EAST WEST UNIVERSITY

CSE303: Statistics for Data Science
[Spring 2023]

Project Report

Course Code : CSE303
Course Title : Statistics for Data Science
Section : 2
Group Number : 11

Submitted by:

Student ID	Student Name	Contribution Percentage
2020-1-60-127	Abdullah al Tamim	33.33%
2020-1-60-155	Md. Adnan Morshed	33.33%
2020-1-60-115	Fatema Akter	33.33%

1. Introduction

The Domain Name System (DNS) attack classification dataset is a cybersecurity dataset that includes network traffic information about DNS attacks. The goal of this dataset is to classify DNS traffic as either benign or malicious. Then the DNS traffics were further classified into 4 categories as light attack, light benign , heavy attack and heavy benign. There were two types of features in the data set . Stateless features and Stateful features . Stateful features require information about previous packets or queries to analyze the data effectively. It needs analyzing multiple packets over a period of time. On the other hand, Stateless features require only the current packet or query to be analyzed. It does not require any information about the previous packets or queries . We have worked on the Stateless features . We observed how the stateless features are giving proper information about the DNS attack, and noticed the impact of it.

There are 757211 rows and 17 columns in the dataset. The columns are described below:

1. 'Timestamp' : The time when the data was collected.
2. 'FQDN count' : Total count of characters in FQDN(fully qualified domain name).
3. 'Subdomain length' : Count of characters in subdomain.
4. 'Upper' : Count of uppercase characters.
5. 'Lower ' : Count of lowercase characters.
6. 'Numeric' : Count of numerical characters.
7. 'Entropy ': Entropy of query name : H(X) =

$$-\sum_{k=1}^{N} p(x_k) log2 p(x_k)$$

   X= query name, N= total number of unique characters, p(xk)= the probability of the k-th symbol

8. 'Special' : Number of special characters; special characters such as dash, underscore, equal sign, space, tab.
9. 'Labels': Number of labels; e.g., in the query name "www.scholar.google.com", there are four labels separated by dots.
10. 'Labels_max' : Maximum label length.

11. ‘’Labels_average’ : Average label length.
12. ‘Longesr_word’ : Longest meaningful word over domain length average.
13. ‘Sld’ : Second level domain.
14. ‘Len’ : Length of domain and subdomain.
15. ‘Subdomain’ : Whether the domain has a subdomain or not.

We have added two new columns these are :

16. ‘Attack ’ : This column was added to classify attack and benign. Here 0 means that the traffic was benign and 1 means that the traffic was malicious.
17. ‘Origin’ : This column was added to further classify the attack type. There are a total four classes of this column, those are :- heavy_attack, heavy_benign, light_attack, light_benign.

2. Exploratory Data Analysis

From the discrete numerical data distribution, we can see that half the features have a data distribution with a bell shape but have a high mode and some outliers as well . The other half of the distribution does not have a bell shaped distribution.

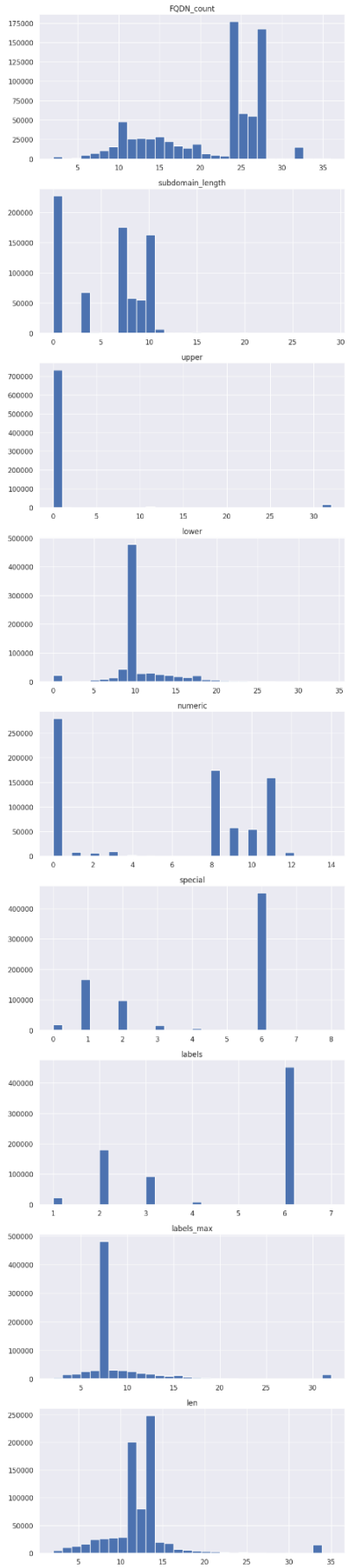

Fig -1: Discrete numeric data distribution

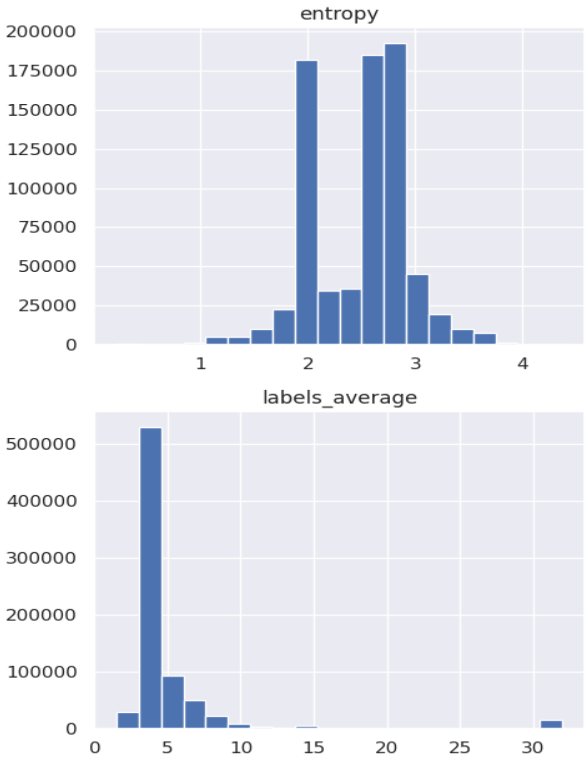

Fig -2: Continuous numeric data distribution

From the continuous numerical data distribution, we can see that the features of the data distribution are with a bell shaped, but has a high mode .

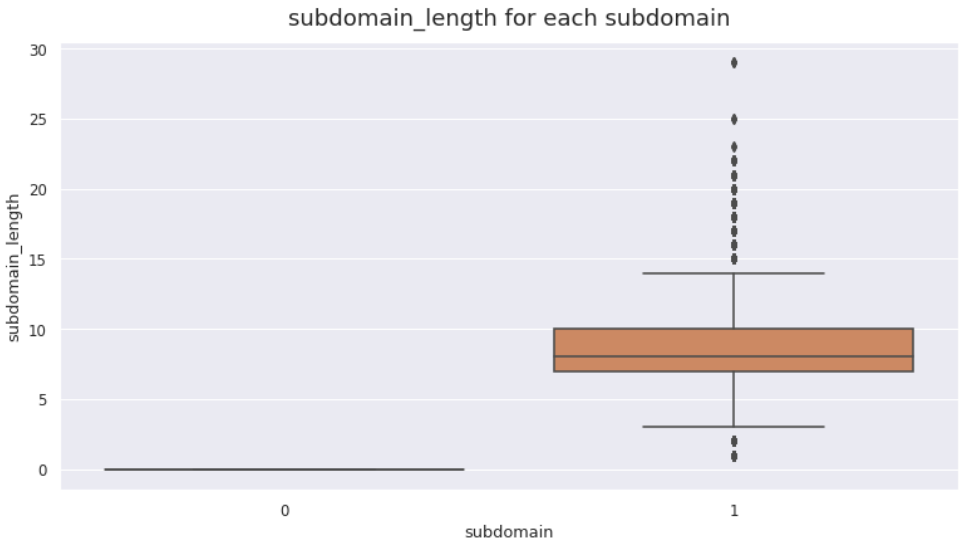

Fig -3: Box plot of the subdomain length for each subdomain

In the above boxplot, we can see that, where there is subdomain there are many outliers in the subdomain length. Subdomain 0 means there is no subdomain , as a result there is no subdomain length.

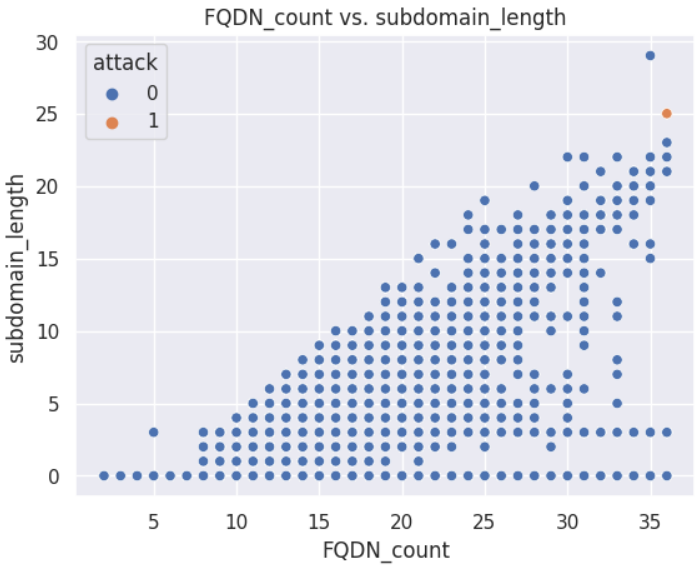

Fig -4: Scatter plot of FQDN_count vs subdomain_length

In the above scatterplot, we can see that there is a linear relationship between FQDN count and subdomain length. It means that when FDQN_count is increasing the subdomain_length is also increasing. Most of the data is benign. In the higher count of FDQN_count we can see the attack class.

Fig -5 Word cloud for longest word

In the longest cloud feature, the most appeared word as 2 and 4 respectively.

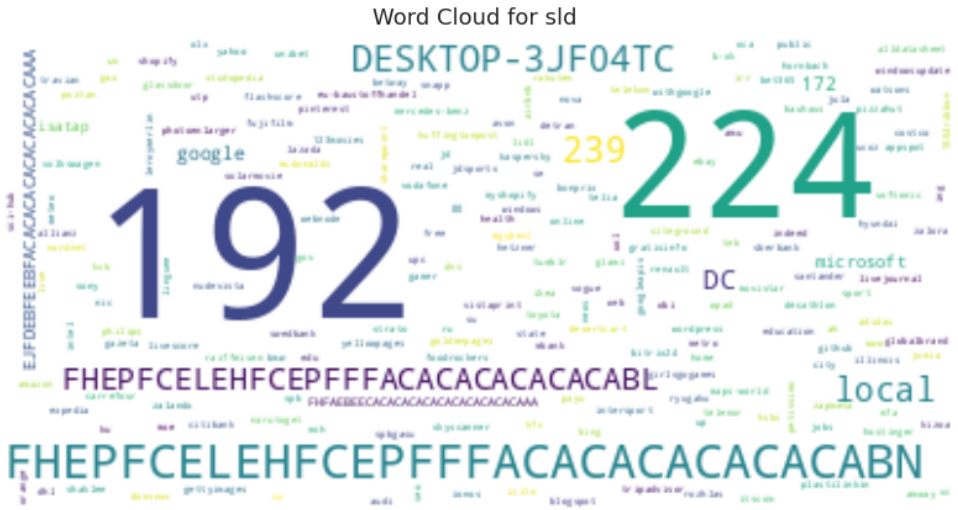

Fig -6 Word cloud for sld

In the sld feature, the most appeared word was 192 .

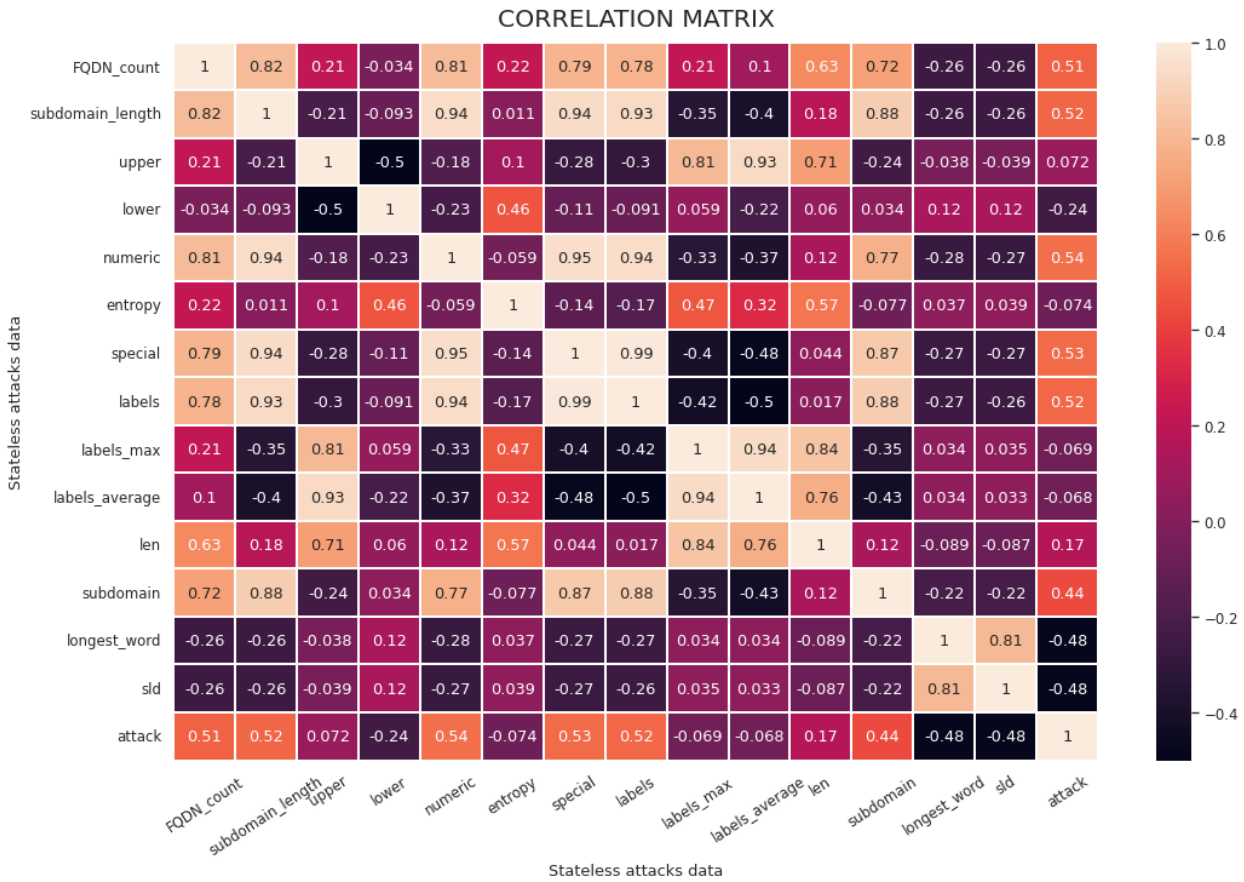

Fig -6 Correlation matrix of Stateless attacks data

From the above correlation matrix we can see that ‘FQDN_count’, ‘subdomain_length’, ‘numeric’, ‘special_labels’ are correlated with the target variable ‘attack’ , and ‘longest word’ and ‘sld’ is negatively correlated with ‘attack ’. Still the score is not that high. Rest of the features are not linearly correlated at all . We can say that there is hardly any strong correlation between the features and the target variable.

3. Data Preprocessing

The preprocessing steps include handling missing values, encoding categorical variables, and scaling numerical variables.

- We only had missing values in one column which is the longest_word. We have filled the missing values for this column using the empty strings.
- We dropped the duplicate values as these were too small (526 instances) compared to our dataset (757211 instances).
- We dropped the timestamp column as it was representing the time when the packets were received and hence unique.
- We have encoded categorical variables like sld and longest_word using ordinal encoding.
- Finally as part of preprocessing, we have scaled all the numeric columns using Standard Scaler.

4. Machine Learning Models

We have used Random Forest and Logistic Regression in our project.

Random forest:

Random Forest is a machine learning algorithm that is used for classification and regression. It works by creating a forest of decision trees, where each decision tree is trained on a subset of the data and some random features. During training, the algorithm creates multiple decision trees, each with its own set of rules for making predictions. Then the final prediction is given

combining the decisions of all the trees. Random Forest algorithm has the ability to handle large amounts of data with high accuracy. It can also handle missing data and outliers well, and it is generally robust to overfitting.

Some of the parameters of this model is discussed below:

- n_estimators: the number of trees in the forest.
- max_depth: the maximum depth of each decision tree.
- random_state: the random seed used for random number generation during the fitting process. Setting a random seed ensures that the same results are obtained each time the code is run.

We did 5 fold cross validation of our random forest model to evaluate the performance of the model on a limited sample size.

Logistic regression:

Logistic Regression is a machine learning method used for binary classification tasks normally. It models the relationship between the dependent variable and one or more independent variables by estimating the probability using a function called Sigmoid function or Logistic function. The algorithm works by fitting a logistic curve to the data, which is derived from the Sigmoid function and it transforms any real-valued input to a value between 0 and 1.

During training, the logistic regression model learns the optimal coefficients for the input variables, which allow it to predict the probability of the dependent variable taking a certain value. The model can then make binary classifications by setting a threshold value above or below which the output variable is assigned one of the two possible values.

Some of the parameters of logistic regression is discussed below:

- penalty: it means the type of regularization to be applied to the model. The default value is ‘l2’ which uses LASSO regression. Other options include ‘l1’ which uses RIDGE regression and ‘elasticnet’ is used to balance between l1 and l2.

- C: It is the inverse of regularization strength. Smaller values of C indicate stronger regularization. The default value is 1.0.

5. Performance Evaluation and Discussion

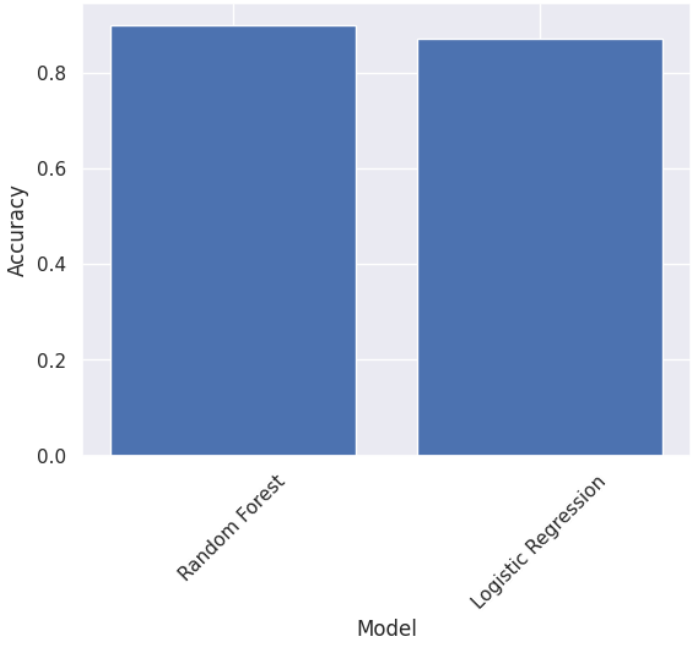

Fig -7 Accuracy Comparison of Models

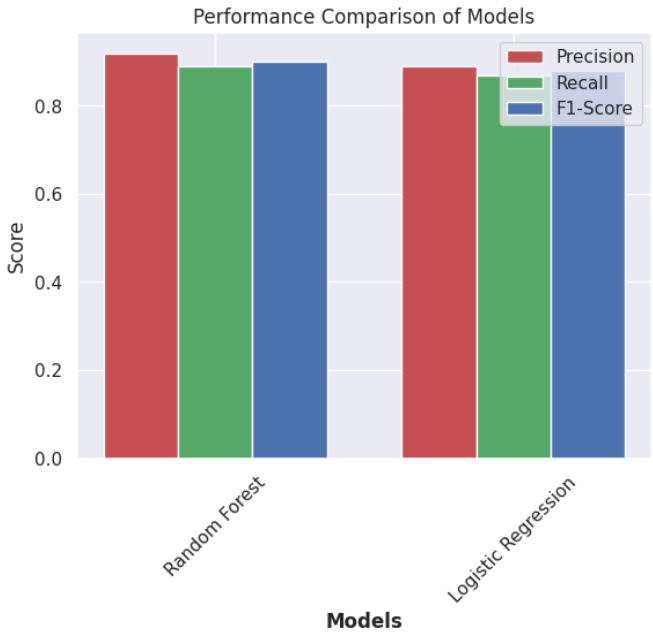

Fig -8 Performance Comparison of Models

From the bar charts of the two models, we can see that Random Forest is doing better than Logistic Regression . The Random Forest model has a higher accuracy, precision, recall and f1-score for both classes 0 (no-attack) and 1(attack) which was divided into more subclasses like light attack, light benign , heavy attack and heavy benign.

The potential reasons why Random Forest performed better than Logistic Regression have given below:

1. Random Forest is a non-linear machine learning model which can capture the non-linear relationship between the features and the target label. On the other hand, Logistic Regression is a linear machine learning model which matches the linear relationship between the features and target column. From the correlation table of our dataset, we saw that the relation among the features were non linear in most of the cases . As there was a little linear correlation between the features, Random Forest did well in terms of classify correctly than Logistic Regression .
2. Random Forest is an ensemble machine learning model that creates multiple trees on randomly selected subsets of the features and the rows. It helps the models to learn the importance of the individual features in order to classify them. On the other hand,

Logistic Regression uses all the features to predict the class. There might be a possibility that some of the features are not that important to predict the class and are adding noise to the Logistic Regression model.

3. From the boxplots shown, we can see that many of our numeric columns have many outliers. Random forests can handle the outliers by using decision trees with a limited depth. On the other hand, Logistic Regression needs all the data to be available and can be sensitive to outliers.

These are the potential reasons why Random Forest performed better than Logistic Regression in our dataset. In both of the models we can see that our scores are almost generalization error free. As Logistic Regression has a lower score it might have a higher bias and less variance. In contrast, the score of Random Forest is much better, so there is a possibility of less bias and higher variance. In the 5-fold cross validation of Random Forest the performances were almost similar. So we can conclude that Random Forest was generalized and performed well.