Cyber Analytics Visual Analytics Team 1 Project 2 Report

Zicheng Liu
Department of Computer Science
University of Delaware
zcliu@udel.edu

Ezeanaka Kingsley. U
Department of Computer Science
University of Delaware
ekingsly@udel.edu

Wanxin Li
Department of Computer Science
University of Delaware
wanxinli@udel.edu

ABSTRACT

This report recorded an experiment done by graduate students from Cyber Analytics Course lectured by John Cavazos from University of Delaware made an extension to a technique called Class Signature to analyze a dataset of 1100 malware samples to distinguish discriminative features of malwares. The target is to set up a widget consisting of 66 T-SNE map representing 66 sub-cluster of 11 malware families with 347 features with 10 top ranked discriminative features below.

1. INTRODUCTION

This report extended the part 1 of this project. In part 2, we used all the dataset (except for one for testing) Tristan has given to us to train a decision tree model used as a filter for the test dataset. We set a high bound and a low bound as thresholds for good and bad enough predictions of the test dataset. Thus, we filtered the results of moderate prediction of test dataset and kept good and bad enough results of prediction, that way we got a filtered test dataset. After that we used the filtered test dataset as the input of Clustering Module to generate 66 new sub-clusters. Then we applied the pipeline of project 1 on these new sub-clusters. Besides, for each of 11 malware families we also generated corresponding ROC curves and confusion matrices to evaluate the goodness and badness of predictions

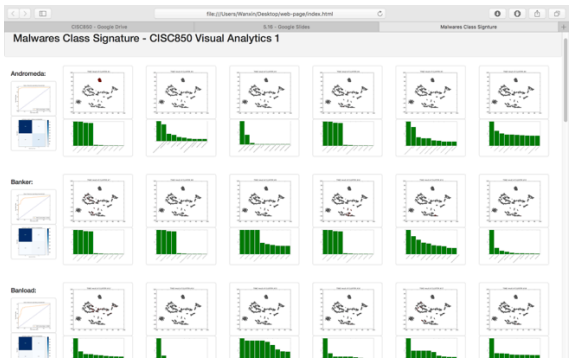

Figure 1 Finished Target Sample

The functionalities we added in project 2 would help us enhance the discriminativeness of top features of each sub-cluster of 11 malware families. We used

Scikit Learn Python API to help create visualizations. Thus, we completed the extension to Class Signature technique and finished our expected target sample shown as fig. 1

1.1 Filtering

1.2 T-SNE

1.3 Feature ranking

1.4 ROC and Confusion Matrix

In statistics, a receiver operation characteristic curve, i.e. ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied[1]

In this project, we have 11 target classes. Since ROC curve is applied in the binary classifier system, we converted 11 classes into 11 different corresponding combinations of binary classes with the specific class being positive. Thus, we generated 11 ROC curve along with corresponding positive class.

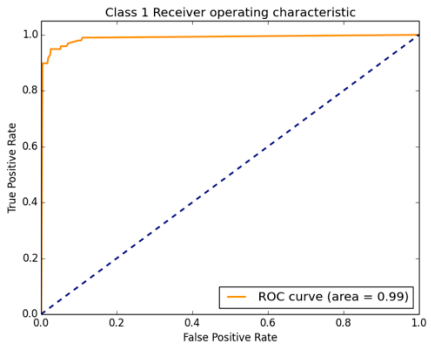

Figure 2 ROC Curve of Class 1

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another) [2]

In this project, we used scikit learn's confusion matrix API to generate 11 confusion matrices for each of combinations of binary classes. The degree of color's darkness represents the number of samples.

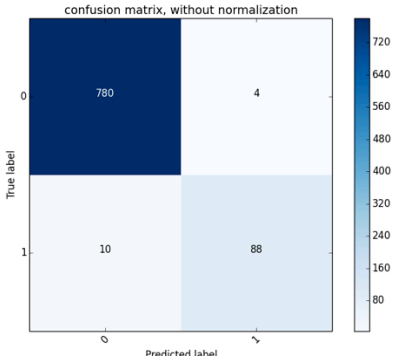

Figure 3 Confusion Matirx of Class 1

2. PROGRAM PIPELINES

2.4 ROC & Confusion Matrix

After we got the filtered data set, we converted 11 classes into 11 different corresponding combinations of binary classes with the specific class being positive. That is, for example, in the binary class of Andromeda, we set all samples labeled with 'Andromeda' positive and all samples with other labels negative. Then with the

combination of file named 'filtered_pridiction.csv' which is used to generate positive labels' scores, we created 11 inputs for ROC functions. That way we generated 11 ROC curves for each of Malware Class.

As for the binary class files, not only did we generate real labeled binary classes, but also the predicted binary classes according the decision tree model. That way we were able to create confusion matrices for each of Malware Class.

3. LIMITATIONS

3.4 Non-manually Set Thresholds of ROC

Since we used scikit learn as our external tool to help create ROC curves, the built-in functions do not provide the interfaces with setting the number of thresholds when drawing a ROC curve. It was automatically set according the result of predictions. Thus, we could not generate a ROC curve that was a great finish after our experiments with setting different number of thresholds

4. CONCLUSIONS & FUTURE WORK

The plan we will do in the future regarding this project is about its wrap-up process. That is to create a shell that can automate the pipeline of this project with user-friendly interface. We are also going to extend the functionality of this software to make it adaptive, saying that we can process any dataset that provides us with feature vectors, predictions and classes and generate the target Class Signature visualization to help apply visual analysis on the dataset.

5. REFERENCES

[1] Wikipedia. (2017, May 17). *Receiver operating* characteristic. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Receiver_operating_characteristi

[2] Wikipedia. (2017, May 14). *Confusion matrix*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Confusion_matrix