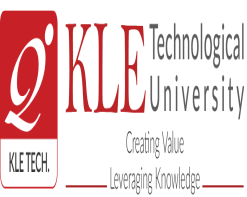
KLE Society's

KLE Technological University



**Data Mining and Analysis (18ECSC301)**

**Course Project Report**

**On**

**Network Traffic Analysis**

**Submitted By**

|  |  |
| --- | --- |
| Aayush Rajwade | 01fe18bcs005 |
| Aman Kumar | 01fe18bcs029 |
| Annapoorna Pattar | 01fe18bcs040 |
| Ayush Utsav | 01fe18bcs059 |

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

HUBLI–580 031 (India).

Academic year 2020-21

**CONTENTS**

1. Introduction
   1. Motivation……………………………………………………………………3
2. Problem Statement……………………………………………………………….4
3. Literature Survey…………………………………………………………………5
4. Methodology
   1. Data Preprocessing…………………………………………………………..6
   2. ML Models…………………………………………………………………..7
5. Result
   1. Dataset Description………………………………………………………….8
   2. Performance Evaluation Parameters
      1. Bias-Variance trade-off……………………………………………...9
      2. Confusion Matrix…………………………………………………...10
      3. ROC Curve………………………………………………………….11

**5.3** Exploratory Data Analysis………………………………………………….12

**6.** Conclusion………………………………………………………………………..16

**7.** References

1. **INTRODUCTION**

**1.1 Motivation**

Classifying network traffic is the basis for important network applications. Prior research in this area has faced challenges on the availability of representative datasets, and many of the results cannot be readily reproduced. Such a problem is exacerbated by emerging data-driven machine learning based approaches.

Network traffic classification is vital for many applications such as QOS (Quality of service control), resource allocation and the most important malware detection particularly from security point of view and increasing cyber attacks . The first and easiest port-based approach has become obsolete since newer applications mostly use dynamic port allocation instead of using standard registered port numbers. These days researchers are employing learning methods with flow statistical features since they do not rely on port number and payload itself.

There have been a plethora of research attempting to classify and analyze network flows using a variety of datasets. However, unlike the open datasets such as ImageNet and COCO in computer vision research, it is very difficult to find comprehensive datasets for researchers in networking domain to evaluate their proposed techniques, and for the community to reproduce prior art. NetMl Challenge is one such step forward in terms of providing comprehensive dataset for research and analysis purposes.

1. **PROBLEM STATEMENT**

**“**For the given NetML dataset, prediction of top-level annotation for binary classification of malware.**”**

According to the problem statement the expectation is to predict the top-level annotation for binary classification of malware i.e. Malware or Benign class for the given NetML dataset. Usually network traffic analysis uses port-based approaches but because of advancement in science and application development newer application uses dynamic port allocation instead of using standard registered port numbers. So in this scenario ML methods with flow statistical features may be employed, since they do not rely on port numbers or payload itself.

1. **Literature Survey**

There are four set of features :(1) Metadata,(2) TLS,(3) DNS, (4) HTTP .Metadata features are protocol independent features such as number of packets, bytes inbound and bytes outbound, time length . On the other side, there are protocol-specific features such as TLS, DNS and HTTP. Number of ciphersuites and extensions supported by the client or server are a subset of the TLS features. Similarly, DNS query name and DNS answer IP can be given as example for DNS features.Finally, HTTP code and HTTP method are two examples for HTTP features. While Metadata features can be extracted for any kind of flow, protocol-specific features can only be extracted if the given flow contains packets with one of these protocols . [1]

The baseline classifiers require the input data as a two dimensional array whose columns represent features and rows stand for flow samples. Therefore, we need to convert the given data into matrix format. Firstly, we discard source and destination IP address features as they are already masked. All other Metadata features in the datasets contain numeric values and hence it is straight-forward to import these datasets into a matrix form that will be used to train the classifiers. For array-like features such as hdr\_ccnt[], we just treat each of those dimensions in this array as a separate feature such as *hdr\_ccnt\_0*, *hdr\_ccnt\_1* etc. up to *hdr\_ccnt\_k* for k+1 dimensional hdr\_ccnt[] feature and place each of those into seperate columns in the final data matrix. Before training, the data is standardized in the preprocessing step so that data in each feature column follows standard normal distribution .[1]

1. **Methodology**

**4.1 Data Preprocessing**

**4.1.1 Data Cleaning**

**Missing Values:** The NetML dataset had 4 types of flow features that is categorized into 2 main features metadata features and protocol-based features. So protocol-based features had 80-90 % missing values which was significantly higher so these features were dropped.

**4.1.2 Data Integration**

Merging of the NetML dataset with the training\_annotation\_top dataset increasing the column size to 32. Breaking the tuples that were in the form of object tuples into individual elements and merging with existing dataset and feeding the dataset to the ML model. The merging results in increase in the column from 32 to 122.

**4.1.3 Data Reduction**

Removal of all the redundant features like protocol dependent set of features (TCP, DNS and HTTP).

**4.1.4 Data Transformation**

Encoding the Protocol feature UDP as ‘6’ and TCP as ‘17’. Encoding Malware as ‘1’ and Benign as ‘0’.

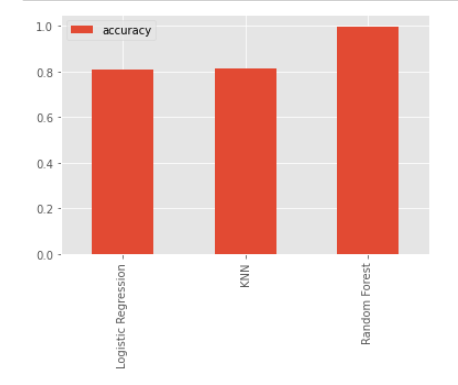
**4.2 ML MODELS**

The ML model used for Binary Classification of the NetML dataset are as follows:-

1. Random Forest Classifiers

2. Logistic Regression

3. KNN Classification



**Figure 4.1:** Model Comparison

Figure 4.1The comparison between the three used ML models. From the graph it is inferred that Random Forest model gives the highest accuracy among all other used models.

Accuracy scores of the models on validation dataset are shown below:

1. Random Forest Classifiers: 99.77%

2. Logistic Regression: 81.05%

3. KNN Classification: 81.53%

1. **RESULT**

**5.1 Dataset description**

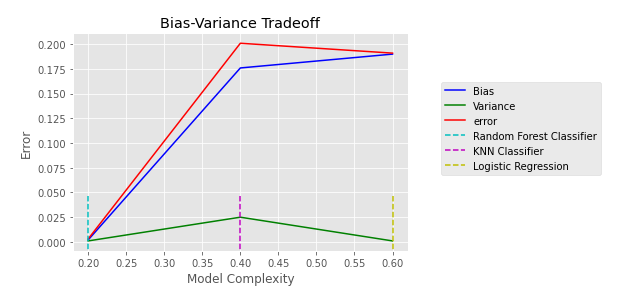
The training dataset consists of 387268 rows and 62 columns. These 62 columns are the summation of 4 set of features namely-metadata, TLS, DNS and HTTP. TLS, DNS and HTTP are protocol dependent features whereas metadata features are protocol independent. Metadata set of features have 32 columns, rest 30 column belongs to protocol-based set of features. Each row is identified by unique id whose o/p label contains binary categorical values: malware and benign, stored in a separate o/p file. We have 387268 unique ids same as the number of rows.

**5.2 Performance Evaluation Parameters**

**5.2.1 Bias-Variance Tradeoff**

**Table 5.1:** Bias variance trade-off values of each model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no** | Model | MSE | Bias | Variance |
| **1.** | Random Forest | 0.003 | 0.002 | 0.001 |
| **2.** | Logistic Regression | 0.201 | 0.176 | 0.025 |
| **3.** | KNN Classification | 0.190 | 0.190 | 0.001 |

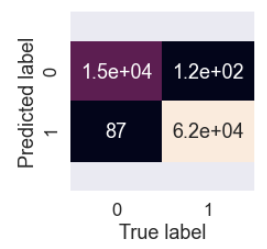


**Figure 5.1:** Bias-Variance Tradeoff comparison.

From Table 5.1 and Figure 5.1 it can be clearly inferred that Random Forest has the least bias as well as the least variance among the three models which suggests that the model that needs to be chosen is Random Forest. So the chosen model needs to be tuned with hyperparameters for better results.

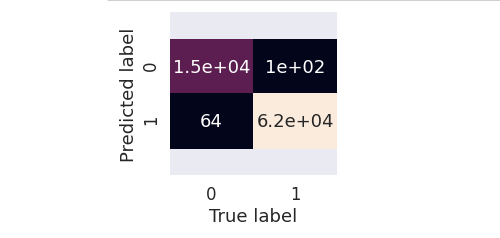
The accuracy score is the highest in RF Classifier model. The Bias-Variance tradeoff clearly showed the least error of 0.3%. The Domain Knowledge required for the feature extraction is difficult in short time so this model is the best for such situations.

**5.2.2 Confusion Matrix**

****

**Figure 5.2**

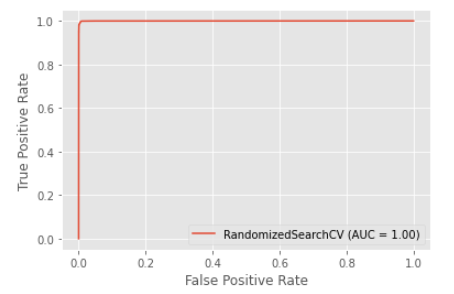
Figure 5.2shows the Confusion Matrix of RF Classifier model with hyperparametric feature of 1000 trees. The Confusion Matrix indicates 207 incorrect predictions whereas 77,000 correct predictions with model accuracy score of 99.7792%.



**Figure 5.3**

Figure 5.3The confusion matrix with hyperparameter feature of 1500 trees. The confusion matrix shows 164 incorrect predictions whereas 77,000 correct predictions with the accuracy score increasing to 99.7856%.

**5.2.3 ROC Curve**

****

**Figure 5.4:** Roc Curve of RF model.

Figure 5.4 shows the ROC (Receiver Operating Characterstics) curve of the tuned RF Classifier model. Since **perfect classifier**is a combination of two straight lines both moving away from the baseline towards the top-left corner. It can be inferred that the tuned model is perfect classifier model.

**Table 5.2**

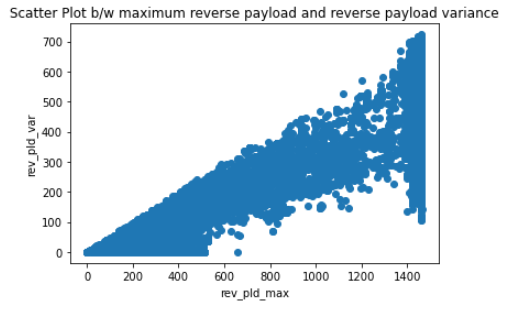
|  |  |  |
| --- | --- | --- |
| TPR | FAR | Overall Accuracy |
| 0.99931 | 0.00590 | 0.99341 |

Table 5.2 shows the final result after prediction file is submitted for evaluation. For the submitted prediction the TPR(True Positive Rate) value was 0.99931, FAR(False Alarm Rate) value was 0.00590 and Overall Accuracy as 0.99341.

**5.3 Exploratory Data Analysis**

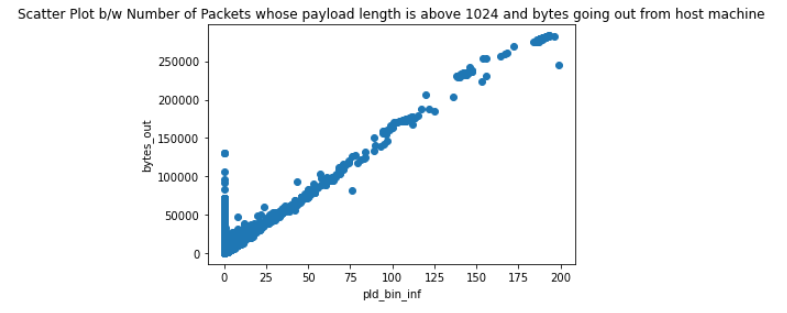
Since the domain knowledge is not that clear so, the best way to find relationship between different input features is through correlation matrix. Heat Map of the correlation matrix gives the clear picture of how on the scale of -1 to 1 different features are correlated.

On the basis of correlation value found from above correlation matrix following are the scatter plots showing relationship b/w different features.

****

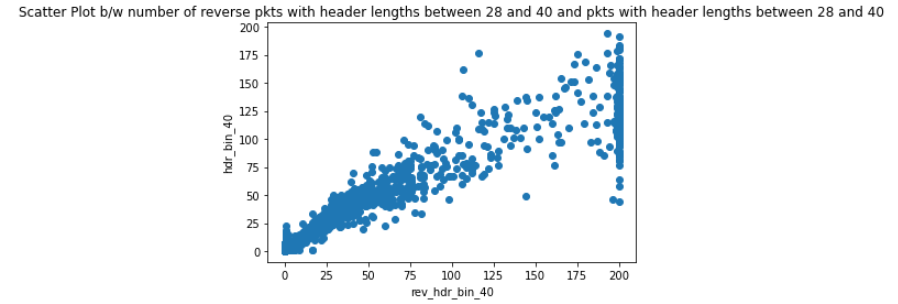
**Figure 5.5:** Scatter plot b/w rev\_pld\_max and rev\_pld\_var.

Figure 5.5 shows the Scatter plot with correlation value of 0.97 between reverse payload variance and maximum reverse payload value. In spite of a large correlation between them there is no logical correlation.



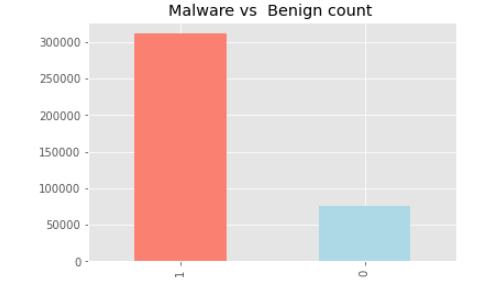
**Figure 5.6:** Scatter plot b/w bytes\_out and pld\_bin\_inf.

Figure 5.6Shows the Scatter plot with the correlation value of 0.94 between number of bytes out from the host machine and number of packets whose payload length are above 1024. 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.



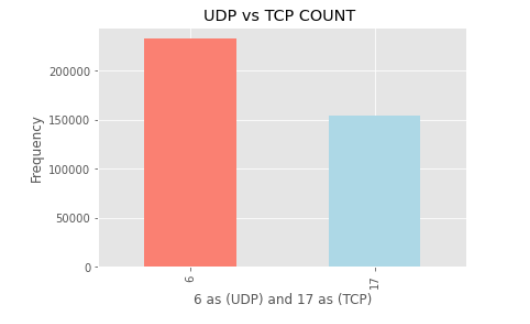
**Figure 5.7:** Scatter plot b/w hdr\_bin\_40 and rev\_hdr\_bin\_40.

Figure 5.7 Scatter plot with the correlation value of 0.9286 between number of packets between header length 28 and 40 and reverse of number of packets between header length 28 and 40. This implies that there is a connection and reverse connection between host and attacker machine, potentially indicating malware.



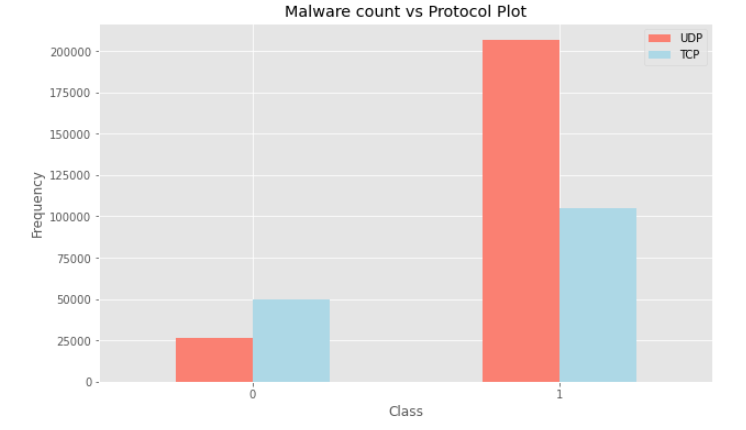
**Figure 5.8:** Bar plot of malware vs benign count.

Figure 5.8The graph between the Malware versus Benign count that shows the Malwares count compared to the Benign count present in the dataset.



**Figure 5.9:** Bar plot of UDP vs TCP count.

Figure 5.9 The graph for UDP count versus TCP count. From the graph it is inferred that UDP count is greater in number as compared to TCP count.



**Figure 5.10:** Malware count vs protocol plot.

Figure 5.10The graph between the output predicting class count of Malware/Benign versus the protocol attribute. The Malware frequency is represented on y-axis and Malware is represented on x-axis as 1 and Benign as 0. From the graph it is inferred that most of the network traffic flow are using UDP connection and they may be using applications which uses UDP connection and since UDP connection is not the secure therefore there is a lot of chance of Malware attacks.

**6. Conclusion**

So we conclude that data driven machine learning techniques are proving to be a boon in Network Flow Analytics. It was found that Random Forest was the best performing model for malware detection tasks. The accuracy of prediction on NetMl test dataset was 99.341%.

We learnt that it is important to promote Machine Learning based Network Traffic Analysis because of better accuracy of predictions. We got deeper insights on how machine learning techniques can be used to solve problems that are most relevant in current scenario.

Implementing [a solution that can continuously monitor network traffic](https://www.rapid7.com/products/insightidr/features/network-traffic-analysis/) gives us the insight we need to optimize network performance, minimize attack surface, enhance security, and improve the management of resources. However, knowing how to monitor network traffic is not enough. It’s important to also consider the data sources for your network monitoring tool; two of the most common are flow data .

**7. Reference`s**

[1] Barut, Onur, et al. "NetML: A Challenge for Network Traffic Analytics." *arXiv preprint arXiv:2004.13006* (2020).

[2] Yiu ,Tony . “Understanding Random Forest: Howthe Algorithm Works and Why it Is So Effective ” . towards data science :Jun 12 , 2019 https:// [Understanding Random Forest. How the Algorithm Works and Why it Is… | by Tony Yiu | Towards Data Science](https://towardsdatascience.com/understanding-random-forest-58381e0602d2) .

[3] Edwards, Gavin. “Macine Learning: an Introduction”. Towards data science Nov 18, 2018 https://[Machine Learning | An Introduction | by Gavin Edwards | Towards Data Science](https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0)

[4] SRIVASTAVA ,TAVISH. “Important Model Evaluation Metrics for Machine Learning Everyone should know”. Analytics Vidhya Aug 6, 2019 https:// [Evaluation Metrics Machine Learning (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/).