Decoding GAN-Generated Malware using Explainable AI Techniques

Matteo Tiozzo • 6th week 2024/10/22 - 2024/10/29

Overview

Progress

- TF-IDF
- PCA
- Grayscale images

Major risk

- PCA

TF-IDF

Before applying TF-IDF, I removed mnemonic pairs with fewer than 200 occurrences from the "mnemonics_summary.txt" file, resulting in the "filtered_mnemonics_summary.txt". Then, following the method explained last week, I calculated the TF-IDF and stored each result in the main table "tfidf_table_main_assembly.csv".

The same process was applied to the hexadecimal data.

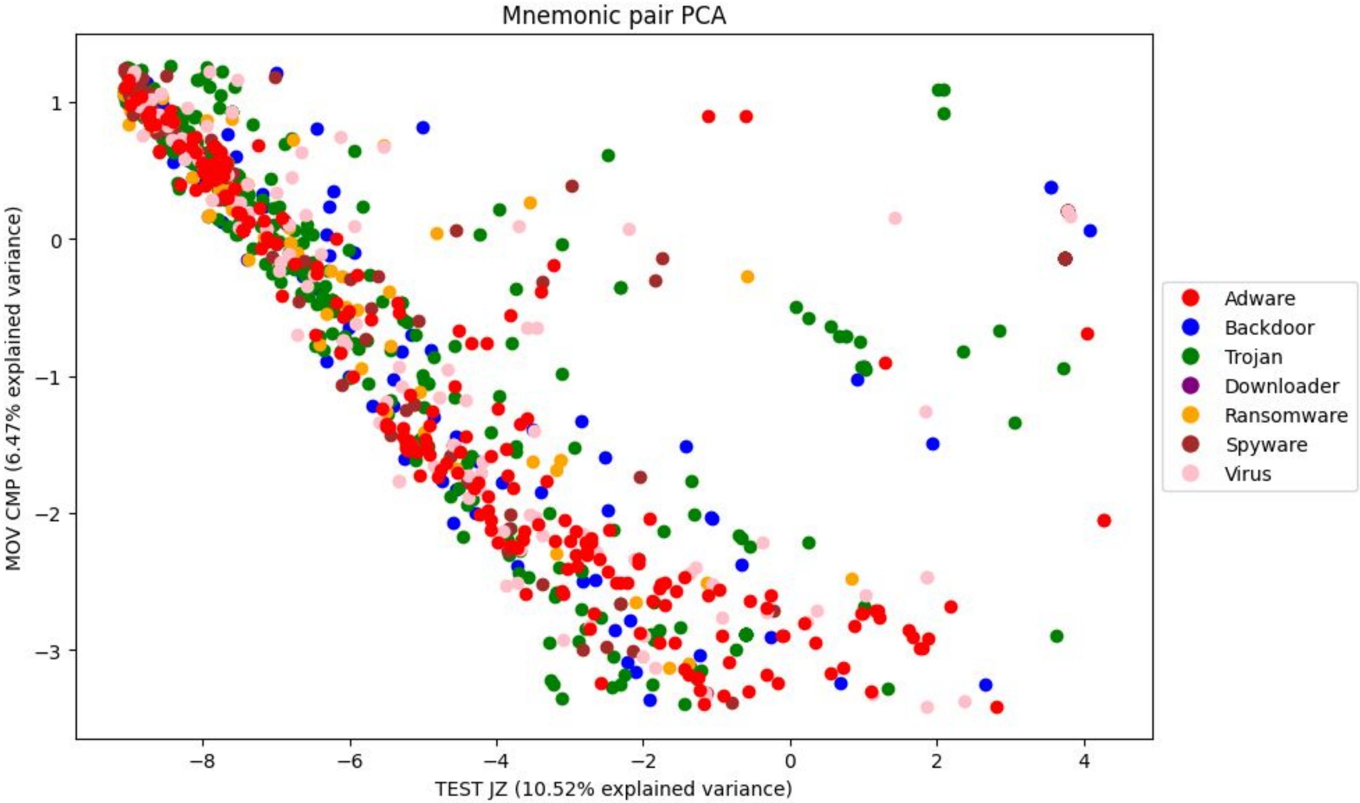
PCA

After completing TF-IDF and obtaining the required table, I applied PCA to reduce the dimensionality of the data.

The code takes all instructions as input and calculates the PCA, selecting "TEST JZ" as the first principal component (x-axis) and "MOV CMP" as the second principal component (y-axis).

A standard normalization was applied to the data.

PCA


Mnemonic pair PCA
MOV CMP (6.47% explained variance)
TEST JZ (10.52% explained variance)
1
0
−1
−2
−3
−8
−6
−4
−2
0
2
4
Adware
Backdoor
Trojan
Downloader
Ransomware
Spyware
Virus

Grayscale images

I generated grayscale images using the filtered folder from the previous step as input. Specifically, I used the folder containing files within the 2.5% to 97.5% percentile range.

Tools used this week

Python

- TF-IDF
- PCA
- Graphs
- Grayscale Images