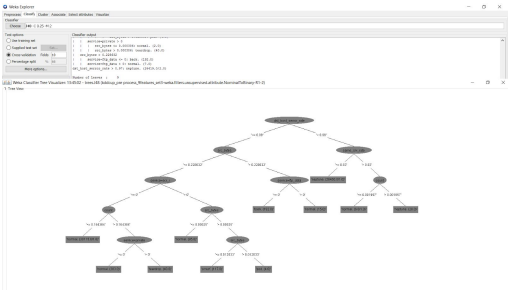
Dataset 4 (J48)



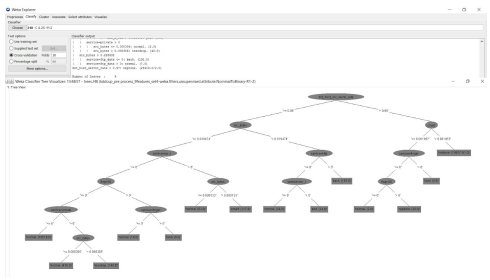
Dataset 1 decision tree (J48)



Dataset 2 decision tree (J48)



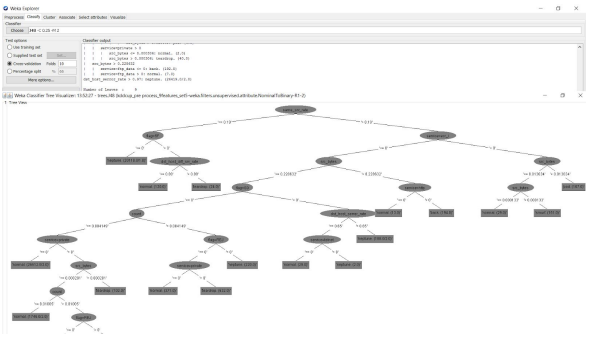
Dataset 3 Decision tree (J48)



Dataset 4 Decision tree (J48)

```
Time taken to build model: 64 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      142380               99.9881 %
Incorrectly Classified Instances        17                0.0119 %
Kappa statistic                          0.9998
Mean absolute error                      0
Root mean squared error                  0.0058
Relative absolute error                  0.035  %
Root relative squared error              2.1899 %
Total Number of Instances           142397

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        normal.
                 1         0          1         1         1          1        neptune.
                 1         0          1         1         1          1        smurf.
                 0.99      0          1         0.99      0.995      0.995    pod.
                 1         0          0.998     1         0.999      1        teardrop.
                 0.789     0          0.938     0.789     0.857      0.908    land.
                 1         0          1         1         1          1        back.
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
 87826     3     0     0     2     1     0 |     a = normal.
     5 51815     0     0     0     0     0 |     b = neptune.
     0     0   641     0     0     0     0 |     c = smurf.
     2     0     0   203     0     0     0 |     d = pod.
     0     0     0     0   918     0     0 |     e = teardrop.
     2     2     0     0     0    15     0 |     f = land.
     0     0     0     0     0     0   962 |     g = back.
```

Data Set all (J48)



Dataset 5 (J48)

Data sets 1 - 5 run on NBTree:

```
Time taken to build model: 27.31 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       58096               99.9897 %
Incorrectly Classified Instances         6                0.0103 %
Kappa statistic                          0.9997
Mean absolute error                      0.0001
Root mean squared error                  0.005
Relative absolute error                  0.0627 %
Root relative squared error              2.4226 %
Total Number of Instances            58102
Ignored Class Unknown Instances             73

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        neptune.
                 1         0          1         1         1          1        normal.
                 1         0          1         1         1          1        smurf.
                 1         0          1         1         1          1        back.
                 0         0          0         0         0          1        land.
                 1         0          1         1         1          1        pod.
                 1         0          0.976     1         0.988      1        teardrop.
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
 10362     1     0     0     0     0     0 |     a = neptune.
     3 47370     0     0     0     0     1 |     b = normal.
     0     0   128     0     0     0     0 |     c = smurf.
     0     0     0   192     0     0     0 |     d = back.
     0     1     0     0     0     0     0 |     e = land.
     0     0     0     0     0     4     0 |     f = pod.
     0     0     0     0     0     0    40 |     g = teardrop.
```

Dataset 1 (NBTree)

```
Time taken to build model: 14.51 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       66485               99.9895 %
Incorrectly Classified Instances         7                0.0105 %
Kappa statistic                          0.9998
Mean absolute error                      0.0001
Root mean squared error                  0.0061
Relative absolute error                  0.0403 %
Root relative squared error              2.162  %
Total Number of Instances            66492

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        neptune.
                 1         0          1         1         1          1        normal.
                 1         0          0.992     1         0.996      1        smurf.
                 1         0          1         1         1          1        back.
                 0.75      0          1         0.75      0.857      0.987    pod.
                 1         0          0.952     1         0.976      1        teardrop.
Weighted Avg.    1         0          1         1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f   <-- classified as
 26417     0     0     0     0     0 |     a = neptune.
     4 39705     0     0     0     2 |     b = normal.
     0     0   128     0     0     0 |     c = smurf.
     0     0     0   192     0     0 |     d = back.
     0     0     1     0     3     0 |     e = pod.
     0     0     0     0     0    40 |     f = teardrop.
```

Dataset 2 (NBTree)

```
Time taken to build model: 13.62 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       57417               99.9826 %
Incorrectly Classified Instances        10                0.0174 %
Kappa statistic                          0.9997
Mean absolute error                      0.0001
Root mean squared error                  0.0065
Relative absolute error                  0.0559 %
Root relative squared error              2.4107 %
Total Number of Instances            57427
Ignored Class Unknown Instances            982

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1        1         1          1        neptune.
                 1         0          1        1         1          1        normal.
                 0.983     0          1        0.983     0.991      1        smurf.
                 0.995     0          1        0.995     0.997      1        back.
                 0         0          0        0         0          0.993    land.
                 1         0          0.667    1         0.8        1        pod.
                 1         0          1        1         1          1        teardrop.
Weighted Avg.    1         0          1        1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
 26469     0     0     0     0     0     0 |     a = neptune.
     3 30598     0     0     1     1     0 |     b = normal.
     0     1   115     0     0     1     0 |     c = smurf.
     0     1     0   191     0     0     0 |     d = back.
     1     1     0     0     0     0     0 |     e = land.
     0     0     0     0     0     4     0 |     f = pod.
     0     0     0     0     0     0    40 |     g = teardrop.
```

Dataset 3 (NBTree)

```
Time taken to build model: 16.56 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       38877               99.9897 %
Incorrectly Classified Instances         4                0.0103 %
Kappa statistic                          0.9998
Mean absolute error                      0.0001
Root mean squared error                  0.0053
Relative absolute error                  0.0483 %
Root relative squared error              1.9586 %
Total Number of Instances            38881

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1        1         1          1        neptune.
                 1         0          1        1         1          1        normal.
                 1         0          1        1         1          1        smurf.
                 0.995     0          1        0.995     0.997      1        back.
                 1         0          1        1         1          1        land.
                 1         0          0.923    1         0.96       1        pod.
                 0.993     0          1        0.993     0.996      0.996    teardrop.
Weighted Avg.    1         0          1        1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
 19856     0     0     0     0     0     0 |     a = neptune.
     0 18536     0     0     0     2     0 |     b = normal.
     0     0   117     0     0     0     0 |     c = smurf.
     0     1     0   191     0     0     0 |     d = back.
     0     0     0     0    14     0     0 |     e = land.
     0     0     0     0     0    24     0 |     f = pod.
     0     1     0     0     0     0   139 |     g = teardrop.
```

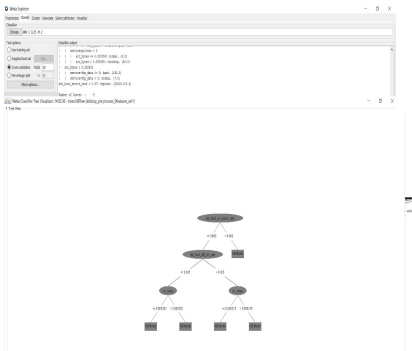
Dataset 4 (NBTree)

```
Time taken to build model: 39.88 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       50531               99.9901 %
Incorrectly Classified Instances         5                0.0099 %
Kappa statistic                          0.9998
Mean absolute error                      0.0001
Root mean squared error                  0.0051
Relative absolute error                  0.0437 %
Root relative squared error              1.8784 %
Total Number of Instances            50536

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1        1         1          1        neptune.
                 1         0          1        1         1          1        normal.
                 1         0          1        1         1          1        smurf.
                 0.995     0          1        0.995     0.997      1        back.
                 1         0          1        1         1          1        teardrop.
                 0.994     0          1        0.994     0.997      1        pod.
                 0         0          0        0         0          0.97     land.
Weighted Avg.    1         0          1        1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
 20533     0     0     0     0     0     0 |     a = neptune.
     1 28828     0     0     0     0     0 |     b = normal.
     0     0   151     0     0     0     0 |     c = smurf.
     0     1     0   193     0     0     0 |     d = back.
     0     0     0     0   658     0     0 |     e = teardrop.
     0     1     0     0     0   168     0 |     f = pod.
     2     0     0     0     0     0     0 |     g = land.
```

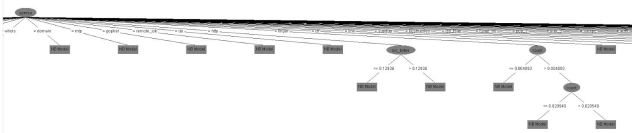
Dataset 5 (NBtree)

```
Time taken to build model: 88.7 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      142376               99.9853 %
Incorrectly Classified Instances        21                0.0147 %
Kappa statistic                          0.9997
Mean absolute error                      0.0001
Root mean squared error                  0.0056
Relative absolute error                  0.038  %
Root relative squared error              2.1103 %
Total Number of Instances           142397

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1        1         1          1        normal.
                 1         0          1        1         1          1        neptune.
                 1         0          0.998    1         0.999      1        smurf.
                 0.99      0          0.981    0.99      0.985      1        pod.
                 1         0          1        1         1          1        teardrop.
                 0.842     0          0.8      0.842     0.821      1        land.
                 1         0          1        1         1          1        back.
Weighted Avg.    1         0          1        1         1          1

=== Confusion Matrix ===

     a     b     c     d     e     f     g   <-- classified as
 87818     5     1     4     0     4     0 |     a = normal.
     2 51618     0     0     0     0     0 |     b = neptune.
     0     0   641     0     0     0     0 |     c = smurf.
     2     0     0   203     0     0     0 |     d = pod.
     0     0     0     0   918     0     0 |     e = teardrop.
     1     2     0     0     0    16     0 |     f = land.
     0     0     0     0     0     0   962 |     g = back.
```

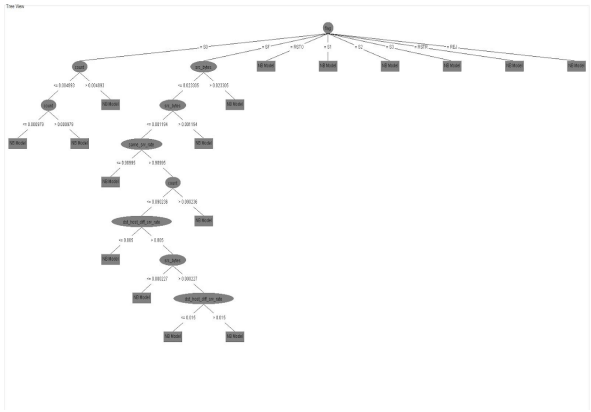
Dataset All DOS (NBTree)



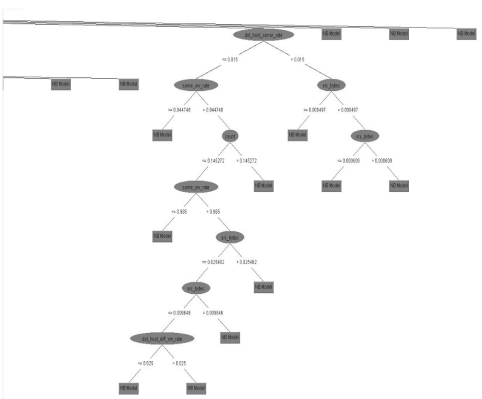
Decision tree 1 (NBTree)



Decision tree 2 (NBtree)



Decision tree 3 (NBTree)



Decision tree 4 (NBtree)

Analysis of results:

By looking at the test results above we can infer that even though NBTree applies classification on the leaf nodes first and then applies the Naive Bayes model on top of it to classify the attacks as normal or DOS attack type, fails to achieve a consistent result as compared to the J48 classification method. The detection rates achieved using NBTree was inconsistent as compared to J48 which produced a more consistent result.

Also NBTree consumes more times to build the tree and the model as compared to J48 (at least 4 times) and this can be a big impact while running big datasets.

This work led us to change our approach on binning the data in a more sensible way in the lines of distribution of attack classes and then apply J48 on all three datasets.

8. References

DARPA Intrusion Detection Evaluation. (n.d.). Retrieved May 06, 2016, from https://www.ll.mit.edu/ideval/docs/attackDB.html#teardrop

Kayacik, G., Zincir-Heywood, N., & Heywood, M. (n.d.). Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets. doi:10.1075/ps.5.3.02chi.audio.2f

KDD Cup 1999 Data. (n.d.). Retrieved May 06, 2016, from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Olusola, A. A., Oladele, A. S., & Abosede, D. O. (n.d.). Analysis of KDD '99 Intrusion Detection Dataset for Selection of Relevance Features. Retrieved May 6, 2016, from http://www.iaeng.org/publication/WCECS2010/WCECS2010_pp162-168.pdf

Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets - H. Günes Kayacık, A. Nur Zincir-Heywood, Malcolm I. Heywood