Problem Statement

This is our problem statement. our problem statement is basically the prediction of top level annotation for binary classification of Malware for the given NetML dataset. Usually Network traffic analysis used approaches but because of advancement in Science and application development newer application uses dynamic port allocation instead of using standard registered port Numbers. So, we are employing ML methods with flow statistical features, since they do not rely on port Numbers or payload itself.

Dataset Description

our training dataset consists of 3,87,268 rows and 62 columns and test dataset has 48,394 rows and 62 columns.

Here this 62 columns are the summation of 4 set of features- metadata, TLS, DNS & HTTP. TLS, DNS and HTTP are dependent features while metadata features are protocol independent. Metadata features have 32 columns, rest other 30 colomns belongs to protocol based set of features that is TLS, DNS and HTTP.

Each of the rows in dataset identified by a unique_id whose output label, contains binary categorical value that is malware and benign which are stored in seperate output files. Here we have 387268 unique ids same as the number of rows. We are going to predicting whether the output is malware & benign.

Exploratory Data Analysis

Slide-1

Here Fig:3.1 shows the Heat map or the correlation matrix. Basically we used this to find & identify the strong correlation between input colums that is within different features. Since our domain information require expertise in this field so considering the situation heat map was the most apt choice.

Slide-2

Here Fig 3.2 shows the scatter plot between reverse payload variance and maximum reverse payload value withe the correlation value of 0.97. Inspite of the high correlation between these features, there is no logical correlation.

Comming to Fig 3.3, here the scatter plot is between numbers of bytes out from the host machine and number of packets whose payload length are above 1024 with the correlation value of 0.94.

P.no:)

This implies that the host machine might be infected by malware since it is sending payload length of more than 1024 as well as data which is represented here in terms of bytes out is being send from host to attacker or a server.

Now comming to Fig 3.3, it shows the scatter plot between no of packets between header length 28840 and reverse no of packets between header length 28840. with the correlation value of 0.9286. This implies that there is a connection & reverse connection between host and attacker machine, potentially indicating malware.

Slide-3

In Fig 3.5 the graph shown is between Malware & benign count, from the graph we can infer that the malware count is higher as compared to benign count present in the Data set.

Comming to Fig 3.6 Here the graph shown is between UDP count versus TCP count, from the graph we can infer that UDP count is greater than TCP.

Now comming to Fig 3.7 here the graph shown is between the output predicting class of malware/benign versus the protocol attribute. The malware frequency is represented on y-axis and malware and benign class

are represented as 1 and 0 on the x-axis. From the graph we can infer that most of the Network traffic flow are using UDP connection and they maybe using applications which uses UDP connection and since UDP connection is not the secure one therefore there is a lot of risk & chance of malware attacks.

→ Since our problem statement involves binary classification we are restricted to only predicting a class to be malware or benign & not into further malware classification.

Data preprocessing

Slide-1

Note → Refer previous Pdf.

S-6 Data Preprocessing

→ These are the steps followed for the Data preprocessing of our NetML dataset.

1. Starting from Data cleaning. So on analyzing the dataset we found out that there were no duplicate tuples apart from that all the relevant attribute or features had no missing values. Here Non-relevant attribute means NaN values from Protocol dependent features.

2. on analyzing outliers on the NetML training dataset we found out that 3 of the features namely reverse payload max, reverse payload distinct and number of packets out have outlier in them, also we decided not to remove these outliers because we found out that after after removing all the outlier tuples from this 3 features there is significant amount of reduction in the dataset that is from 387,268 such tuples we are left with only 2,10,914 which may significantly impact our learning model.

2. → Now Comming to data integration. Since we have 2 datasets 1 the input NetML training dataset and other training-annotation-top data set. we merge them both on the basis of unique ids present in both the datasets (common. Thereby having input feature & output target label in same dataframe.

→ Also there were 8 attributes which were of object tuples & needed to be broken to individual elements and to be merged with existing dataset so that it can be fed to our ML model leading to increase in the columns from 32 to 122.

3. → Now comming to Data reduction. Here we removed redundant features like the TLS, DNS & HTTP features which are protocol based features since almost 80-85% flow features for these are Not available for all of the flows in the dataset, therefore we only use metadata features to use baseline results.

4. → Now comming to Data Transformation.
Here the output target label is a binary class namely benign & malware which are categorical so we encoded malware class as 1 & benign class as 0. Similarly protocol features has two categorical value UDP & TCP. we encoded TCP as 17 and UDP as 6.

Note → Refer previous Pdf.

Slide-2

These are the graphs which were created to Check whether the ~~these~~ features whose Correlation Value was above 0.9 & were selected as we have discussed in the previous slides of EDA, has outliers in them or Not.

→ For explaining Figures use report.

P.no:

S-7 Model building

→ After doing all the essential preprocessing steps we went for building our ML model since our dataset that is NetML dataset had 2 level of annotation – toplevel & finedlevel. And top level annotation was assigned to us, which was nothing but a binary classification problem so we ran 3 classification models – logistic regression, KNN classification and Random forest.

→ Following is our model comparison you can see in the bar graph below which clearly indicates that RF performs the best with the accuracy score of 99.82.

→ After that we evaluated our model by drawing ROC curve and confusion matrix.

ML model

Before pdf (previous)

Bias Variance tradeoff

As we know bias is the difference between average prediction of our model and the correct value which we are trying to predict. Higher bias leads to two things First very less attention to the training data, second over simplifies the model leading to high error on training & test data.

Variance is the variability of the model prediction for a given data point or a value which tells us spread of our data.

So high variance means paying a lot of attention to training data and does not generalize on data which he hasn't seen before.

Fig 5.2 shows bias-variance trade off graph as well as the value shown in the left portion, it clearly shows less bias and also less variance of our best choosen mode that is RF (classifier model).

Post model analysis

For Post model analysis of our RF model 2 metrics were used ① ROC curve & ② confusion matrix.

Fig 6.1 shows the Feature importance of different input features used in our optimal model that is RF classifier model. The graph here shows them in decreasing order of their importance.

S-8

→ Comming To ROC Curve on the bottom Right side we can see that our training model after hyperparameter tuning ~~per~~ was a near perfect model with an AUC matric score of 1

↳ (Area under ROC curve)

* AUC measure the entire 2-dimensional area under the ROC curve. so Here AUC provides an aggregate measure of performance across all possible classification threshold.

We use AUC because of 2 reasons

(i) AUC is scale invariant, so it measures how well predictions are ranked rather than their absolute values also the second thing is AUC is classification threshold invariant that is it measures the quality of model prediction irrespective of what classification threshold is choosen.

Fig 6.3 shows confusion matrix of RF model before hyperparametric tuning. Here the wrong predictions count that is 87 and 120. 87 being benign class predicted as malware class whereas 120 being malware class predicted as benign class and 15,000 and 62,000 are the correct predictions of malware and benign class.

Fig 6.4 shows confusion matrix of RF model after hyperparametric tuning. After hypertuning with the best parameters we got the number of False predicted values dropped increasing our accuracy. So FP value reduced to 64 & FN values reduced to 102 thereby increasing our model accuracy by 0.01%.