Technical Report

Group 8 – Philipa Kolade, Smith Adoctor, Dorcas Owusu

Objectives

- Pre-process raw extracted OpCode files into 1-Gram and 2-Gram features
- Apply SVM, KNN and Decision Tree classifiers on 1-Gram and 2-Gram OpCodes
- Evaluate the performance of the three classifiers

Introduction

By analyzing the Opcodes of software, we can make assumptions about the nature of a program. The knowledge gained from this analysis can assist in distinguishing between benign programs and malware. It can also reveal associations with specific APT groups.

This report explores the application of three machine-learning classifiers—Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree—to 1-Gram and 2-Gram OpCode sequences. Through systematic pre-processing, feature extraction, and performance evaluation, our goal was to assess the efficacy of these classifiers in accurately identifying APT groups based on OpCode data.

Methodology

Dataset Overview:

- The first step taken by our team was developing a pre-processing script which reads opcode data and their labels from opcode files and converts the opcode sequences into 1-gram features. The script creates a csv file named 1gram_features and writes the 1-gram features into it.
- We maintained the same process as above and created a separate script for pre-processing our opcode files into 2-gram sequences. The script stored its output in a csv file named 2gram_features.
- Once we had tested our scripts to ensure they worked correctly, we manually removed the APT group subdirectories which had less than five opcode files from our main input directory, leaving us with 19 APT groups. We did this because, such groups provide insufficient data for machine learning algorithms to learn meaningful patterns, leading to

overfitting or unreliable generalization. These small classes often introduce noise and can degrade overall model performance by prioritizing unrepresentative minority classes. By focusing on groups with sufficient data, we enhance the model's ability to generalize, reduce noise, and improve overall performance metrics, ensuring a more robust and meaningful analysis. We then ran the pre-processing scripts to get our working datasets.

Data preparation:

- Data Cleaning: Analysis of our datasets revealed that we had no missing values, however our datasets contained duplicate instances, 179 duplicates for the 1gram_features dataset and 177 duplicates for 2gram_features dataset. Our datasets contained outliers, which constituted 6.13% of the data, however we retained them because they may represent unique and critical behaviors in opcode usage specific to certain APT groups. The 1gram_features dataset did not have constant features, but the 2gram_features dataset had 154 constant features. We performed data cleaning by removing the duplicate instances and constant features.
- Feature Selection: Due to the large number of features in the 2gram_features dataset (12,737), we implemented feature selection to reduce computational complexity and memory usage. We used ANOVA F-test to measure how different feature values are between classes and kept only statistically significant features. This reduced the number of features to 6453.
- Class Balancing: Given the imbalance across APT groups, we applied SMOTE to create balanced training datasets. This method generates synthetic samples for minority classes, helping to balance the dataset and improve model performance.
- Data Normalization: We implemented StandardScalar (standardization) to normalize our data because SVM and KNN perform better with standardized features, and we do not need our features to be in a specific range.
- Data Splitting: We split our data into three sets for training, validation and testing. We used 60% (980) of our data for training, 20% (327) for validation and another 20% (327) for testing.

ML classifiers:

- We trained the SVM, KNN and Decision Tree classifiers using our clean and normalized data, and calculated the following metrics - Accuracy, Recall, Precision, F-measure and Confusion Matrix to evaluate the performance of each classifier in identifying APT groups from Opcodes

Results

1-gram analysis

SVM

Test Metrics

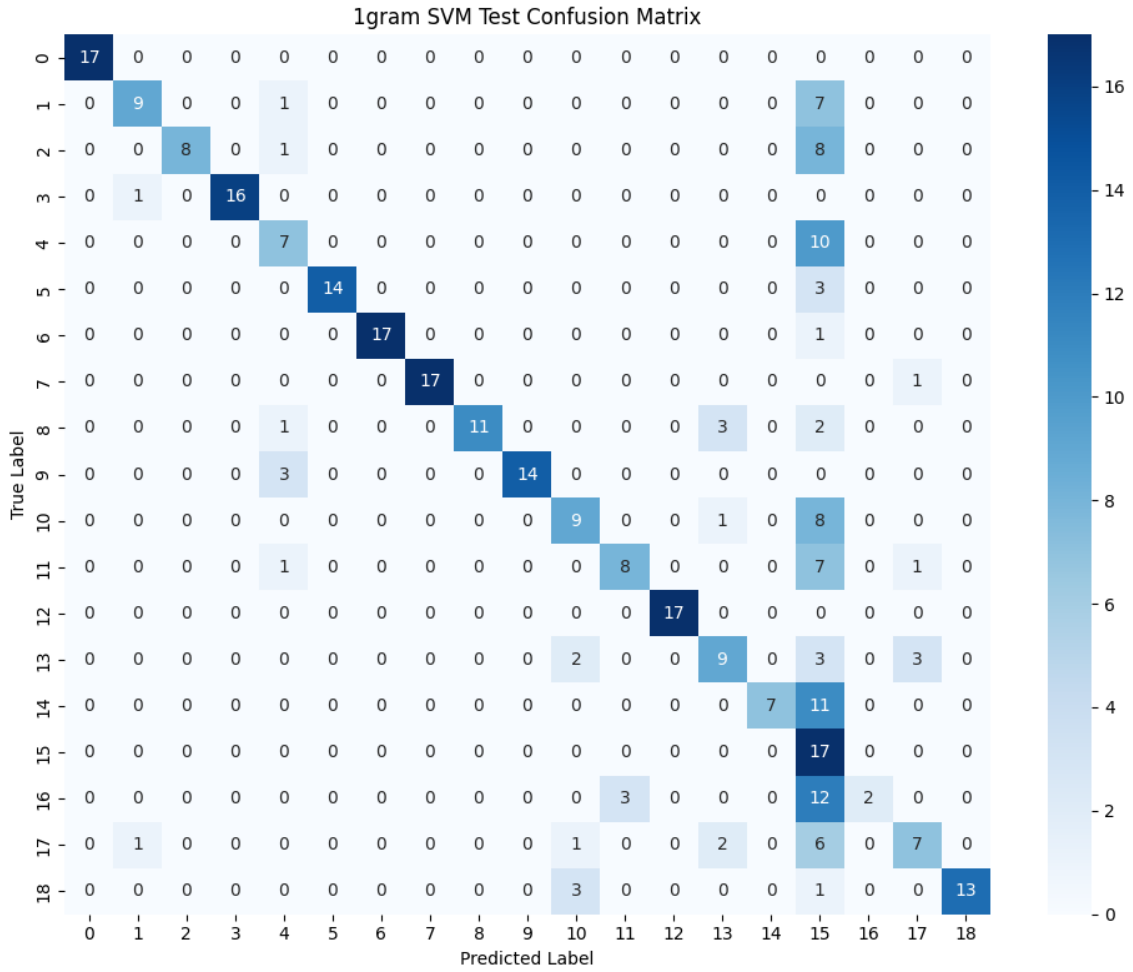
Accuracy: 0.6697

Precision: 0.8431

Recall: 0.6697

F1-score: 0.7028

Confusion Matrix:



KNN

Test Metrics:

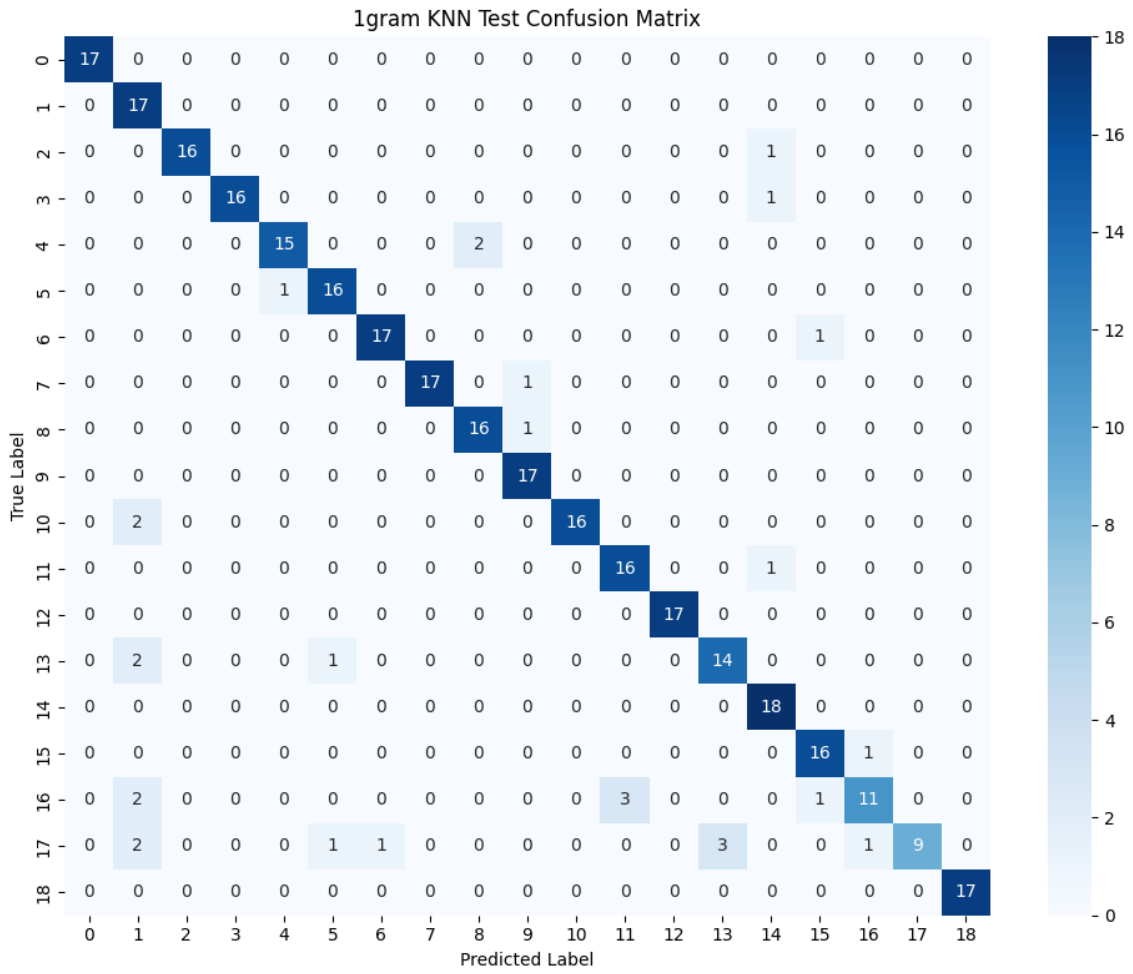
Accuracy: 0.9113

Precision: 0.9210

Recall: 0.9113

F1-score: 0.9091

Confusion matrix:



Decision Tree

Test Metrics:

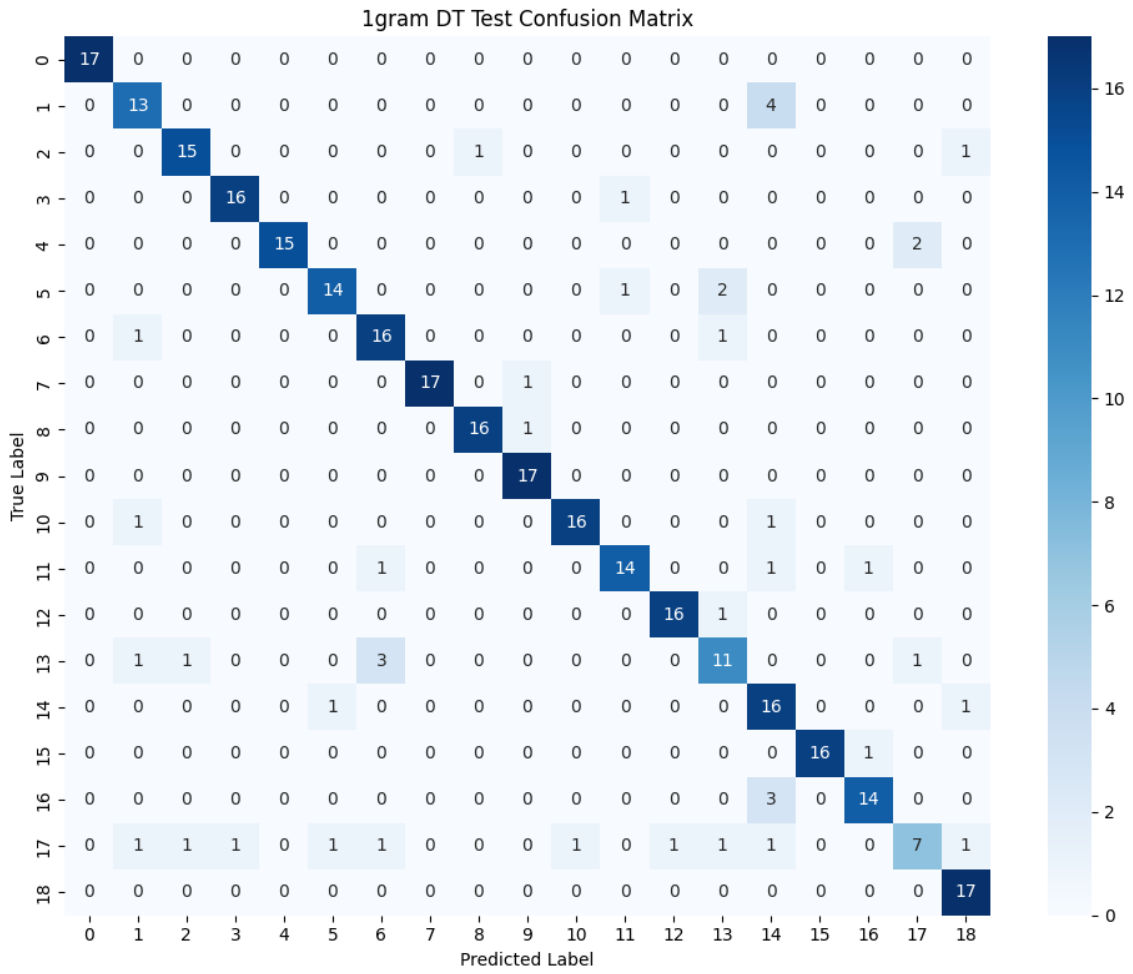
Accuracy: 0.8654

Precision: 0.8704

Recall: 0.8654

F1-score: 0.8634

Confusion matrix:



2-gram analysis

SVM

Test Metrics:

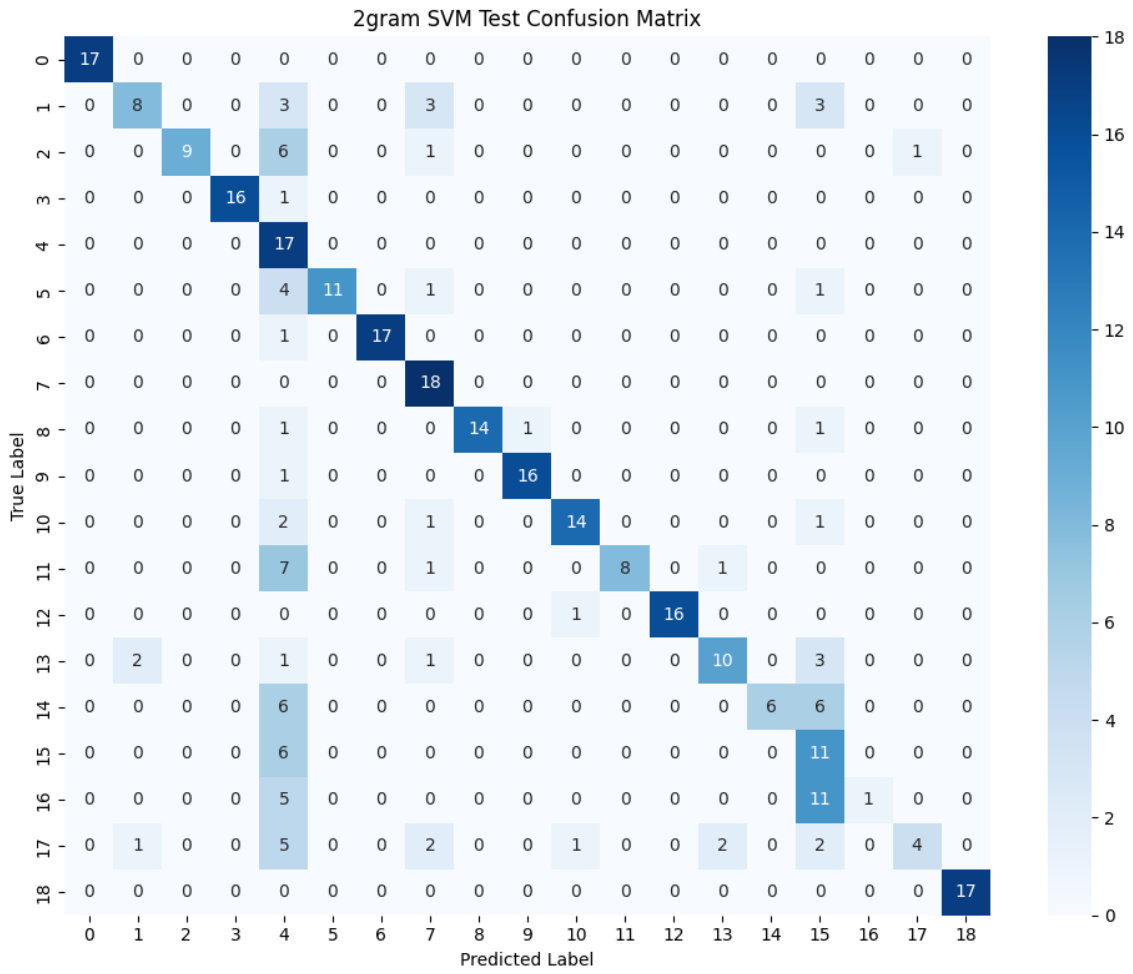
Accuracy: 0.7034

Precision: 0.8579

Recall: 0.7034

F1-score: 0.7110

Confusion matrix:



KNN

Test Metrics:

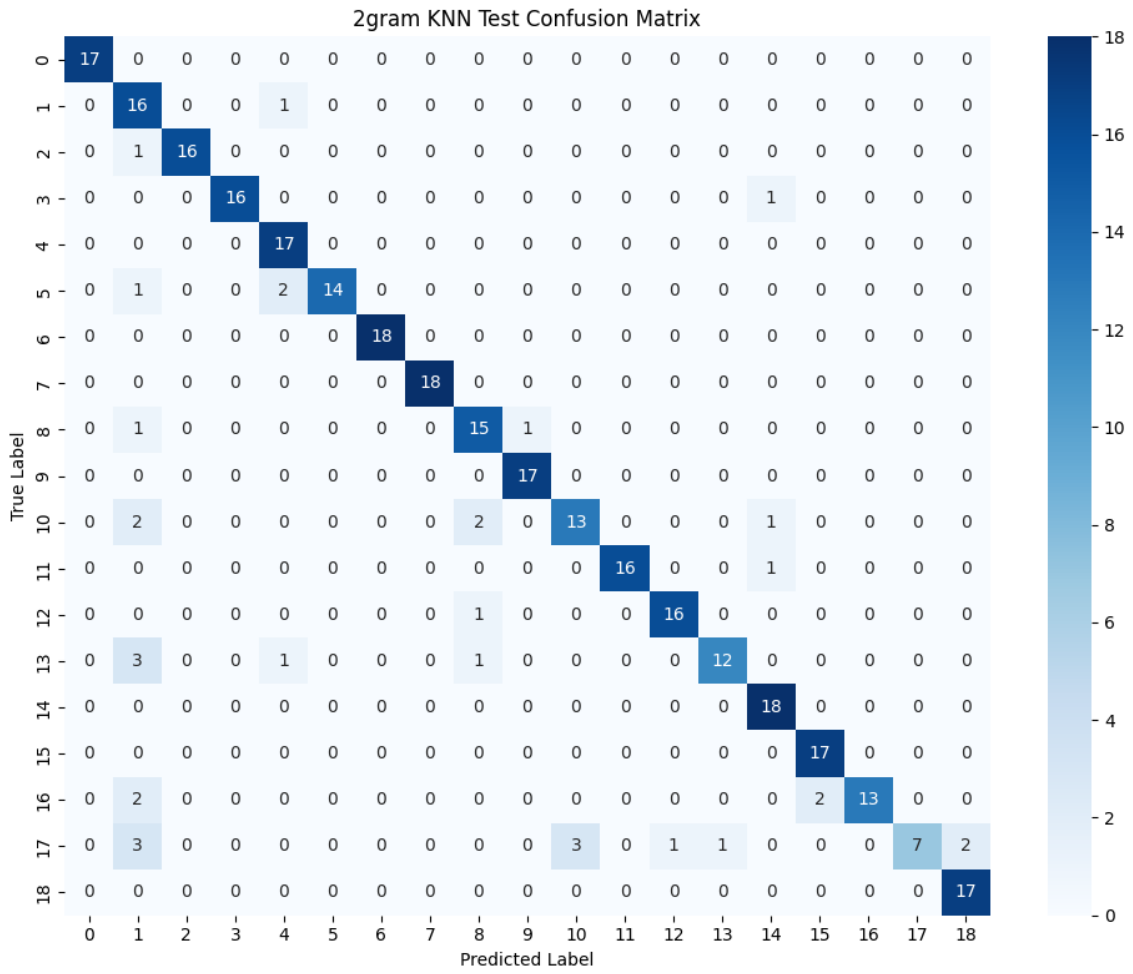
Accuracy: 0.8960

Precision: 0.9168

Recall: 0.8960

F1-score: 0.8938

Confusion matrix:



Decision Tree

Test Metrics:

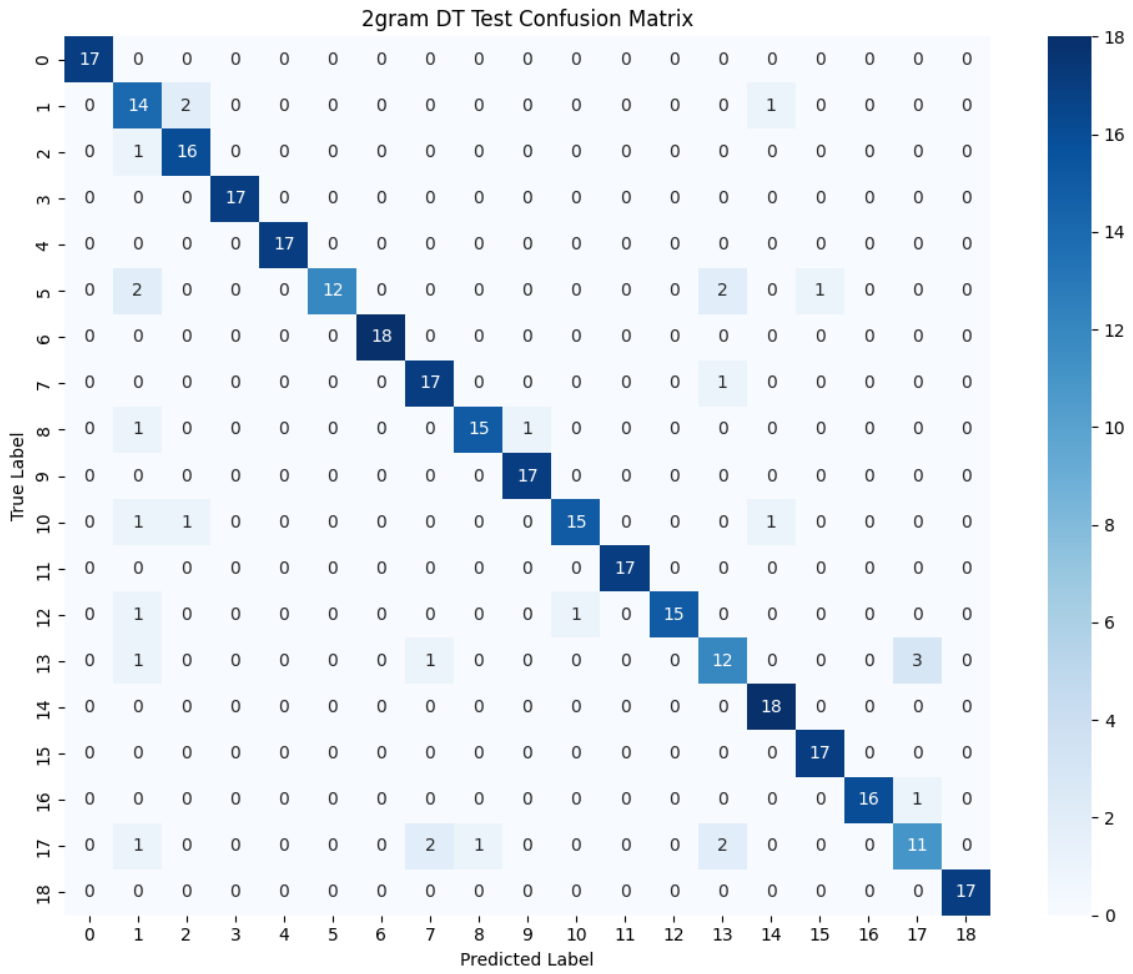
Accuracy: 0.9113

Precision: 0.9175

Recall: 0.9113

F1-score: 0.9114

Confusion matrix:



Analysis of results gotten:

1-gram results:

- KNN was the most consistent classifier, with an accuracy of 91.13%. From the confusion matrix, APT groups 17 and 18 (index 16 and 17 from the matrix) show some misclassifications, but correct predictions are made for most of the classes.
- Decision Tree was the next best classifier with an accuracy of 86.54%. From the confusion matrix, Decision Tree showed misclassifications with APT groups 14 and 18 (index 13 and 17in the matrix).
- SVM had a decent performance, with an accuracy of 66.97%. The confusion matrix shows that it often confused and misclassified test data as belonging to APT 16 (15 in the matrix).

2-gram results:

- Decision Tree performed better in 2-gram classification, with an accuracy of 91.13%, It shows a stronger diagonal and better classification in some APT groups.
- KNN was the most consistent classifier, with an accuracy of 89.60%. It performs well and shows a great diagonal. There are some misclassifications, especially in APT 18 (17 in the matrix)
- SVM also showed improved classification with 2-gram opcodes, like in Decision tree, however it was still the worst performing classifier. In 2-gram classification, it had an accuracy of 70.34%, and was particularly bad in classifying APT 17 (16 in the matrix)

Challenges Encountered

- Data Imbalance: The imbalance in OpCode data among APT groups hindered classifier accuracy and generalization. This challenge was addressed using SMOTE.

- KNN Performance Issues: For KNN, the high dimensionality of the 2-gram dataset posed challenges, as distance-based methods perform poorly in high-dimensional spaces. Feature selection techniques like ANOVA F-tests mitigated this issue to some extent but did not entirely overcome it.

- KNN (k=3.5) did not work in our code. The issue arose because KNN requires k to be an integer, as it represents the number of nearest neighbors. Fractional values for k are not valid in the algorithm's implementation. In our case, this issue was addressed by using k=3, which is the nearest valid integer

- Low Sample Counts for Certain Classes: APT groups with very few samples (fewer than five opcodes) were removed to prevent overfitting and unreliable generalization, streamlining the training process while maintaining meaningful analysis. Inclusion of these APT groups degraded the performance of the classifiers

Conclusion

From the results, KNN emerged as the most consistent classifier across both 1-gram and 2-gram analyses, achieving an accuracy of **91.13%** in 1-gram classification and **89.60%** in 2-gram classification. Its performance was marked by strong precision and recall, though minor misclassifications were observed in some APT groups. Decision Tree followed closely, with an accuracy of **86.54%** in 1-gram classification and **91.13%** in 2-gram classification, demonstrating better handling of group-specific misclassifications. SVM, while effective, lagged with

accuracies of **66.97%** and **70.34%** for 1-gram and 2-gram classifications, respectively. The confusion matrices highlighted consistent misclassification of APT groups with overlapping opcode patterns, particularly in SVM.

The application of SMOTE to address class imbalance contributed significantly to improving model performance across all classifiers, enabling them to generalize better despite the inherent challenges posed by the dataset's complexity. This study highlighted the potential of machine learning in malware classification using OpCode analysis.

Appendix

APT groups and their mapping

APT 41 - 1

broze butler - 2

cobalt group - 3

dark carcacal - 4

deep panda - 5

dragonfly - 6

evilnum - 7

fin7 - 8

gamaredon - 9

gcman - 10

ke3chang - 11

molerat - 12

mustang panda - 13

naikon - 14

nightdragon - 15

pittytiger -16

scarlet mimic - 17

turla - 18

volatile cedar – 19

Link to Github: https://github.com/MCTI-UOG/APT_submission_4