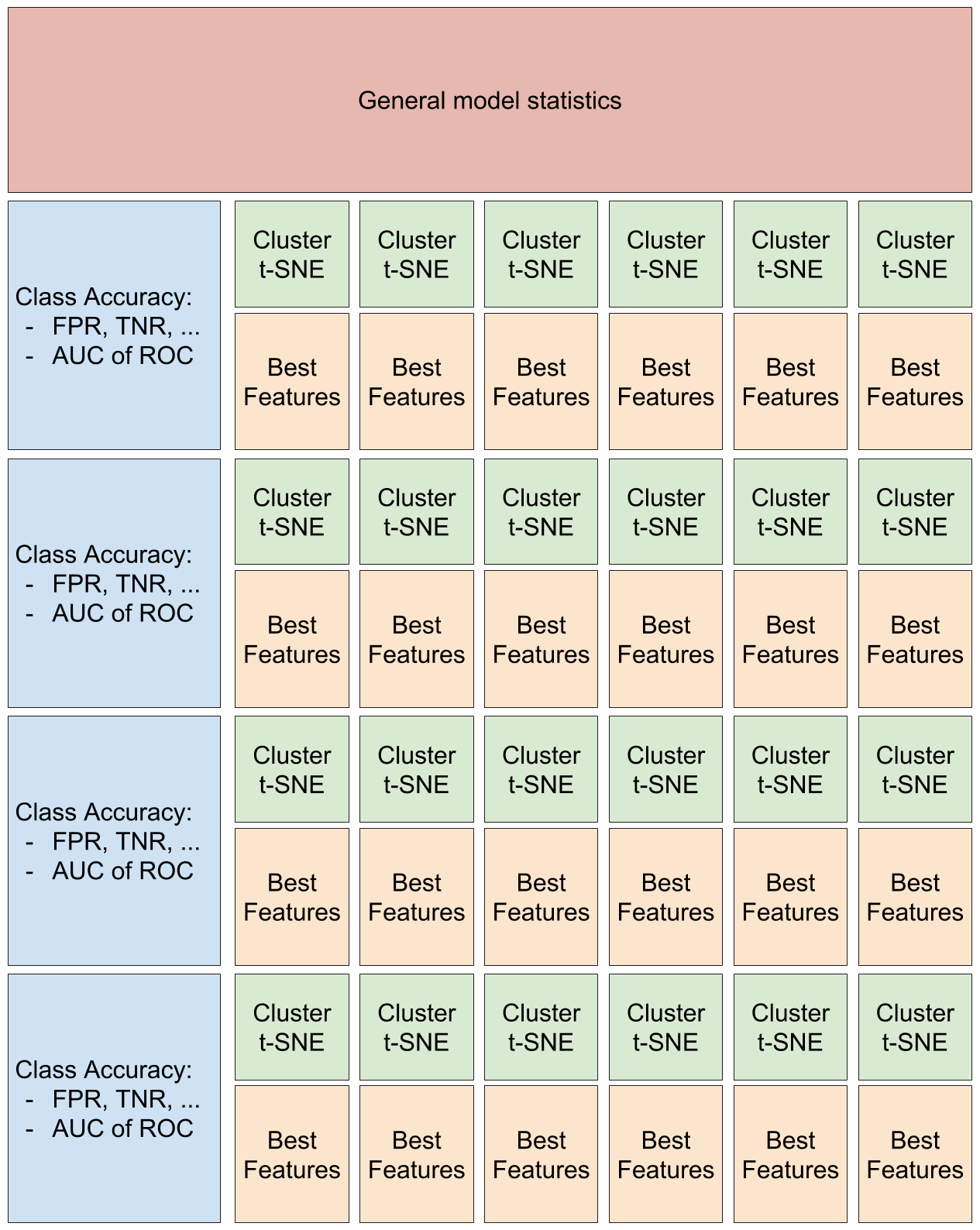
**ABSTRACT**

*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 an 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**

Class Signature proposes a visual analytics workflow to interpret predictive associations between a large set of binary features and a binary target. It usually uses 4 step pipelines, Model, Contrast, Cluster and Rank.[1]. In our Project, we extended this technique to apply it on multiple targets not binary anymore. Our target is to use Class Signature to analyze a dataset of 1100 malware samples with 347 features and 11 target families, to distinguish discriminative features among each sub-cluster of 11 families as shown in Figure 1.



**1.1 Overview of Clustering**

The original data we got from Tristan are malware ID in sha256 values labeled with family names (financial-100-c.csv) and correspondent malware feature vectors(financial-100-c-metrics.csv). There are 3 modules to accomplish our clustering process. Firstly, we needed to apply pre-processing work on the original data. We split the malware samples into 11 groups in accordance to malware family names. Secondly, we applied built-in k-means method of Scit-learn library on each family after splitting original dataset. The K-means generated 6 sub-clusters for 11 malware families. This groups together malware samples that have the similar configuration of features. Thirdly, as for generating proper input for T-SNE and Feature Ranking, Module 3 generated 66 label files in accordance to 66 sub-clusters of original dataset. The 66 label files consist of value 1 and value 0, where value 1 indicates a sub-cluster of malware samples that should be highlighted in T-SNE phase and correspondent features be analyzed in Feature Ranking Phase.

**2 RELATED WORK**

Scikit-learn (formerly scikits.learn)is a machine learning library for the Python programming language. It features various classification, regression, and clustering algorithms including support vector machines, random forests, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries Numpy and Scipy. [6]

K-means clustering is a method of vector quantization.

It aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. [7]

**3 PROGRAM PIPELINES**

We divided our entire project work into 4 steps.

Step 1 is to generate 6 clusters for each family of malware. Step 2 is to visualize the entire dataset of malware using t-SNE and highlight each cluster, thus we will have 66 t-SNE figures, one for each cluster).  Step 3 is to compute Feature Ranking for each cluster. Step 4 is to automate this process and apply on all datasets.

4 **LIMITATIONS**

In our project, we use our cluster.py program to accept the original data files and generate output files that can be used as input for TSNE and Feature Ranking. However, our labor was divided into these three main function blocks, which causes that there are some asynchronized parts among them. For example, cluster program has to be developed first to ensure input for TSNE and Feature Ranking. We developed two versions of cluster program during project 1, where the first version can generate TXT files as outputs and the second version can generate csv files as outputs. And the TSNE can take csv files as input, whereas Feature Ranking code can take txt files as the input. So that our first version code of cluster can serve Feature Ranking and the second version can serve TSNE.Thus, the whole program can not be ran in a single run. Since the first version of cluster code has been overridden, only output in txt format of first version can be accessible in our Repository’s Feature Ranking input folder.

Future Work

In term of clustering part, we will create a program that can serve both our subsequent phases -- TSNE and Feature Ranking.