MODELLING INTRUSION DETECTION: ANALYSIS OF A FEATURE SELECTION MECHANISM

ABSTRACT:

Many Intrusion Detection Systems (IDS) has been proposed in the current decade. To evaluate the effectiveness of the IDS Canadian Institute of Cybersecurity presented a state of art dataset named CICIDS2017, consisting of latest threats and features. Several studies have suggested that by selecting relevant features for intrusion detection system, it is possible to considerably improve the detection accuracy and performance of the detection engine with simpler models rather than complex one like deep neural network. In a large dataset not all features contribute to represent the traffic, therefore reducing and selecting a number of adequate features may improve the speed and accuracy of the intrusion detection system. To achieve that objective, a manual and a recursive feature elimination process was employed and associated with a several machine learning models and the relevant features were identified inside the dataset which helps in the reduction of memory consumption and obtaining the results in optimal time. These results lend to support the idea that feature selection, improve significantly the classifier performance with simple model rather than complex one. Understanding the factors that help identify relevant features will allow the design of a better intrusion detection system.

Keywords—classification; logistic regression; decision tree; random forest; features selection; intrusion detection system;

I. INTRODUCTION

With the recent advance in technologies where concepts like Cloud Computing, Big Data, and Social Media Network have emerged, our society produce enormous quantity of data. Finding useful information among this immense data generated by these technologies became critical for marketers, data scientist and even business corporate. With this amount of data transmitted over a network or internet, security becomes a major concern, although multiple intrusion prevention technologies have been built in the past decade to eliminate potential threats despite that, attacks still continue and increase in complexity, this is the reason there is a need of a mechanism to detect any suspicious or unwanted traffic which may cause damage on a particular network.

This security mechanism can be implemented using an Intrusion Detection System (IDS) which can be describe as a collection of software or hardware device

able to collect, analyze and detect any unwanted, suspicious or malicious traffic either on a particular computer host or network. Therefore to achieve its task, an subsequently reports any malicious activity to the network administrator.

There still exist one main issue regarding the actual intrusion detection technique that is the involvement of human interaction when it comes to label the traffic between an intrusion and a normal one, another major concern is the new challenge of “Big Data” and “Cloud Computing”. These two ubiquitous technologies produce a large amount of data that must be collected and analyzed by the intrusion detection engine dynamically and often the IDS needs to deal with a multi-dimensional data generated by these large quantities of data. It is necessary to consider that the intrusion dataset can be huge in size, not only the number of observations grown, but the number of observed attributes can also increase significantly and may generated a considerably amount of false positives results as it can contain many redundant or duplicate records

Machine learning helps to optimize performance criterion using example data or past experience using a computer program, models are defined with some parameters, and learning is the execution of the programming computer to optimize the parameters of the model using a training data. The model can be predictive to make predictions in the future, or descriptive to gain knowledge from data. To perform a predictive or descriptive task, machine learning generally use two main techniques: Classification and Clustering. In classification, the program must predict the most probable category, class or label for new observation into one or multiple predefined classes or label while IDS should use some statistical or mathematical method to read and interpret the information it collects and

However if the purpose of the IDS is to differentiate between normal or intrusion traffic, classification is recommended and if we seeks to identify the type of intrusion, clustering can be more helpful.

However, a lot of researchers have suggested to use the KDD dataset to detect an attack in the past. Unfortunately, these proposal have failed to ensure a good performance in terms of detection rate. Moreover those existing IDS aims to analyze all features which can result a misclassification of intrusion and quite amount of time when building the model, despite some concern and critic about the system evaluation of the KDD dataset, research still use it to test their model. Thus, in this paper a method has been suggested for selecting and identifying relevant features on the CICIDS2017 dataset.

The rest of this paper is divided as followed: Section II – Description of the CICIDS2017 Dataset, Section III- Previous works, Section IV – Methodology, Section V- Experiment and Results and finally the Section VI- Conclusion and Further Works.

II. DESCRIPTION OF CICIDS2017

Since the inception of CICIDS2017 dataset, the dataset started attracting researchers for analysis and developments of new models and algorithms. We found the shape of a dataset is 692703 instances and 79 features containing 6 class labels. Further, examining the dataset, it has been found

that the dataset contains high bias on the BENIGN class with a instance of 440031. After performing a under sampling in the dataset 949 instances of missing information is found. By removing such missing instances, we found a dataset of CICIDS2017 having 251734 instances. At this moment we queried for any possible redundant instances. We found 12 features with redundant information and we removed it from the dataset. The characteristics of dataset and the detailed class wise occurrence has been presented in the following table.

Table 1: Overall characteristics of CICIDS2017 dataset

DATASET NAME	CICIDS2017
DATASET TYPE	MULTICLASS
YEAR OF RELEASE	2017
TOTAL NUMBER OF RECORDS	692703
NUMBER OF FEATURES	79
NUMBER OF TARGET CLASS	6

Table 2: Class wise instance occurrence of CICIDs2017 dataset

CLASS LABEL	NUMBER OF INSTANCES
BENIGN	440031
DoS Hulk	231073
DoS GoldenEye	10293
DoS slowloris	5796
DoS Slowhttptest	5499
Heartbleed	11

2.1 Shortcomings of CICIDS2017

a. High class imbalance

It can be seen from the table that the prevalence of majority class (Benign) is 63.54% where as for the minority class is 0.000015% (Heartbleed). In such a huge difference of prevalence rate a potential detector may tilt towards benign. The situation becomes worst when a detector is based on a sample of this dataset. It is because, when a random sample on this dataset is ascertained for training and testing of a detector there is a great possibility that instances of a particular attack label such as “Heartbleed” may not found in training Set. As a result, the detector will fail to detect such attack when an instance of type such attack arrives. This is a major drawback that we have noticed in CICIDS2017 dataset.

b. Huge Volume of Data

We noticed that the dataset contains data of all the possible recent attack labels at one place. But, at the same time the size of a dataset becomes huge. This huge volume of data itself becomes a shortcoming. The shortcoming is that it consumes more overhead for loading and processing

2.2 Our Solution:

The missing values has been removed. Though, huge volume of data is a short-coming for a dataset but at the

same time it is inherent to any typical dataset that contains typical information. This shortcoming of high volume can be overcome by sampling the dataset before actual detection process starts. However, it is strongly advised that before sampling the class imbalance problem must be addressed before hand. If the dataset will be balanced the probability of occurrence of instances of all class label will be increased. There are many ways to handle class imbalance problem of a dataset. One of the major ways to resolve class imbalance issue is by using an under sampler. Relabeling of classes includes splitting of majority classes to form more classes or merging of few minority classes to form a class; thus; improving prevalence ratio and reducing class imbalance issue. In the case of CICIDS2017 it is very difficulty to split majority classes to form discrete classes equivalent to minority classes. It is because, the difference in prevalence is 63.54367%, which seems to be too high.

III. PREVIOUS WORK

Most of the proposed research system has used a complex deep neural network that could effectively utilize feature selection process to improve detection rate of their system and minimize the error. Research usually missed to detect new intrusions, especially when the intrusion mechanism used differed from the previous intrusion. In 2009, Shi-Jinn works revealed that not all research carried out feature selection before they trained their classifier, however based on this processes takes a significant part to different types of intrusion identification and features can be excluded without the performance of the IDS to be dropped. Juan Wang et al., in their work proposed a decision tree based algorithm for intrusion detection, even if during their experiments the C4.5 algorithm was achieving a good detection accuracy, the error rate was remaining identical.

Back in 2010, Farid et al. [18], used a decision tree based learning algorithm to retrieve important features set from the training dataset for intrusion detection. Their techniques found relevant features using a combination of ID3 and C4.5 decision tree algorithms. They assigned a weight value to each features. The weight is determined where the minimum depth of the decision tree at which each feature is checked inside the tree and the weights of features that do not appear in the decision tree are allocated a value of zero. Ektefa et al. [19], used different data mining method for intrusion detection and they found that the decision tree classifier was performing better than the SVM learning algorithm. Geetha Ramani et al. [20] used in their paper in 2011, a statistical method for analyzing the KDD 99 dataset. They identified the important features by studying the internal dependences between features. In their paper proposed in 2012, S. Mukherjee and N. Sharma [21] designed a technique called Feature-Vitality Based Reduction Method (FVBRM) using a Naïve Bayes classifier. FVBRM identifies important features by using a sequential search approach, starting with all features, one feature is removed at a time until the accuracy of the classifier reaches some threshold. Their method shows an improvement of the classification accuracy but takes more time and still complex when detecting the U2R attacks. In 2013, support vector machine classifier was used by Yogita B. Bhavsar et al. [22], for intrusion detection using the NSL KDD dataset. The drawback with this technique is the extensive training time required by the

classifier, so to reduce the time, they applied a radial basis function (RBF) to reduce the extensive time. In 2014, O. Y. Al-Jarrah et al. [23], used an ensembles of decision-tree based voting algorithm with forward selection / backward elimination feature raking techniques using a Random Forest Classifier. Their method shows an improvement of detection accuracy when selected important features and it can be suitable for large-scale network.

N. G. Relan and D. R. Patil [24] in their papers have tested two decision tree approach to classify attacks using the NSL KDD dataset. They have found that the C4.5 with pruning offers better accuracy than the C4.5 without pruning and it was necessary to reduce the number of features because using all features degrades the performance of the classifier also its time consuming. After analyzing some previous works, the reasons most of researchers are interested in selecting and identifying relevant features are described as follow:

- In most learning algorithms, the complexity depends on the number of input dimensions, d, as well as on the size of the data sample, N, and for reduced memory and computation, researchers are interested in selecting relevant and important feature to reduce the dimensionality of the problem. Decreasing d also decreases the complexity of the inference algorithm during testing.
- When an input is decided to be unnecessary, the cost of extracting it is saved.
- Simpler models are more robust on small datasets.
- Simpler models have less variance, that is, they vary less depending on the particulars of a sample, including noise, outliers, and so forth.

When data can be explained with fewer features, better idea about the process that underlies the data can be obtained and this allows knowledge extraction. When data can be represented in a few dimensions without loss of information, it can be plotted and analyzed visually for structure and outliers.

IV. METHODOLOGY

A Scikit-Learn Description : During this experiment scikit-learn was used, which is a machine learning library written in python. Most of the learning algorithm implement in scikit learn required data to be stored in a two-dimensional array or matrix. The size of the expected matrix is [samples, features].The first parameter defines the number of samples, each sample is an item to be processed and the second parameter is the number of features that can be used to describe each item in a quantitative manner, generally real-valued but may be Boolean or discrete-valued in some cases. Data in scikit-learn is represented as a feature matrix and a label vector.

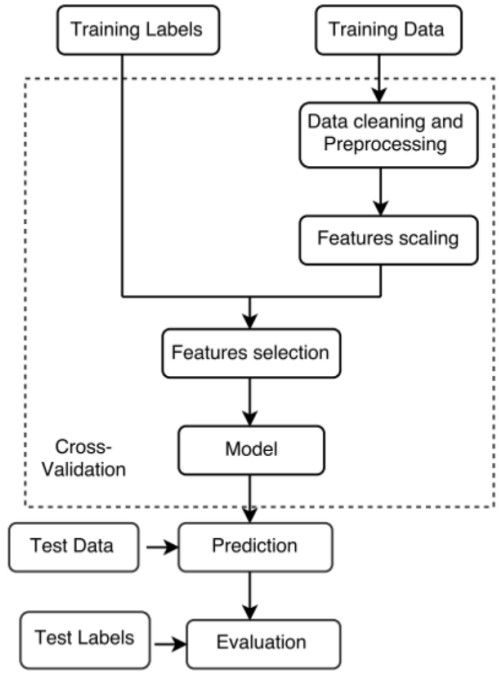
a. Experiment Methodology

The experiment methodology used in this paper, is illustrated in the Figure below and describe as follow:

Step 1: Data Cleaning and Data Preprocessing

Basically in this step the dataset has to go through a cleaning process to remove duplicate or redundant records. Next a Pre-processing operation has to be taken in place because the dataset contains numerical and non-numerical instances. Generally scikit-learn works

well with numerical inputs, so a Label encoding method is used to make that transformation. This technique will transforms each categorical feature to numerical ranked value.



Step 2: Features scaling

Features scaling is a common requirement of machine learning methods, to avoid that features with large values may weight too much on the final results. For each feature, calculate the average, subtract the mean value from the feature value, and divide the result by their standard deviation. After scaling, each feature will have a zero average, with a standard deviation of one.

Step 3: Features Selection

Feature selection is used to eliminate the redundant and irrelevant data. It is a technique of selecting a subset of relevant features that fully represents the given problem alongside a minimum deterioration of presentation, two possible reason were analyzed why it would be recommended to restrict the number of features:

Firstly, it is possible that irrelevant features could suggest correlations between features and target classes that arise just by chance and do not correctly model the problem. This aspect is also related to overfitting, usually in a decision tree classifier. Secondly, a large number of features could greatly increase the computation time without a corresponding classifier improvement.

The feature selection process is carried out in two ways, One by manual feature selection process and the other by using recurrent feature elimination in sklearn.feature_selection.

MANUAL FEATURE SELECTION

The manual process has two major divisions. One is for numerical features and the other is for categorical features.

a.CORRELATION:

The first step of feature selection starts with the correlation of the numerical features. Correlation is a statistical measure that expresses the extent to which two variables are linearly related (meaning they change together at a constant rate). It’s a common tool for describing simple relationships without making a statement about cause and effect.

- We describe correlations with a unit-free measure called the correlation coefficient which ranges from -1 to +1 and is denoted by *r*.

- The closer *r* is to zero, the weaker the linear relationship.
- Positive *r* values indicate a positive correlation, where the values of both variables tend to increase together.
- Negative *r* values indicate a negative correlation, where the values of one variable tend to increase when the values of the other variable decrease.

We have set a 4 different correlation value (0.95, 0.90, 0.85, 0.80) among the numerical features to find the features having multicollinearity in them. We have found that most of the features has a correlation value greater than these threshold and we tend to remove them from the dataset in order to avoid multicollinearity. In such way we got 4 dataset which corresponds to the different correlation value.

b. CRAMER-V TEST

The second step of feature selection is the Cramer's V test for the categorical features to find the internal association between them. Cramér's V (denoted as φ_c) is a measure of association between two categorical variables, giving a value between 0 and +1 (inclusive). It is based on Pearson's chi-squared statistic and was published by Harald Cramér in 1946.We have run the test and removed the columns with higher cramer value.

RECURRENT FEATURE ELIMINATION

The SelectPercentile method in the sklearn.feature_selection module were used, this method select features based on a percentile of the highest scores. Once, the best subset of features were found, a recursive feature elimination was applied which repeatedly build a model, placing the feature aside and then repeating the process with the remained features until all features in the dataset are exhausted. As such, it is a good optimization for finding the best performing subset of features. The idea is to use the weights of a classifier to produce a feature ranking.

A good feature ranking criterion does not necessarily produce a good feature subset generation. The some criteria estimate the effect of removing one feature at a time based on the goal to achieve. They become very suboptimal when it comes to removing several features at a time, which is necessary to obtain a small feature subset. This problem can be overcome by using the following iterative procedure that is Recursive Feature Elimination:

- Train the classifier (optimize the weights of features with respect to criterion).
- Compute the ranking criterion for all features.
- Remove the feature with smallest ranking criterion.

Step 4: Model

1.Logistic regression

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can

model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique.

The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables. This is due to applying a nonlinear log transformation to the odds ratio.

2. Decision tree

Decision tree model was built to partition the data using information gain until instances in each leaf node have uniform class labels. This is a very simple but yet effective hierarchical method for supervised learning (classification or regression) whereby the local space (region) is recognized in a sequence of repetitive splits in a reduced number of steps (small). At each test, a single feature is used to split the node according to the feature's values. If after the split, for every branch, all the instances selected belong to the similar class, the split is considered complete or pure.

One of the possible methods to measure a good split is entropy or information gain. Entropy is an information theoretic measure of the ‘uncertainty’ found in a training set, because of the existence of more than one possible classification. The training set entropy is represented by H. It is calculated in ‘bits’ of information.

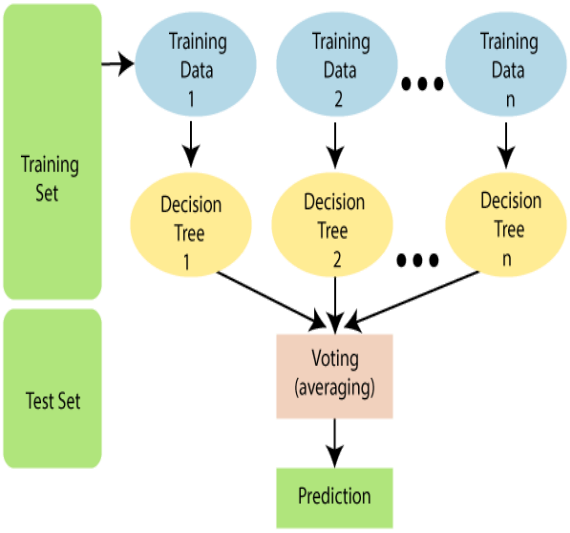
The generation process of a decision tree done by recursively splitting on features is equivalent to dividing the original training set into smaller sets recursively until the entropy of every one of these subsets is zero (i.e everyone will have instances from a single class target). A Decision Tree is made up of internal decision nodes and terminal leaves. A test function is implemented by each decision node with discrete results labelling the branches. Providing an input, at every node, a test is constructed and based on the outcome, one of the branches will be considered. Here the learning algorithm starts at the root and until a leaf node is reached, the process will be done recursively at which moment the value represented in the leaf node is the output. Every leaf node possesses an outcomes label, which it is the class target in case of classification and numeric value for regression. A leaf node can describe a localized space or region where instances finding in this input space (region) possess the same labels for classification and similar numeric value for regression

3.Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of

predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



Step 5: Prediction and Evaluation

The test data was used to make prediction of our model and for evaluation, multiple settings was considered such as the accuracy score, precision, recall, f-measure and a confusion matrix. A 10-fold cross-validation was performed during all the process.

V. EXPERIMENT AND RESULTS

5.1 Experiment

The above mentioned algorithm was used in the experiment. The adequate number of features was selected in manual selection process and in a recursive feature elimination (RFE). RFE was operated with the number of features passed as parameter to identify the feature importance. During the RFE process, first, the classifier is trained on the original set of features and weights are attributed to each features. Then, features whose absolute weights are the smallest are pruned from the current set features. That process is recursively repeated on the pruned set until the desired number of features to choose is finally reached. After the process of feature selection the above mentioned models has been built and its performance is analysed.

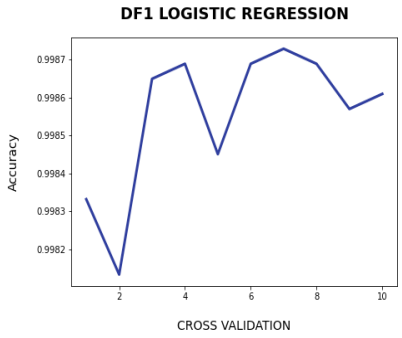
5.2 Discussions and Results

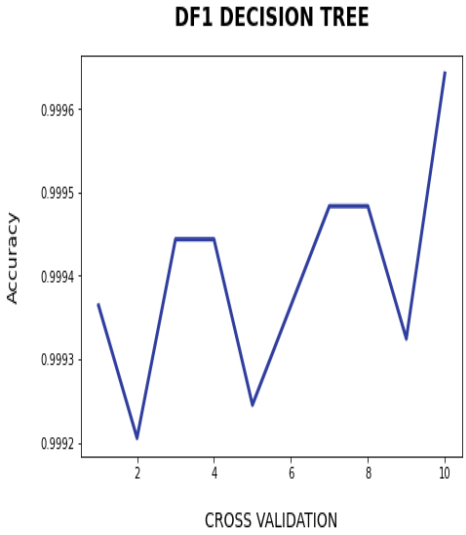
Feature selection is utilized to discriminate the redundant and irrelevant data. It is a technique of selecting a subset of relevant attributes that completely represent the given problem alongside a minimum deterioration of presentation. As consequence, working with a small number of feature may bring better results.

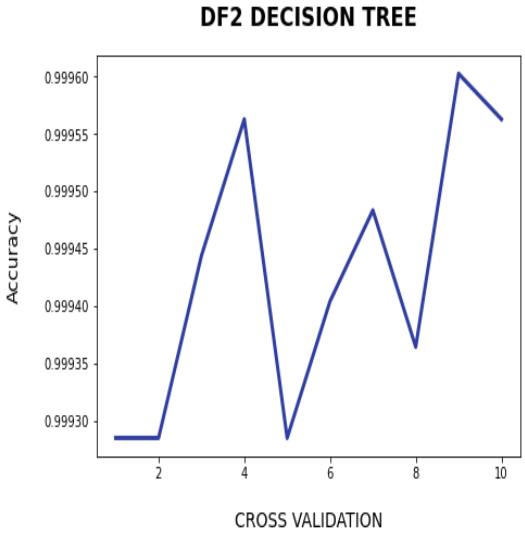
a.MANUAL FEATURE SELECTION RESULTS

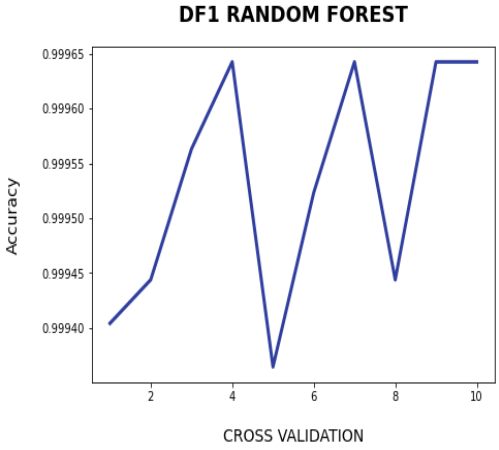
As mentioned above, with different correlation saturation level, we got four different dataset (df1,df2,df3,df4) in manual process. Logistic Regression, Decision Tree and Random Forest models have been built for these dataset with 10-fold Cross Validation and results are as follows:

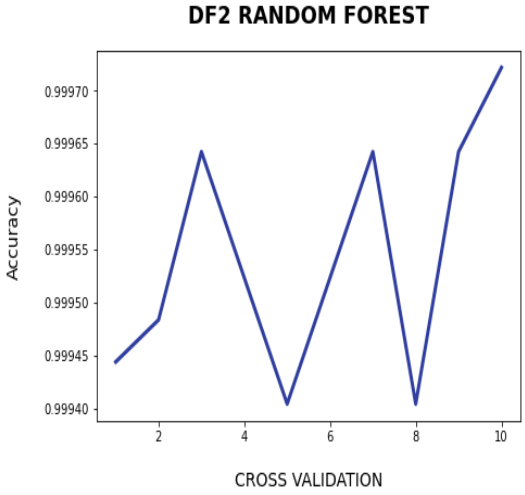
1. DF1 - With Correlation Value 0.95





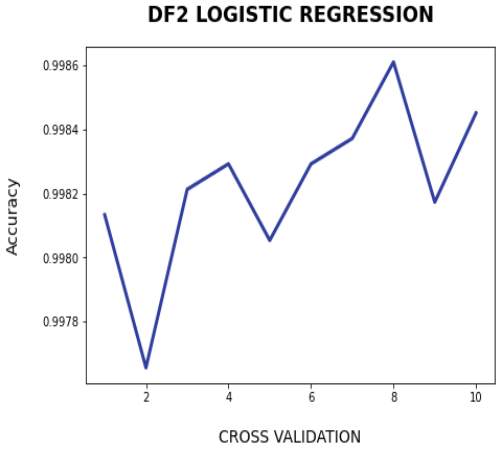


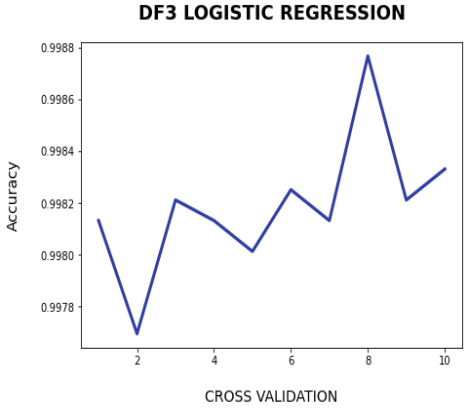


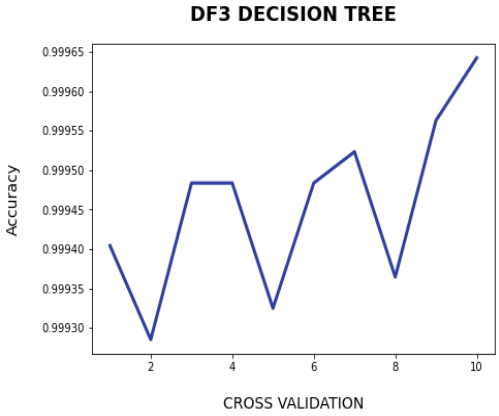


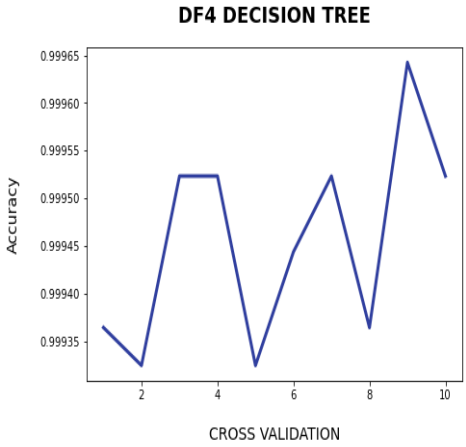
2. DF2 - With Correlation Value 0.90

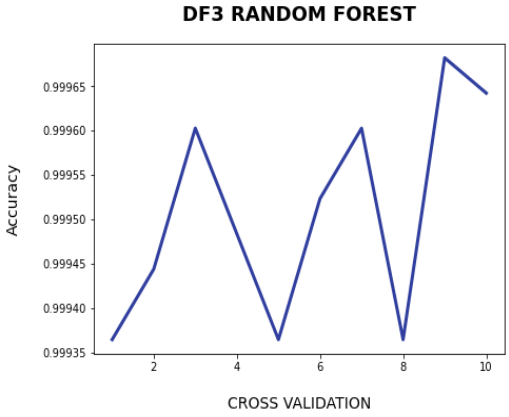
3. DF3 - With Correlation Value 0.85

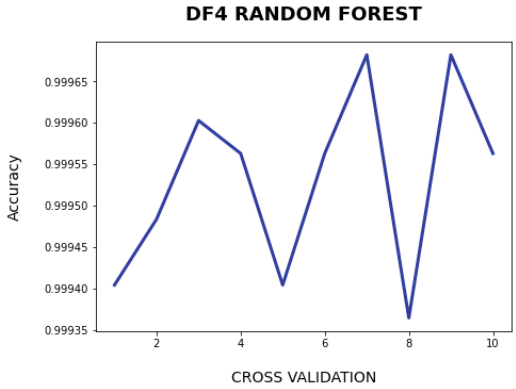




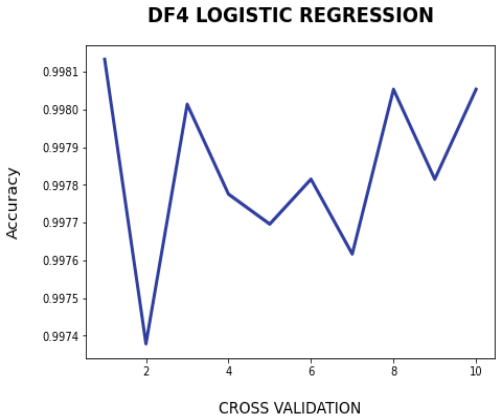








4. DF4 - With Correlation Value 0.80



MEAN ACCURACY OF ALL MODELS

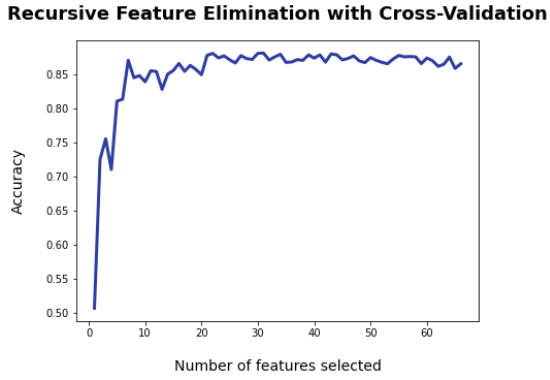
manual_analysis_output

	DATFRAME	LOGISTIC REGRESSION	DECISION TREE	RANDOM FOREST
0	DF1 (0.95 CORR)	0.998554	0.999400	0.999531
1	DF2 (0.90 CORR)	0.998224	0.999428	0.999543
2	DF3 (0.85 CORR)	0.998189	0.999456	0.999507
3	DF4 (0.80 CORR)	0.998554	0.999456	0.999531

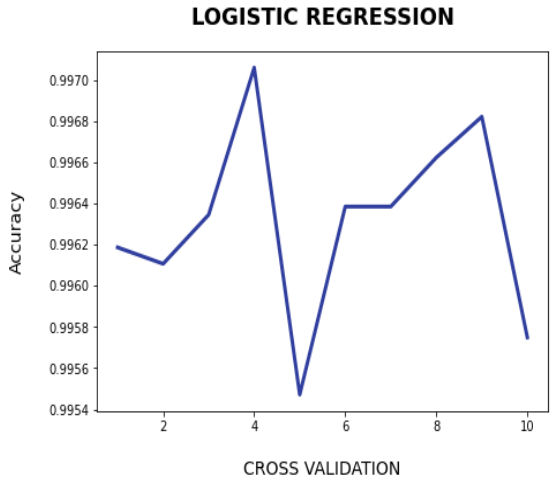
We see that the Random forest model performs well with the correlation value of 0.90 and 0.85 dataset. The DF2 Random Forest Model could overfit the dataset. So lets build the final Random Forest model using DF3 dataset with train test split and validate the results.

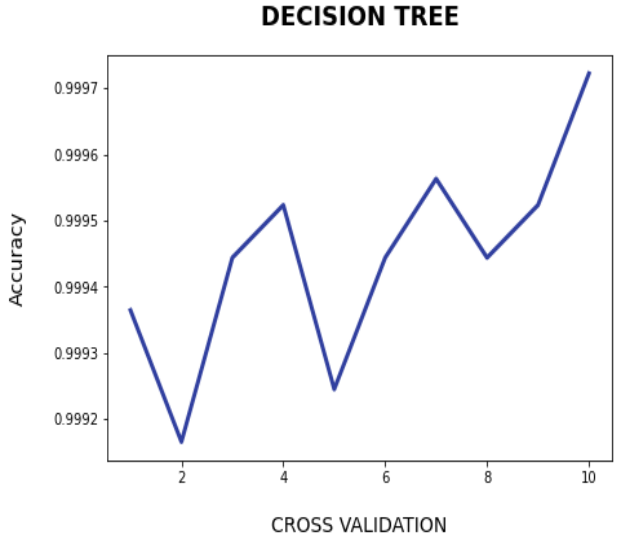
B.RECURRENT FEATURE ELIMINATION

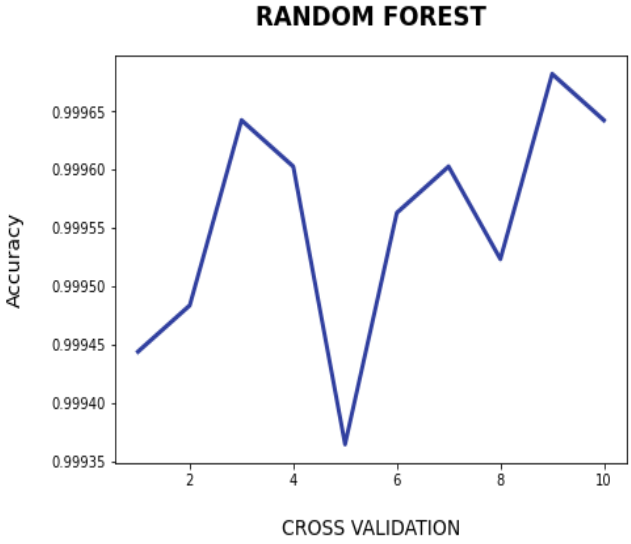
Here Recurrent feature elimination from sklearn.feature_selection is used and the redundant features as been removed. After this process the final dataset contains 31 valid features. Logistic Regression, Decision Tree and Random Forest models have been built for these dataset with 10-fold Cross Validation and results are as follows:



We may clearly see that after a certain number of feature selection our accuracy haven't grown up. Hence we select the optimal number of features throught this process.







LOGISTIC REGRESSION - RFE TESTING RESULT

CONFUSION MATRIX

Predicted/ Actual	BENIGN	DoS Hulk	DoS GoldenEye	DoS slowloris	DoS Slowhttptest	Heartbleed

BENIGN	0	0	1	0	1	0
DoS Hulk	0	1958	17	1	23	0
DoS GoldenEye	0	3	46074	1	33	0
DoS slowloris	0	2	0	1026	77	0
DoS Slowhttptest	0	1	2	19	1107	0
Heartbleed	0	0	0	0	0	1

CLASSIFICATION REPORT

	precision	**recall**	**f1-score**	**support**
BENIGN	0.00	0.00	0.00	2
DoS Hulk	1.00	0.98	0.99	1999
DoS GoldenEye	1.00	1.00	1.00	46111
DoS slowloris	0.98	0.93	0.95	1105
DoS Slowhttptest	0.89	0.98	0.93	1129
Heartbleed	1.00	1.00	1.00	1

ACCURACY SCORE :
0.9964049496494329

DECISION TREE - RFE TESTING RESULT

CONFUSION MATRIX

Predic	**BE**	**Do**	**DoS**	**DoS**	**DoS**	**Hea**

ted/ Actual	**NIGN**	**SHulk**	**GoldenEye**	**slowloris**	**Slowhttptest**	**rtbleed**
BENIGN	1	0	0	1	0	0
DoS Hulk	0	1992	6	1	0	0
DoS GoldenEye	0	5	46106	0	0	0
DoS slowloris	0	2	0	1095	8	0
DoS Slowhttptest	0	0	0	5	1124	0
Heartbleed	0	1	0	0	0	0

CLASSIFICATION REPORT

	precision	**recall**	**f1-score**	**support**
BENIGN	1.00	0.50	0.67	2
DoS Hulk	1.00	1.00	1.00	1999
DoS GoldenEye	1.00	1.00	1.00	46111
DoS slowloris	0.99	0.99	0.99	1105
DoS Slowhttptest	0.99	1.00	0.99	1129
Heartbleed	0.00	0.00	0.00	1

ACCURACY SCORE :
0.999423997457644

RANDOM FOREST - RFE TESTING RESULT

CONFUSION MATRIX

Predicted/ Actual	BENIGN	DoS Hulk	DoS GoldenEye	DoS slowloris	DoS Slowhttptest	Heartbleed
BENIGN	0	0	2	0	0	0
DoS Hulk	0	1577	421	1	0	0
DoS GoldenEye	0	0	46111	0	0	0
DoS slowloris	0	1	290	797	17	0
DoS Slowhttptest	0	0	224	2	903	0
Heartbleed	0	1	0	0	0	0

CLASSIFICATION REPORT

	precision	recall	f1-score	support
BENIGN	0.00	0.00	0.00	2
DoS Hulk	1.00	0.79	0.88	1999
DoS GoldenEye	0.98	1.00	0.99	46111
DoS slowloris	1.00	0.72	0.84	1105
DoS Slowhttptest	0.99	1.00	0.99	1129
Heartbleed	0.00	0.00	0.00	1

ACCURACY SCORE :
0.9809521917889844

FINAL ACCURACY:

```
final_score
```

	Model	Accuracy
0	Logistic Regression	0.996405
1	Decision Tree	0.999424
2	Random Forest	0.980952

We found that decision tree outperformed all other models with the highest accuracy rate of 0.999424.

VI. CONCLUSION

In this paper, the significance of using a set of relevant features with an adequate classification learning algorithm for modelling an IDS has been demonstrated. A presentation and proposition of a feature selection method which consist of a univariate features selection and a recursive feature elimination using a different classifier to identify important features have been done. This process repeatedly builds a model placing the feature aside and then repeating the process with the remaining features until all features present in the dataset are exhausted. The evaluation the effectiveness of the method using different classification metric measurement has been made and it has been proved that by reducing the number of feature, the accuracy of the model was improved. The feature selection method proposed in this paper had achieved a high result in term of accuracy and features were identified based on information gain and ranking technique.

REFERENCES

[1] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, “Using feature selection for intrusion detection system,” 2012 Int. Symp. Commun. Inf. Technol., pp. 296–301, 2012.

[2] M. P. K. Shelke, M. S. Sontakke, and A. D. Gawande, “Intrusion Detection System for Cloud Computing,” Int. J. Sci. Technol. Res., vol. 1, no. 4, pp. 67–71, 2012.

[3] S. Suthaharan and T. Panchagnula, “Relevance feature selection with data cleaning for intrusion detection system,” 2012 Proc. IEEE Southeastcon, pp. 1–6, 2012.

[4] S. Suthaharan and K. Vinnakota, “An approach for automatic selection of relevance features in intrusion detection systems,” in Proc. of the 2011 International Conference on Security and Management (SAM 11), pp. 215-219, July 18-21, 2011, Las Vegas, Nevada, USA.

[5] L. Han, "Using a Dynamic K-means Algorithm to Detect Anomaly Activities," 2011, pp. 1049-1052.

[6] R. Kohavi, et al., "KDD-Cup 2000 organizers report: peeling the onion," ACM SIGKDD Explorations Newsletter, vol. 2, pp. 86-93, 2000.

[7] I. Levin, "KDD-99 Classifier Learning Contest: LLSoft s Results Overview," SIGKDD explorations, vol. 1, pp. 67-75, 2000.

[8] J. McHugh, “Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory,” ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000.

[9] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[10] KDD 99 dataset, Accessed December 2015,http://kdd.ics.uci.edu/databases/kddcup99

[11] NSL KDD dataset, Accessed December2015,https://github.com/defcom17/NSL_KDD

[12] P. Ghosh, C. Debnath, and D. Metia, “An Efficient Hybrid Multilevel Intrusion Detection System in Cloud Environment,” IOSR J. Comput. Eng., vol. 16, no. 4, pp. 16–26, 2014.

[13] Dhanabal, L., Dr. S.P. Shantharajah, "A Study on NSL_KDD Daaset for Intrusion Detection System Based on Classification Algorithms," International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, issue 6, pp. 446-452, June 2015

[14] C. F. Tsai, et al., "Intrusion detection by machine learning: A review," Expert Systems with Applications, vol. 36, pp. 11994-12000, 2009.

[15] V. Bolón-Canedo, et al., "Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset," Expert Systems with Applications, vol. 38, pp. 5947-5957, 2011.

[16] F. Amiri, et al., "Improved feature selection for intrusion detection system," Journal of Network and Computer Applications, 2011.