**Cyber-Attack-Detection**

A COURSE PROJECT REPORT

By

**AYUSH KUMAR GUPTA (RA2011026010038) AKASH PS (RA2011026010003)**

**SWAPNIL SINHA (RA2011026010006) DHRUVA BHATTACHARYA (RA2011026010023)**

Under the guidance of

**Lavanya S K**

*In partial fulfilment for the Course*

of

18CSE388T - ARTIFICIAL NEURAL NETWORKS

in Department of Computational Intelligence (CINTEL)



**FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Chenpalpattu District**

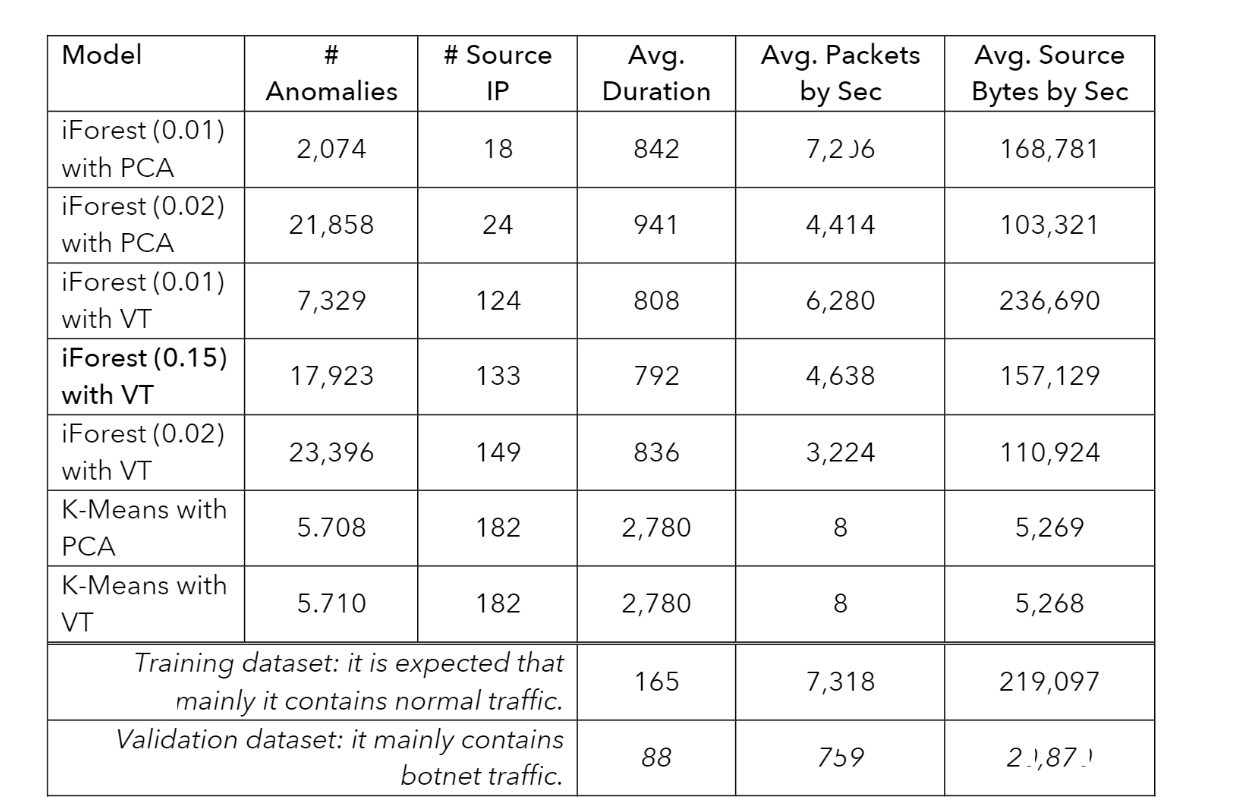
NOVEMBER 2022

1. **Problem Statement**

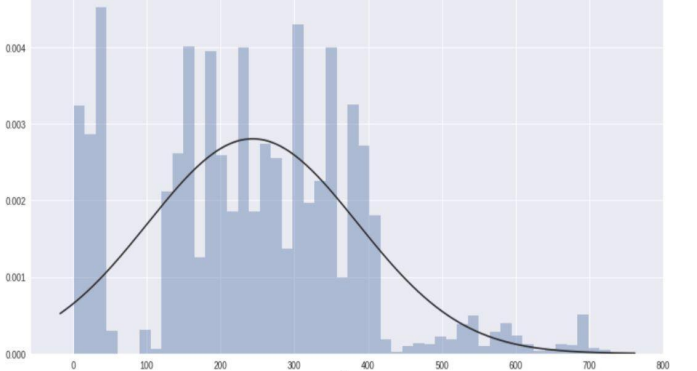
* ***Cyber-attack detection using*** machine learning/Deep learning.
* Applications to the cyber security.
* Supervised vs. Unsupervised.
* Algorithms and methods.
* Conclusions methods used isolation forest and k means clustering.
* Traditional methods of intrusion detection and deep packet inspection, while still largely used and recommended, are no longer sufficient to meet the demands of growing security threats.
* As computing power increases and cost drops, Machine Learning is seen as an alternative method or an additional mechanism to defend against malwares, botnets, and other attacks.
* First, a strong data analysis is performed resulting in 22 extracted features from the initial NetFlow datasets. All these features are then compared with one another through a feature selection process. Then, our approach analyzes five different machine learning algorithms against NetFlow dataset containing common botnets.
* The Random Forest Classifier succeeds in detecting more than 95% of the botnets in 8 out of 13 scenarios and more than 55% in the most difficult datasets.
* Finally, insight is given to improve and generalize the results, especially through a bootstrapping technique.

1. **Introduction**

* The rise in hacking and computer network attacks worldwide has heightened the demand for improved intrusion detection and prevention solutions. Cyberattacks and cybersecurity risks have skyrocketed with new technologies such as cloud computing, fog computing, edge computing, and the Internet of Things (IoT).
* These assaults can infiltrate computer network-related environments, cloud-based services, and social networks, causing financial and reputational damage. Network intrusion detection systems play a significant role in every computer network defense system as they detect and prevent malevolent activities.
* NetFlow analysis needs efficient deep learning methods to perform real-time traffic analysis, aided by analysis of historical network traffic using traditional machine learning methods, hybrid architectures, and novel computing paradigms, such as federated learning and granular computing. With the increasing volume of network traffic in high-speed networks, research on intelligent network traffic analysis is vital for network management and security.
* •Deep learning specifically has been studied to solve many typical NetFlow analysis tasks, which have proven to be a potential general framework for analyzing network traffic. Intrusion detection systems are critical in identifying abnormalities and assaults on the network, which have grown in size, stealth, and pervasiveness. Artificial neural networks have been used in anomaly detection to determine if data behavior is normal or aberrant.
* With reasonable performance, this network can identify both known and unknown threats. Deep neural networks are capable of self-learning and detecting previously unknown types of network attacks, in contrast to existing systems based on signature analysis. Another advantage of neural networks is their ability to detect unknown attacks (zero-day attacks), function in a noisy environment, preserve operability with incomplete or distorted data, forecast user behavior, and the emergence of new attacks.
* This Special Issue aims to provide a state-of-the-art overview of novel emerging deep-learning architectures and models for network intrusion and malware detection. Original research articles with a focus on both practical as well as on theoretical topics and problems, as well as review articles, are welcome.

1. **Data set**

**Gaussian Distribution of Datasets**



Link: [www.kalggle.com](http://www.kalggle.com)

1. **Methodology**

* **Logistic Regression Model**

Logistic regression will be used to solve this problem. We will analyze the packets flowing to determine if the requests are safe or unsafe.

* **Semi-supervised learning**

Semi-supervised learning will be used to identify the malicious packets in the network and block the connection.

* **Encrypted message**

Malicious packets have encrypted message or code embedded in their packets which will be used as data to teach the model

* **Problems faced with the model**

As the model works with the application of unsupervised learning for anomaly detection it is intrinsically more difficult than supervised learning as it does not have corresponding output.

Due to the difficulty, the algorithm might be less accurate as input data is not labeled, and algorithms do not know the exact output in advance.

Against supervised models we have used gradient descent-based method to generate adversarial samples. Therefore, an slower update learning can occur since an update is performed only after we go through all the observations.

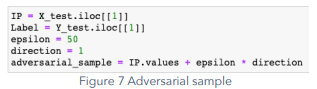
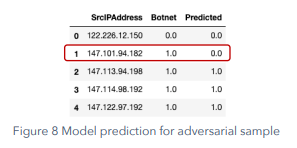
1. **Implementation**

* In this project we have explored two ML projects consisting of cyberattacks. The first task is to analyse different combinations of features anomaly detection. For the second task an sample is generated for evading the detection of a Supervised machine learning model.
* Our primary focus is the 2nd task, i.e., evading anomaly detection. Therefore we have used **two Ml techniques** used for anomaly detection:

iForest Clustering

K-Means Clustering.

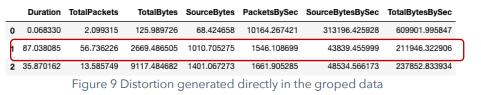
* The implementation of this method is relatively straightforward . For these tasks it was generated an adversarial sample for just one Source IP. From Figure 6 it will be chosen the row 1, where the Source IP Address 147.101.94.182 is correctly predicted as Botnet. The Figure 7 shows the code which generates an adversarial sample for this row. A couple of notes regarding this implementation:
* Epsilon was defined with a large value because the data is not normalized: it is considered unnormalized is more practical for “backpropagate” the epsilon to the NetFlow traffic.
* The sign of direction was directly set as 1: It was not possible to determine the gradient function given a Skit Learn model (there is not direct access to the function weights). Figure 7 Adversarial sample Finally, in Figure 8 it is presented the prediction of the model for this new adversarial sample. As we can verify, now the model does not predict the Source IP as Botnet. The accuracy of the model decreases to 0.89.

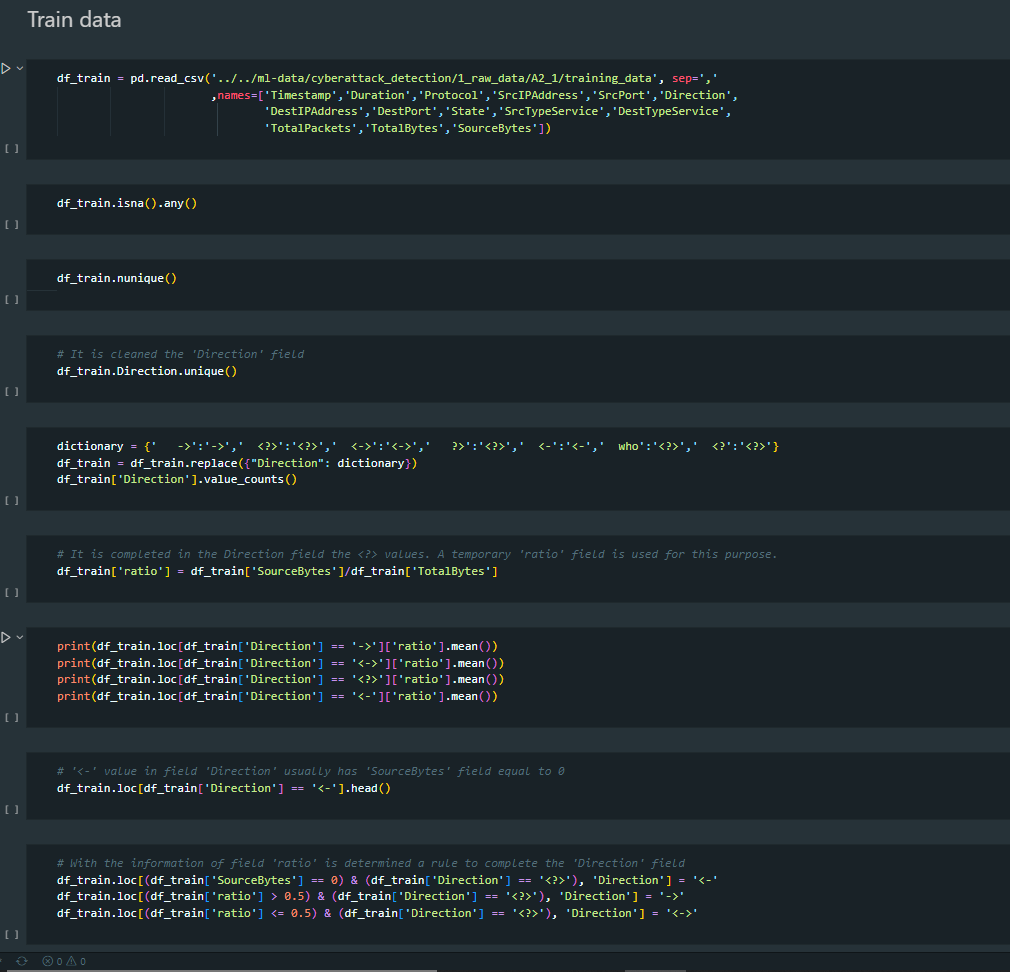


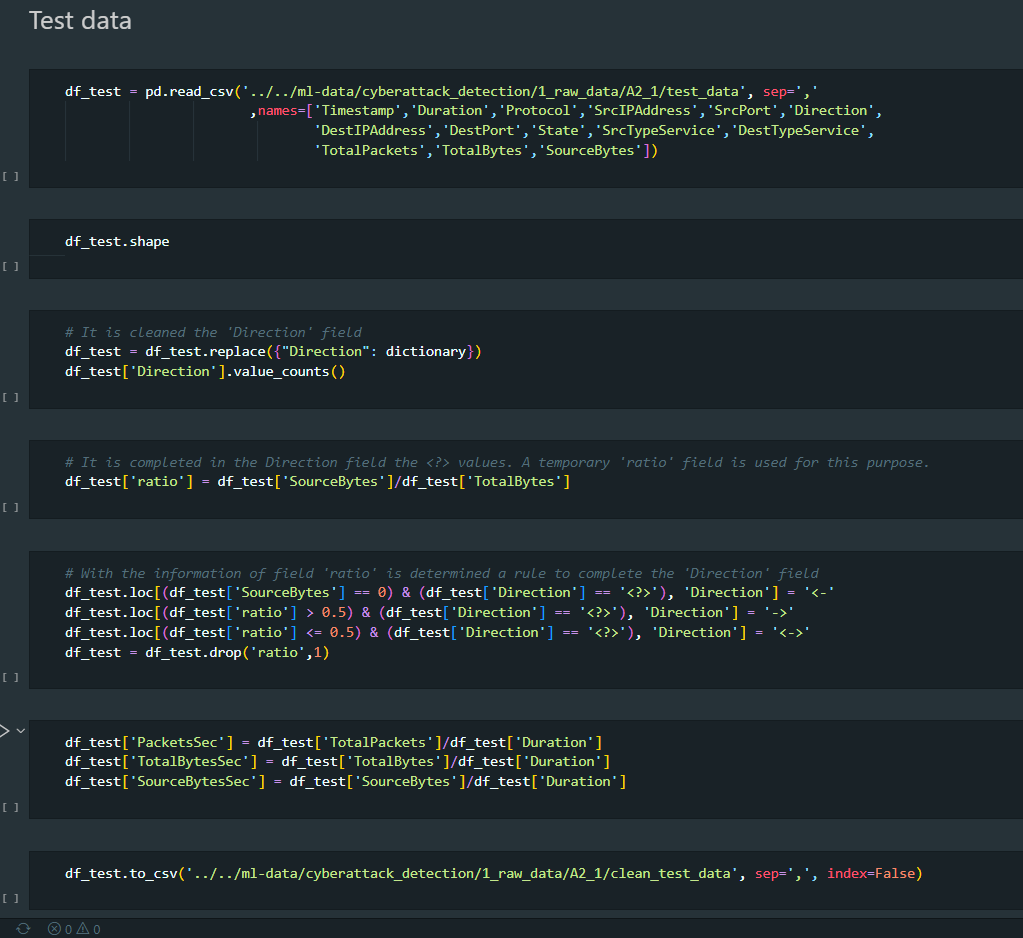
https://lh5.googleusercontent.com/yDnC4__v5fjyFd4VHwxouirJqZS851EQyqWX0EquX1jBpmrmieYpPb1s8LQaofNLL7meuBKYvpyLsz7zHxHSPbL7xniNLDd4PV9j2WFmQZd24GjEVuX7DVuauRpurJPsrG4_Cbqb1MereAClkBiKpCxXnYKC5ZcZLT72zJruHQvDgDvcNrv7GOwk6Q

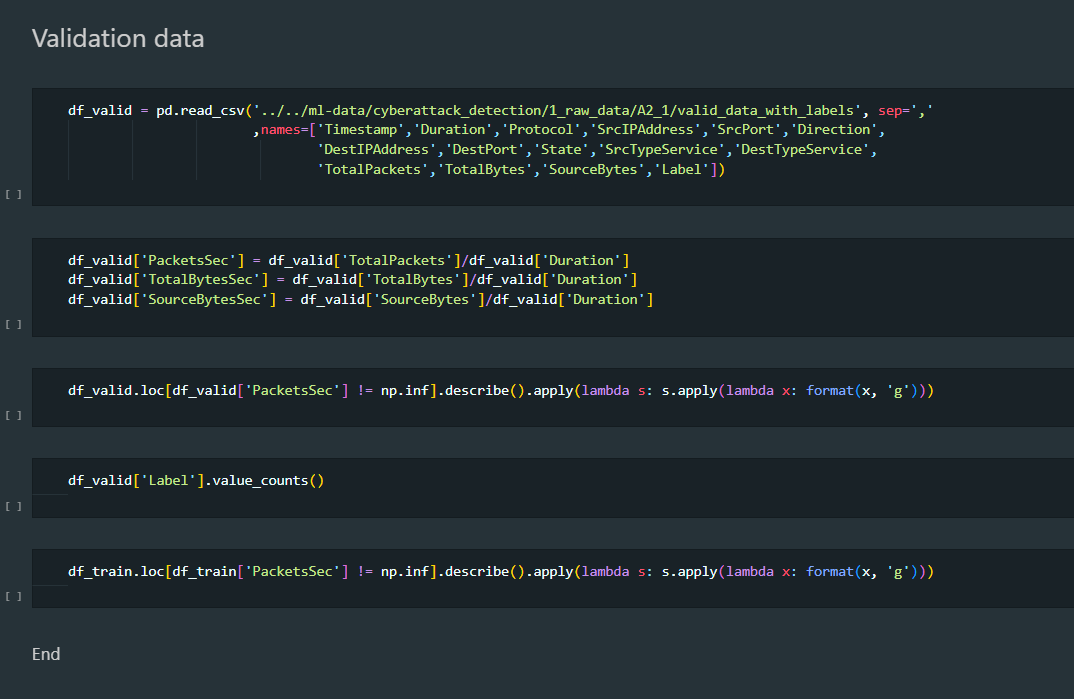
**Traffic perturbation**

* The main challenge was to introduce the epsilon generated in section 8.1 (epsilon = 50) in the NetFlow traffic given under a group operation. The first attempt consisted in just use this epsilon in all the traffic related to the target IP Source. This idea did not work because the model has generated features as Total Bytes by seconds that are created from the division of another two features. The introduction of the epsilon in all the traffic distorted these types of metrics. The second attempt considered introducing the effect of this epsilon accumulated in just one traffic row.
* This attempt is registered in the file A2\_preprocessing-perturb-traffic. Even when it was generated a perturbation very similar to the target, again, the generated features are “defending” the model against this attack. The Figure 9 and Figure 10 show how similar is the distortion generated.

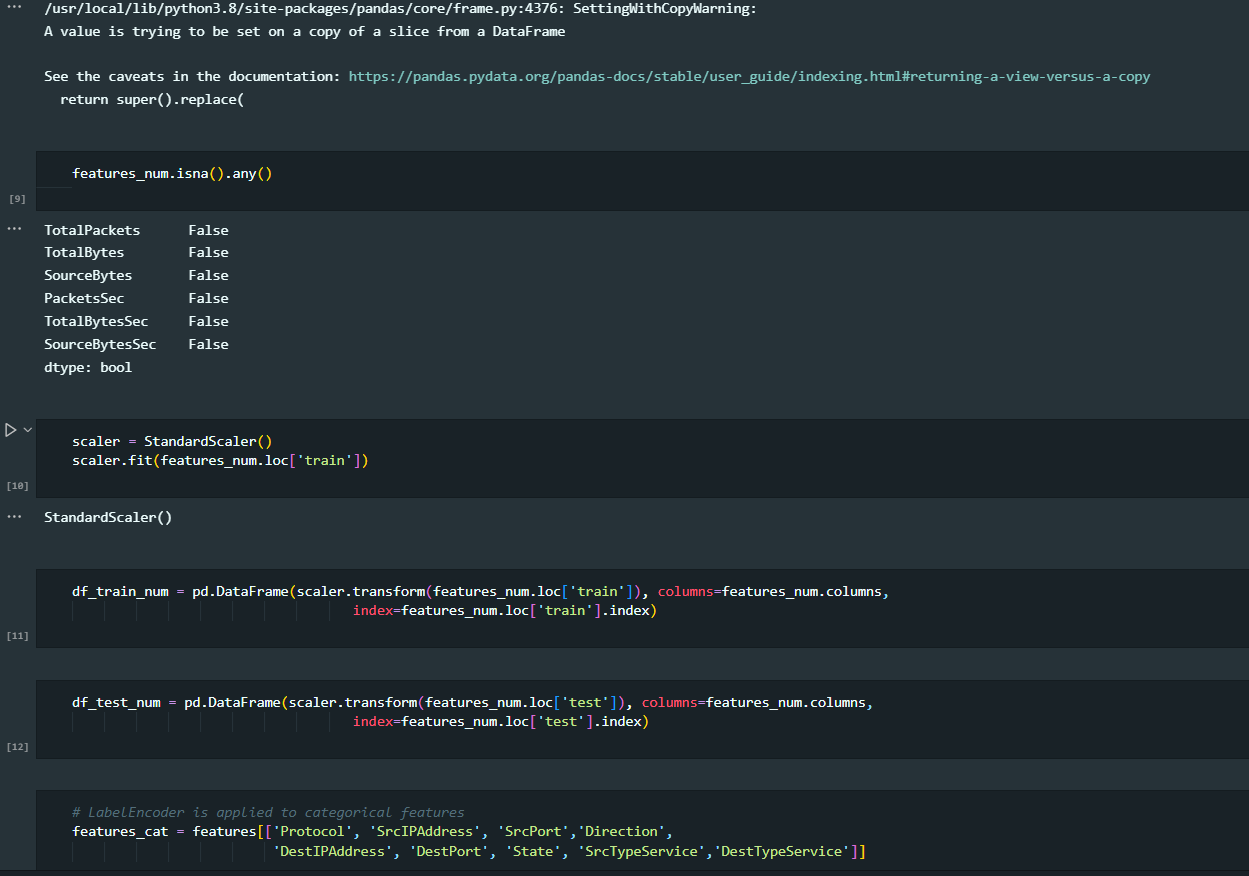
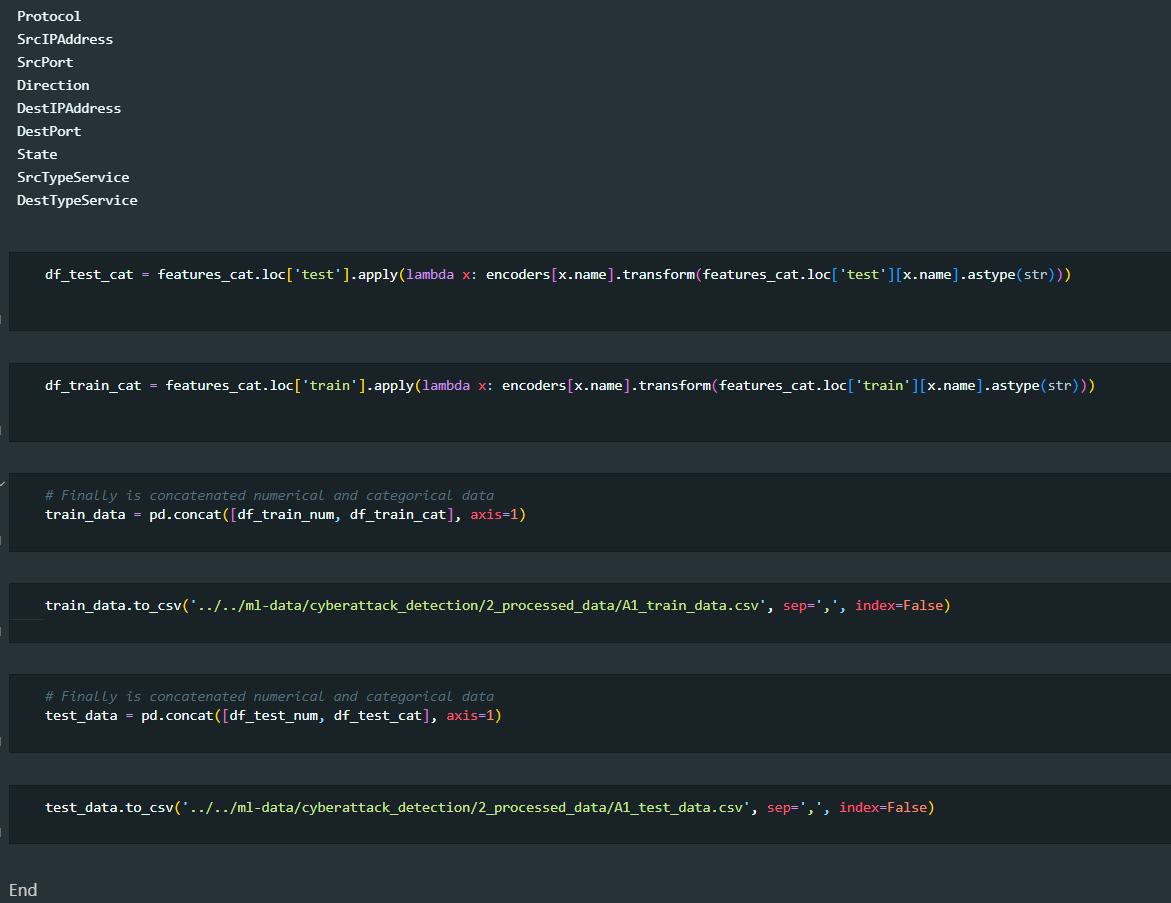


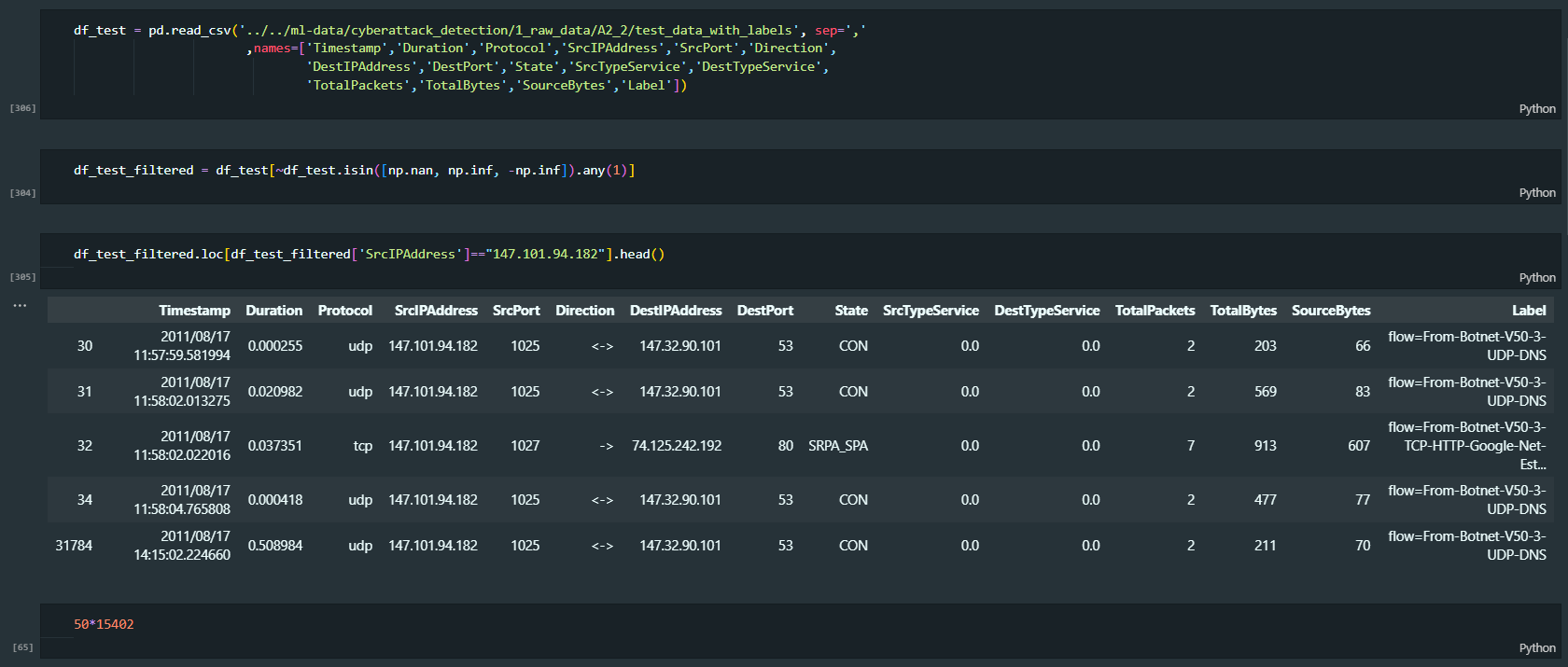


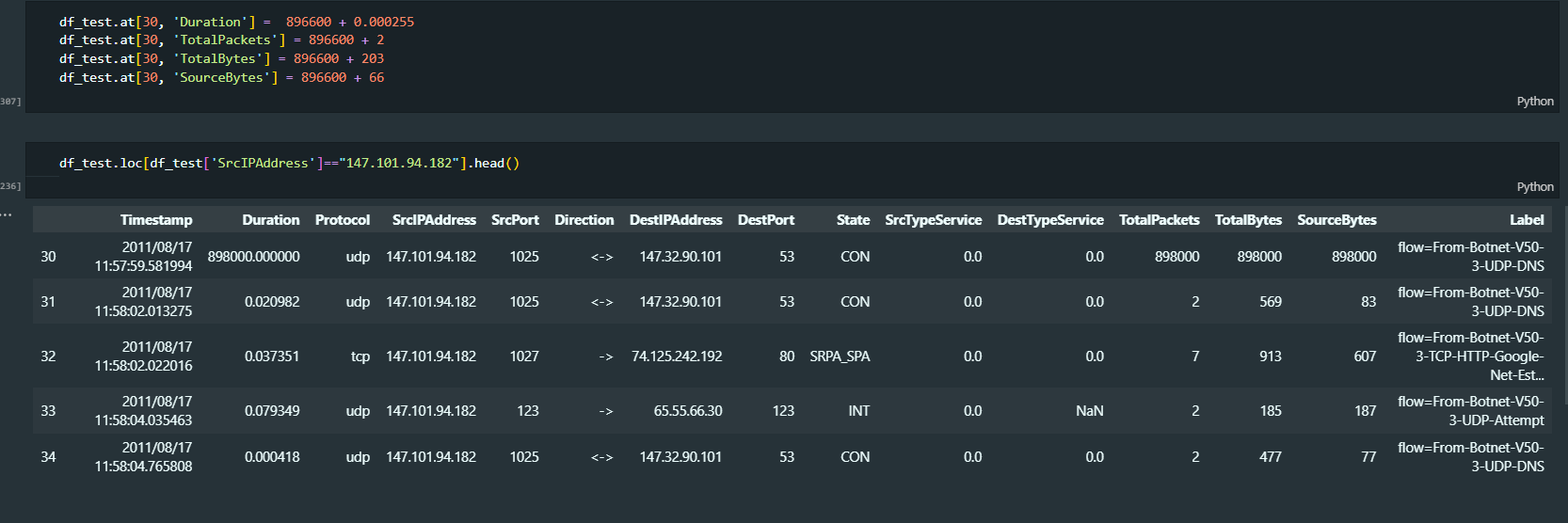


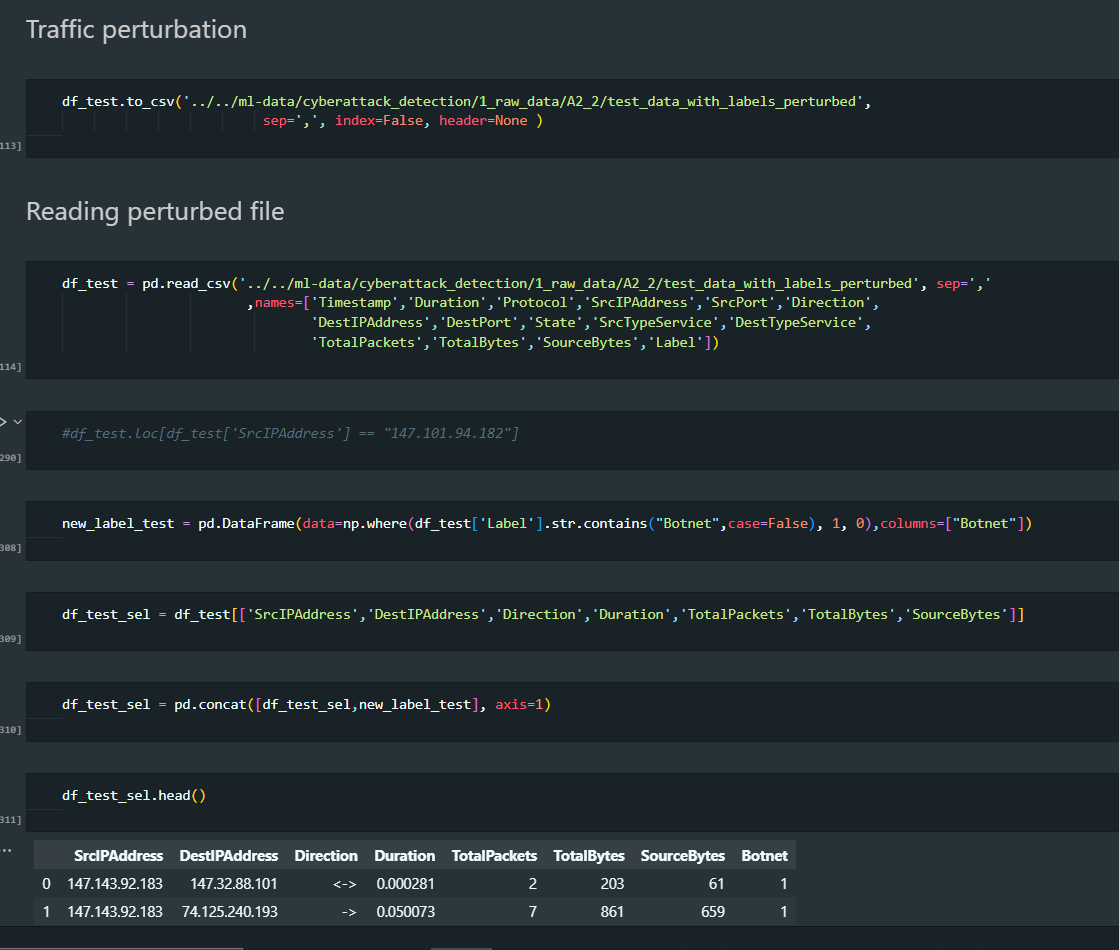


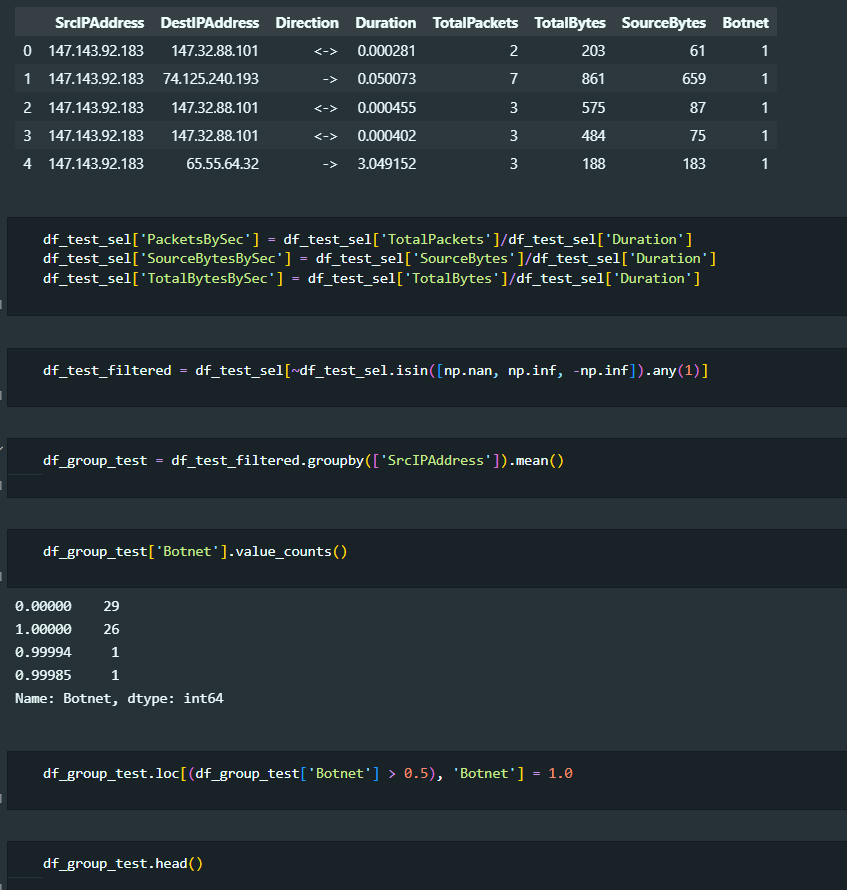


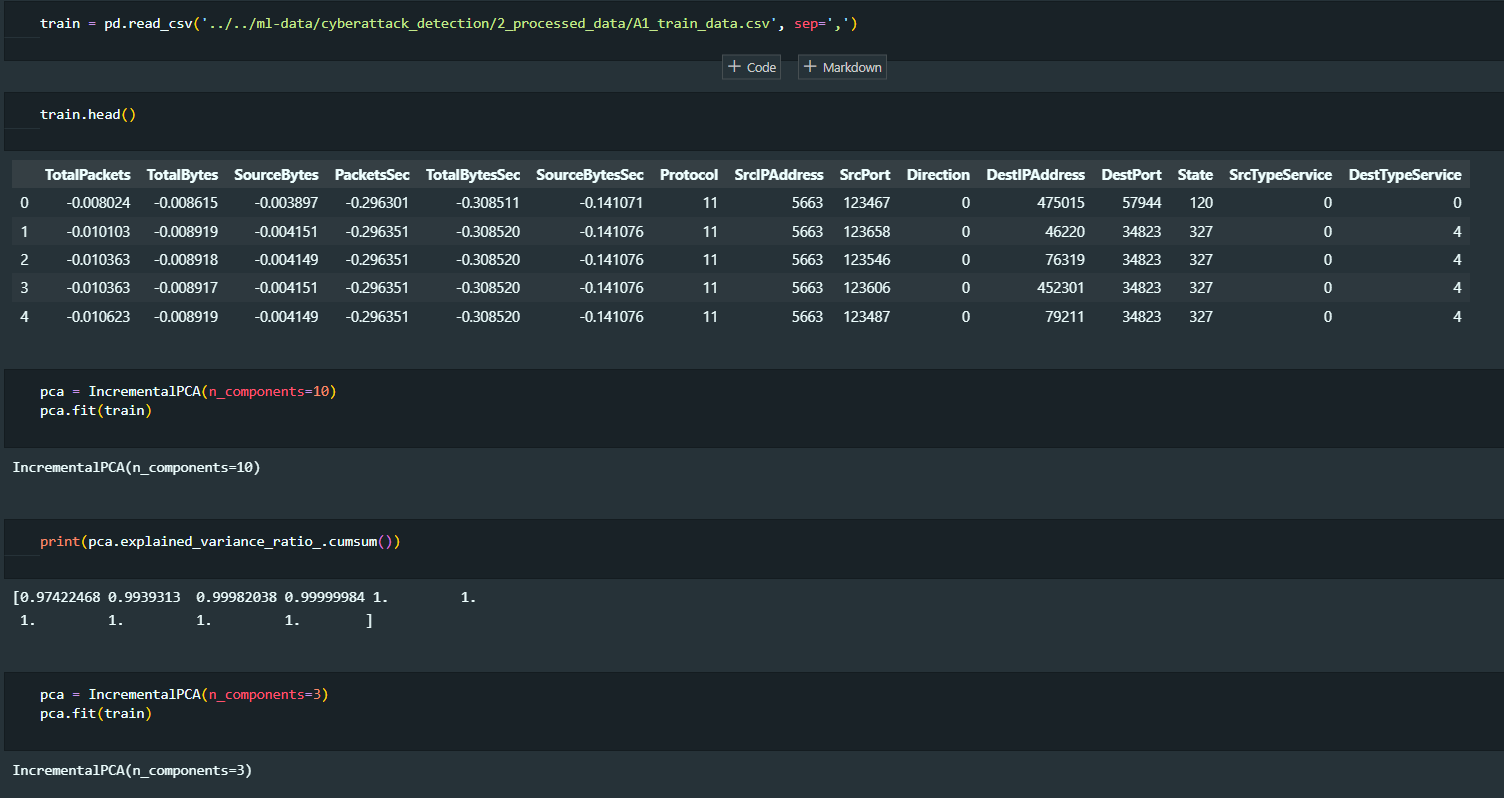


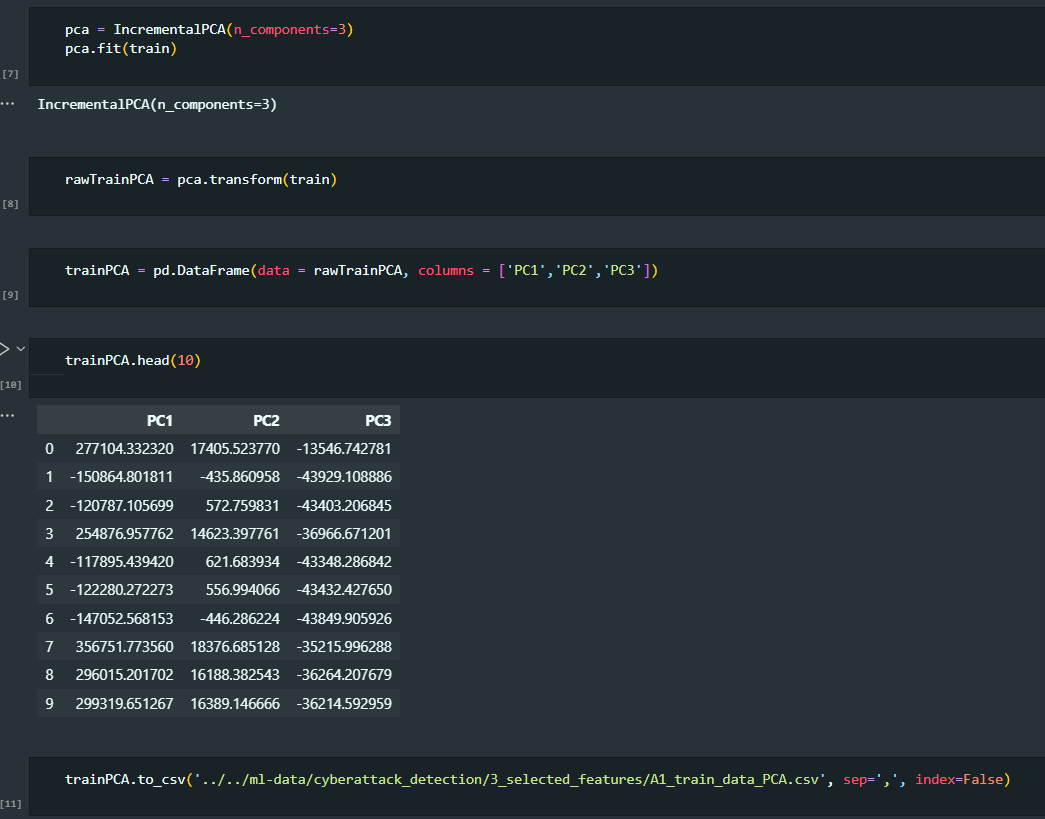


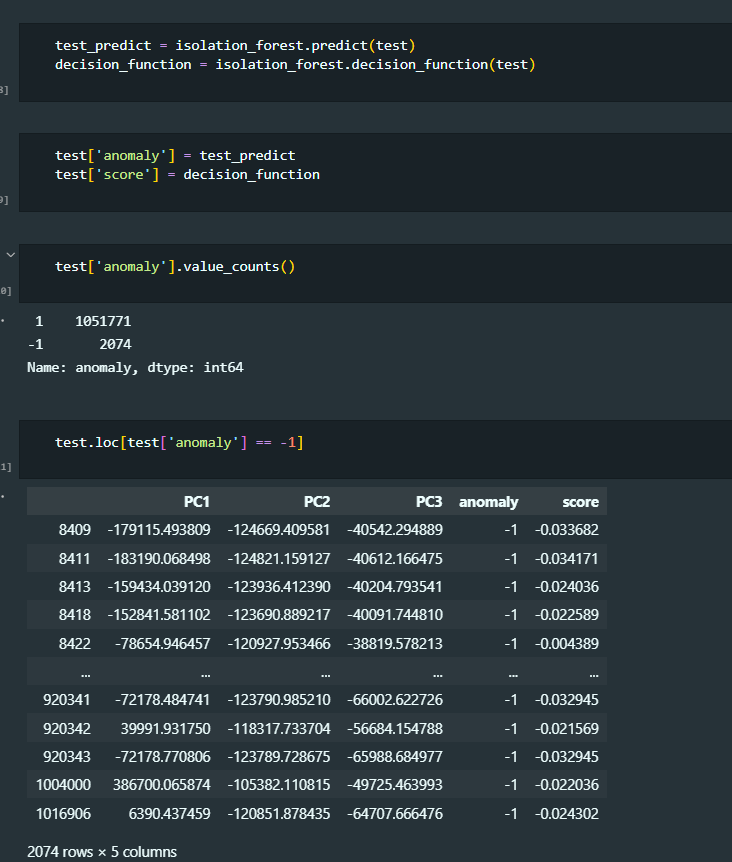


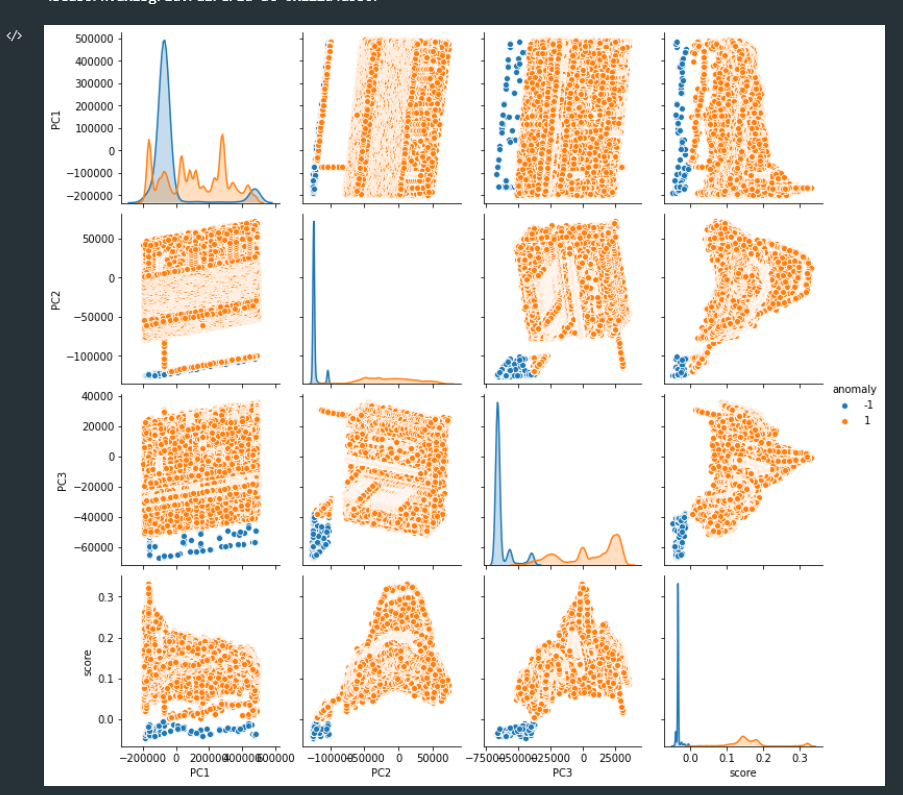


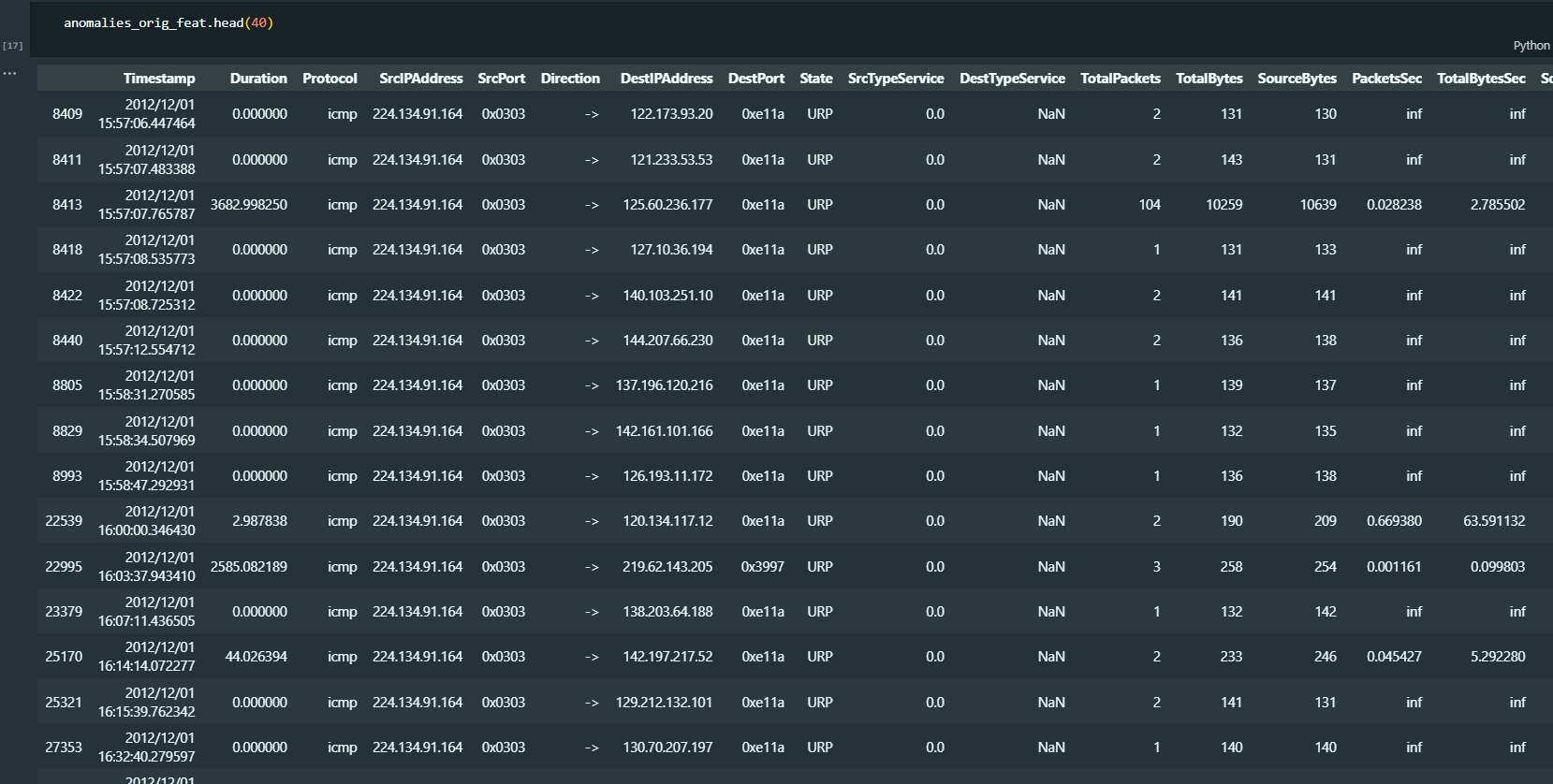


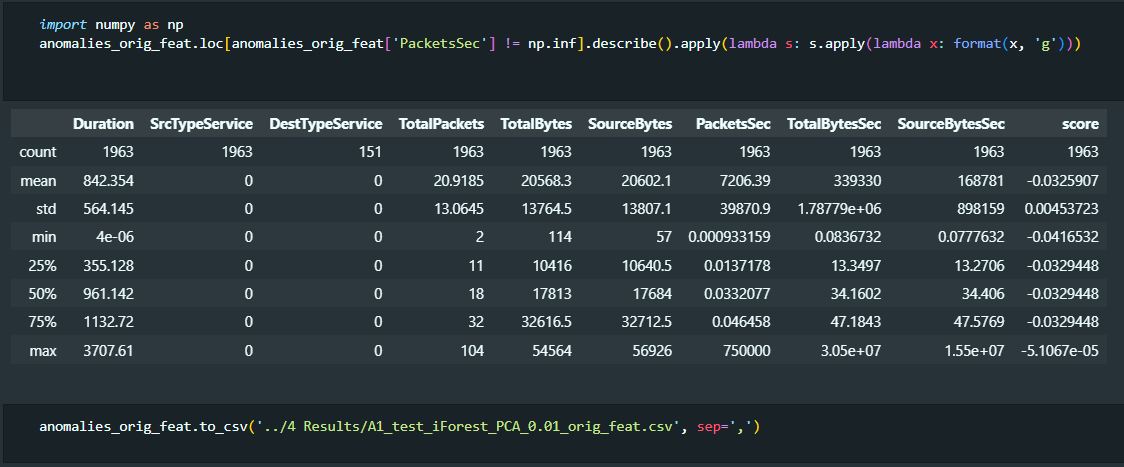


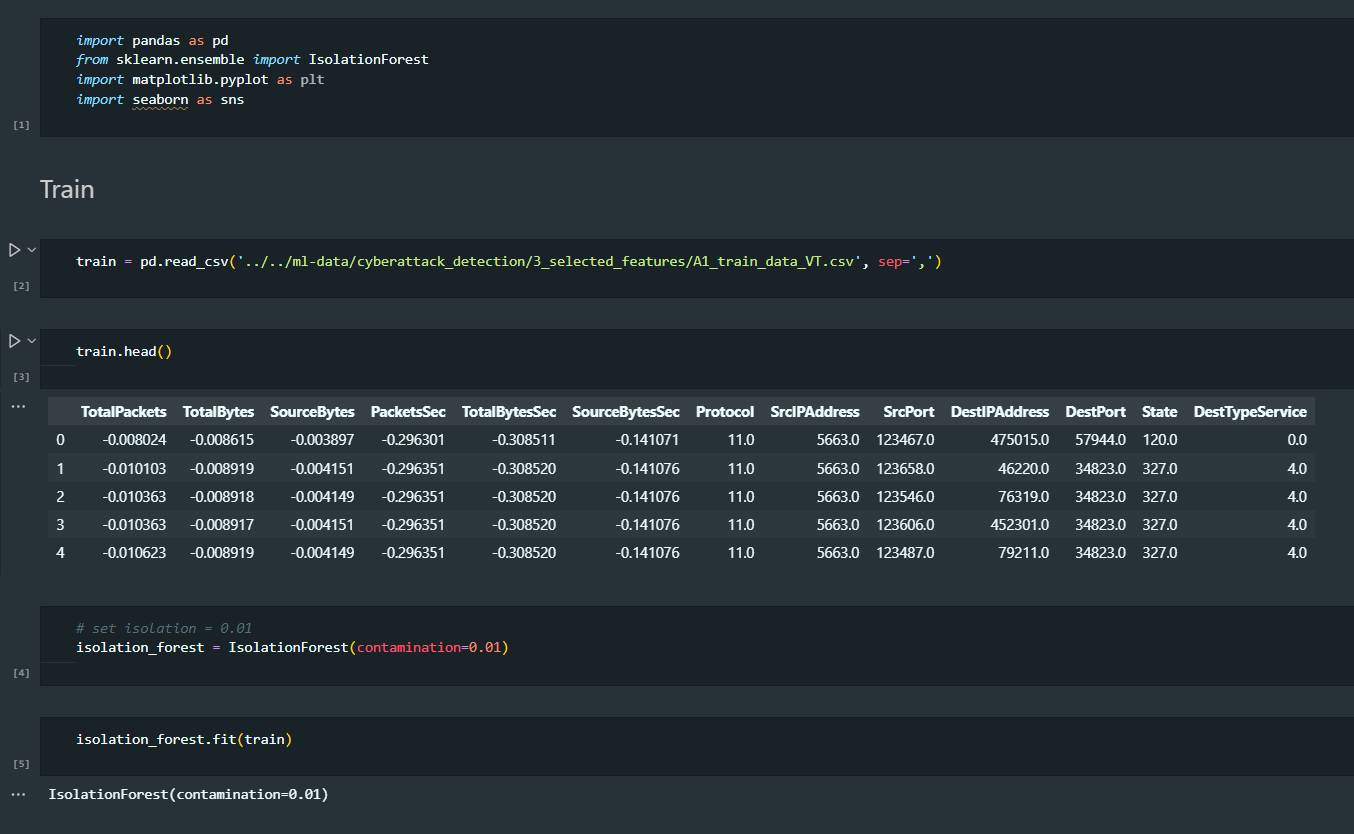


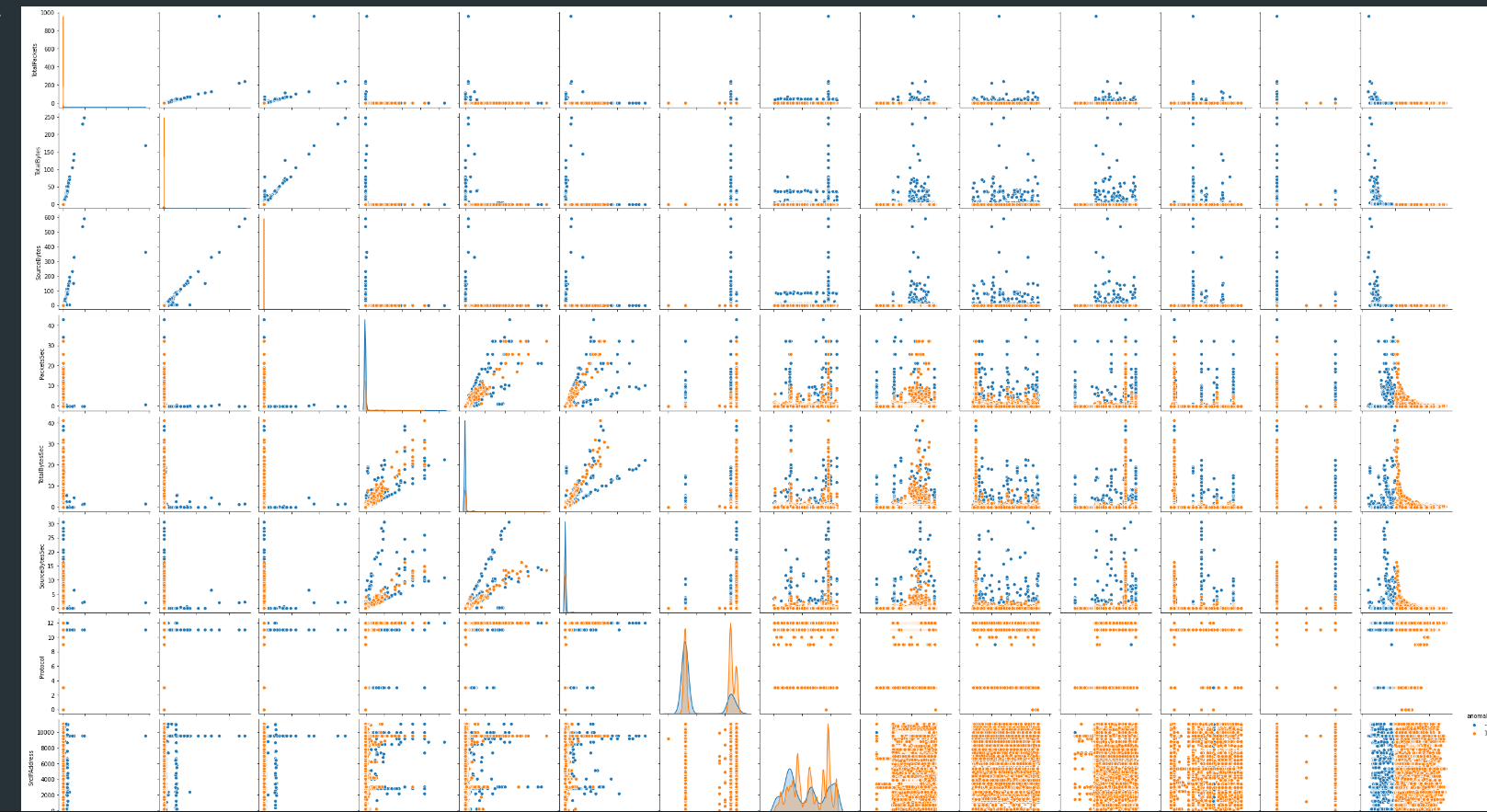


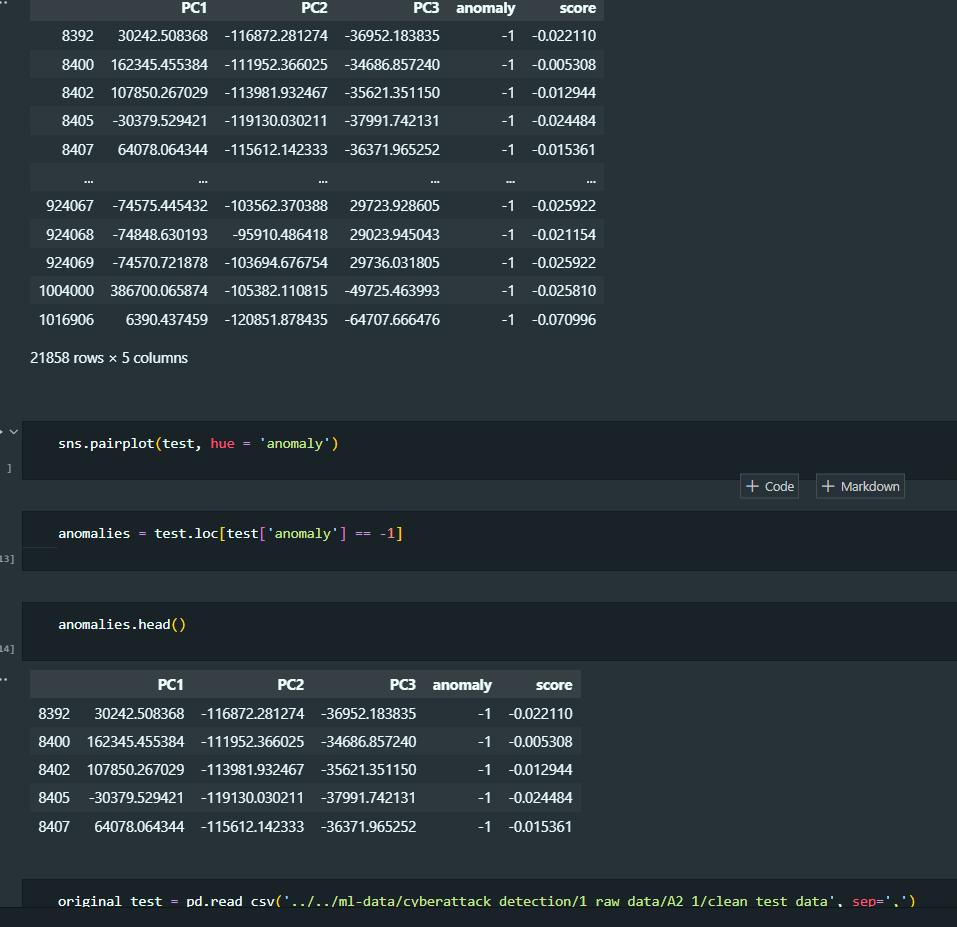


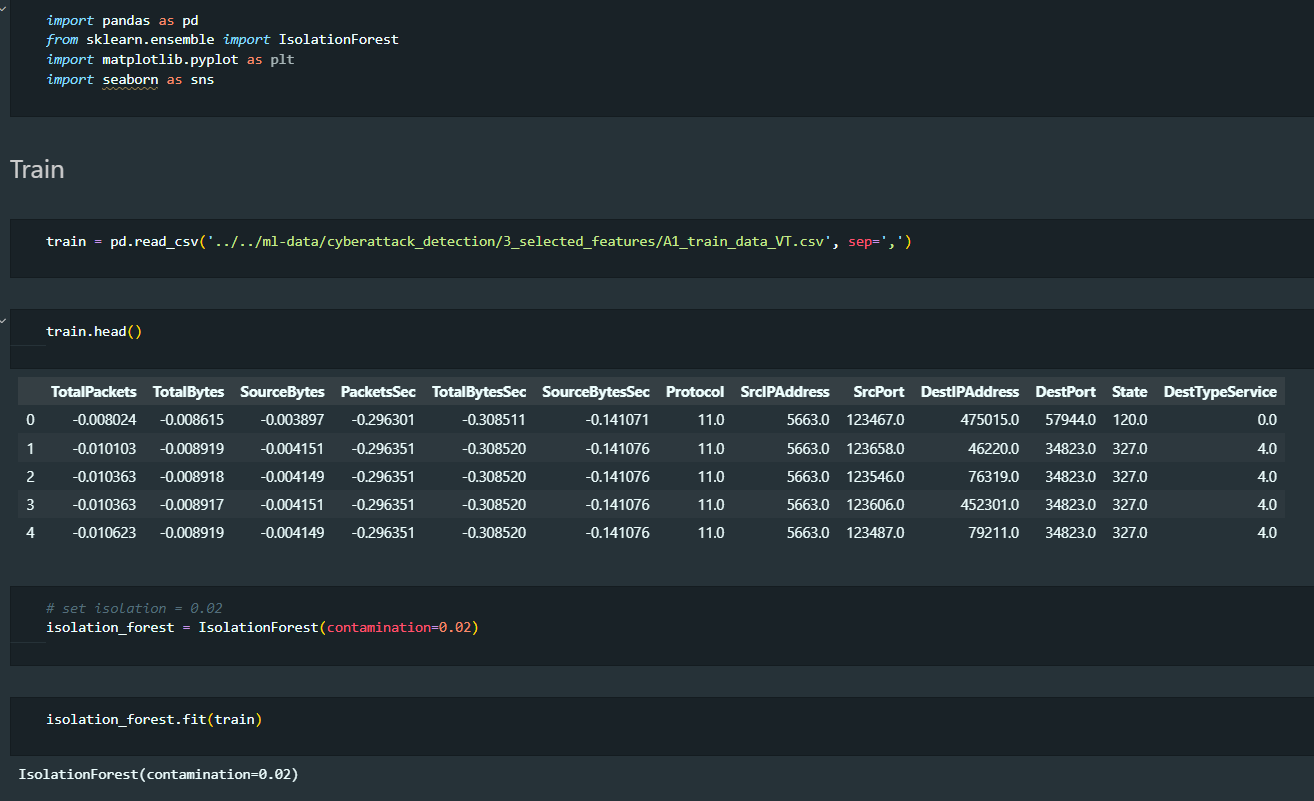


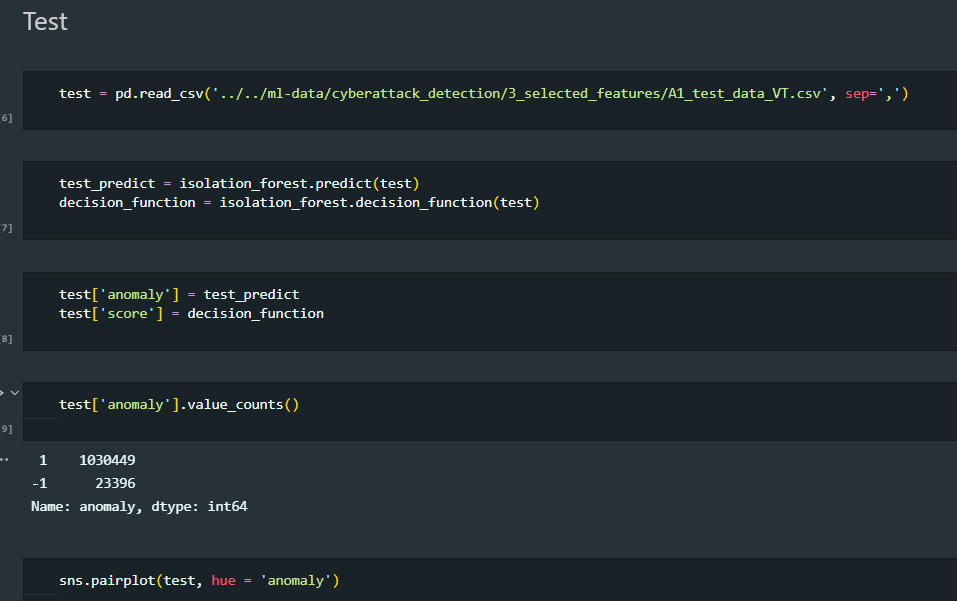


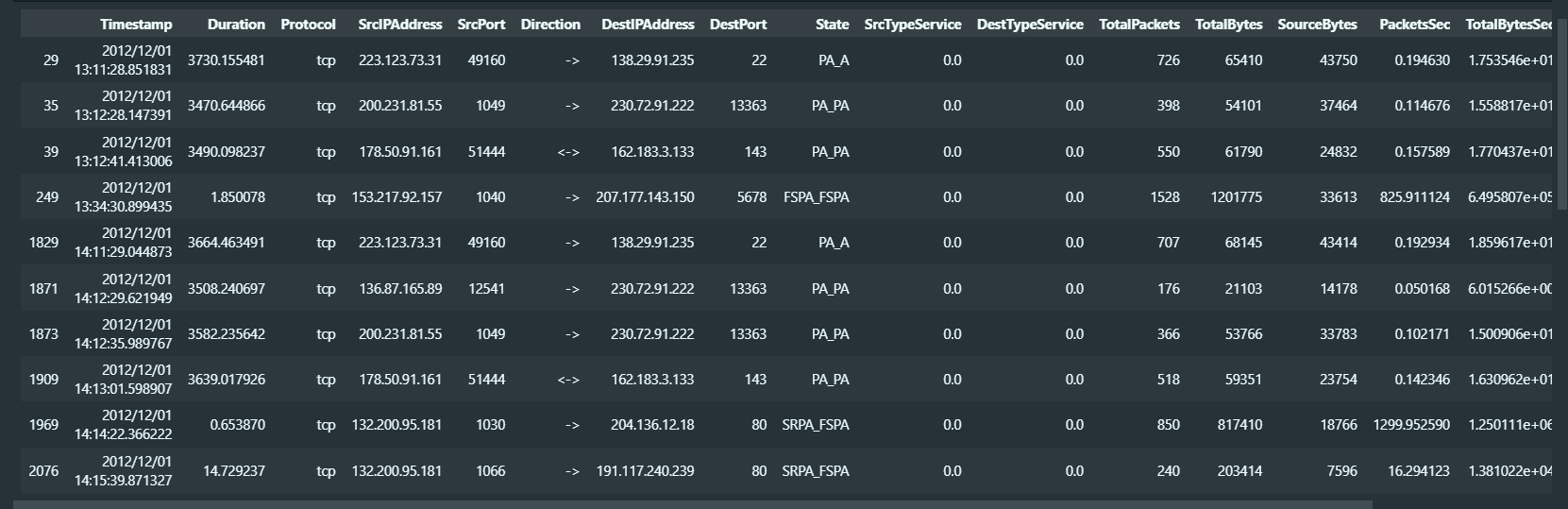


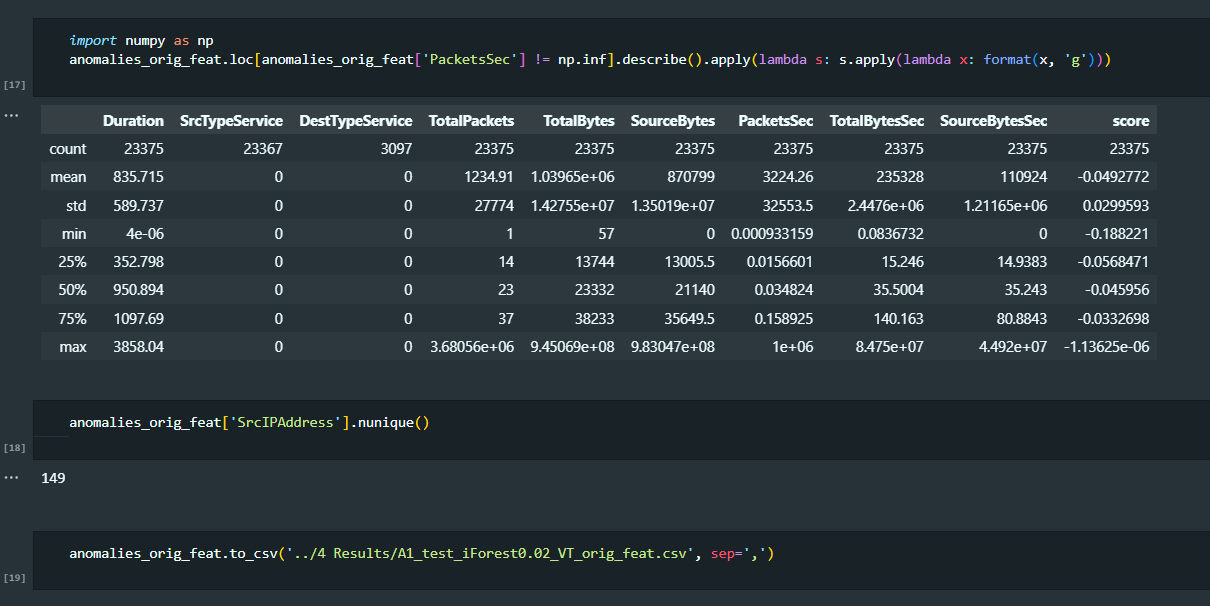


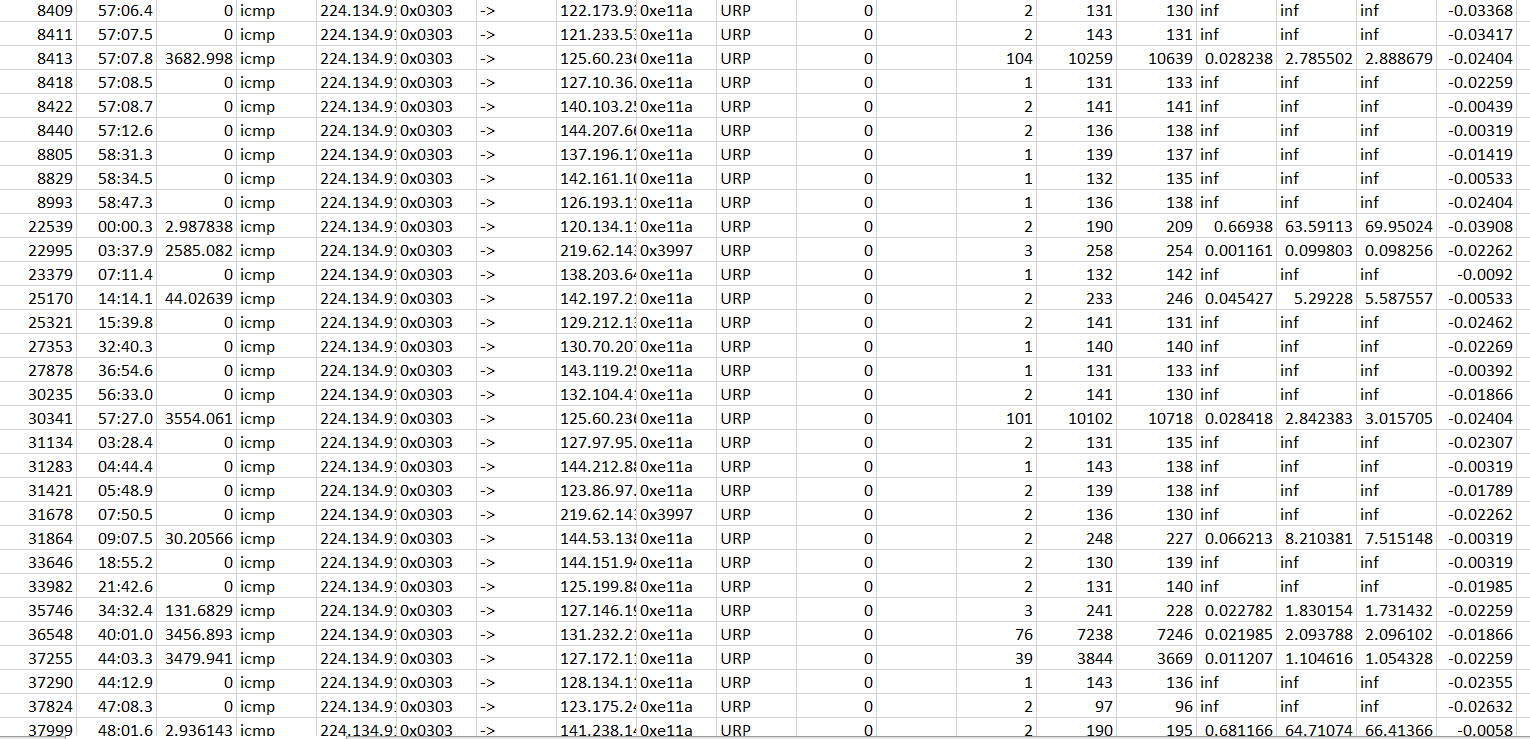


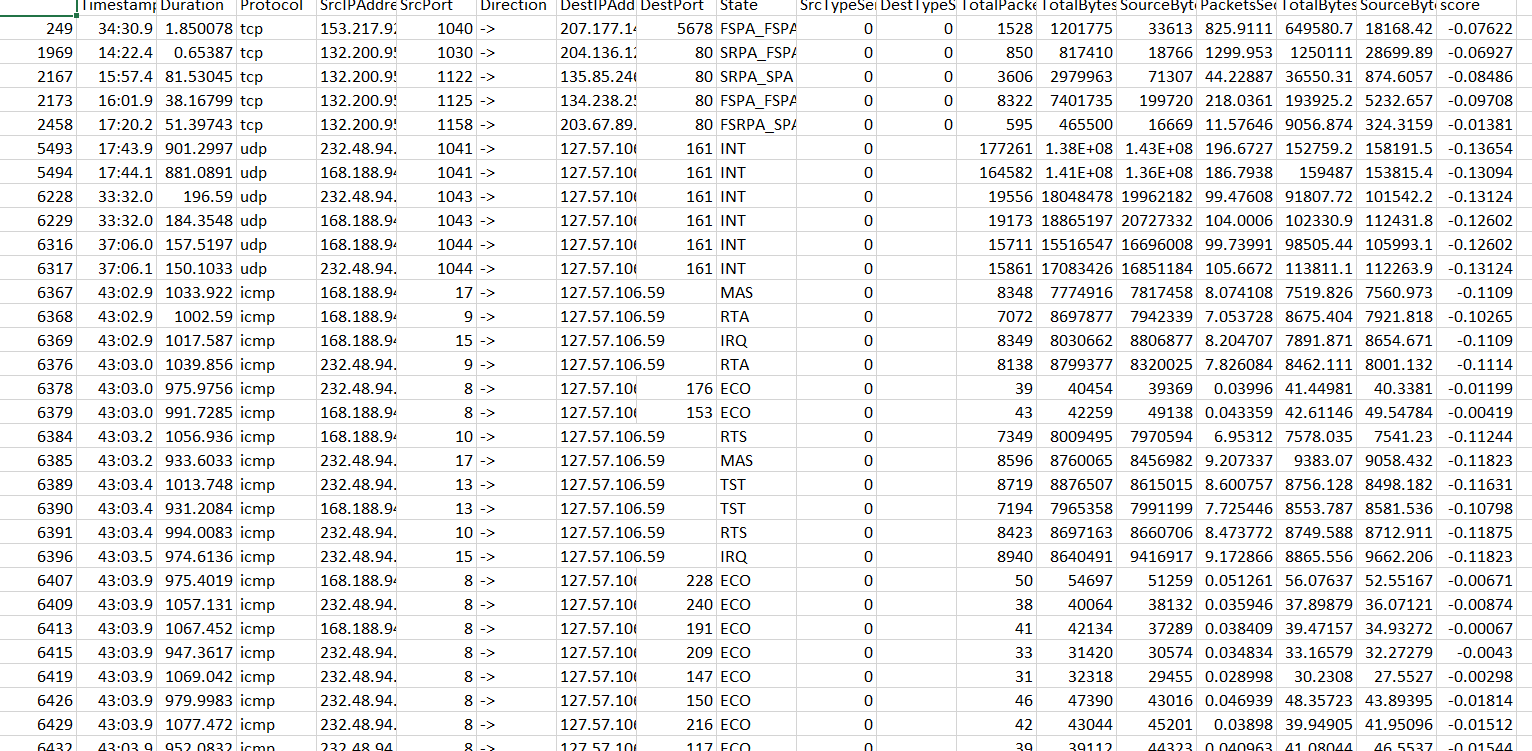


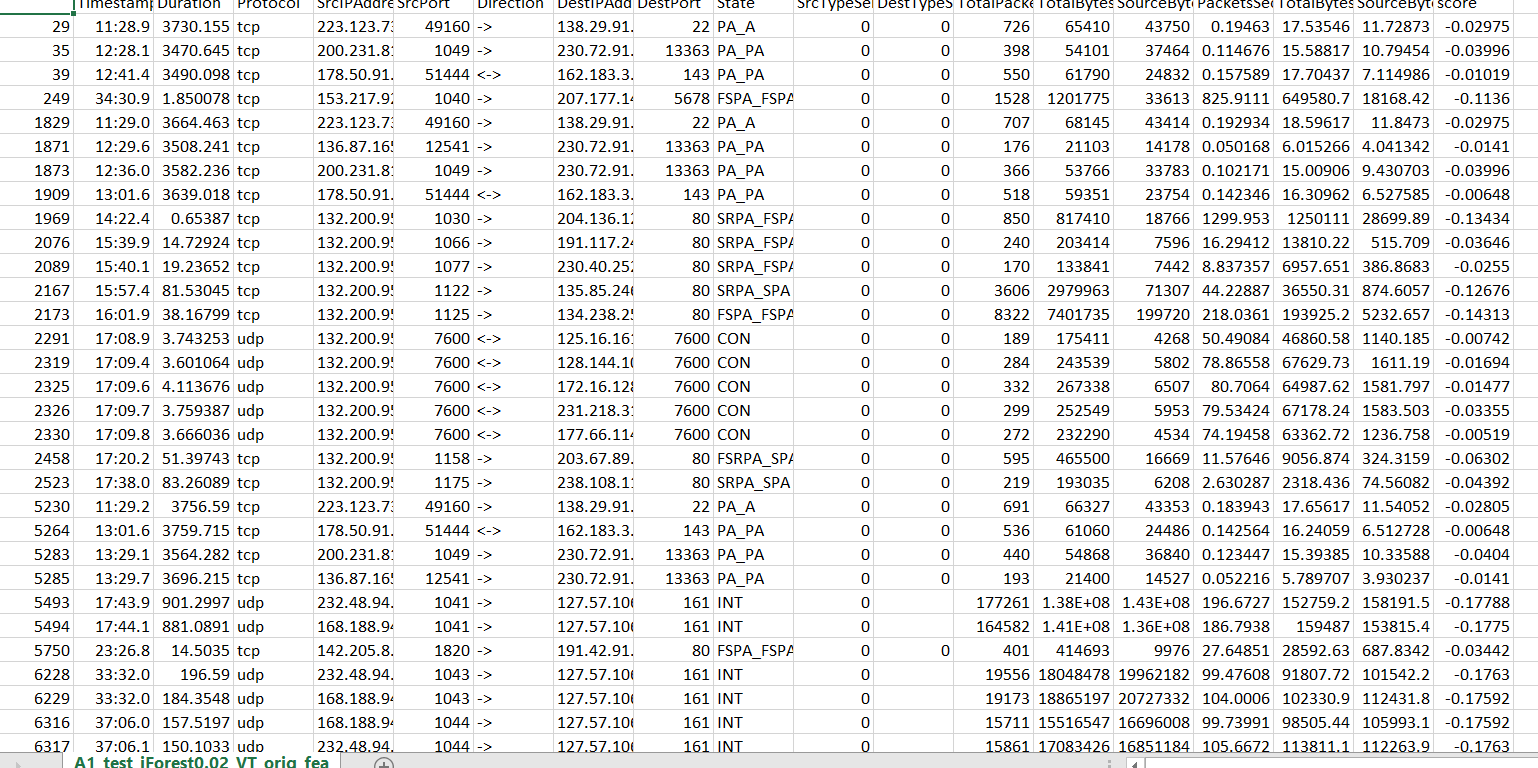


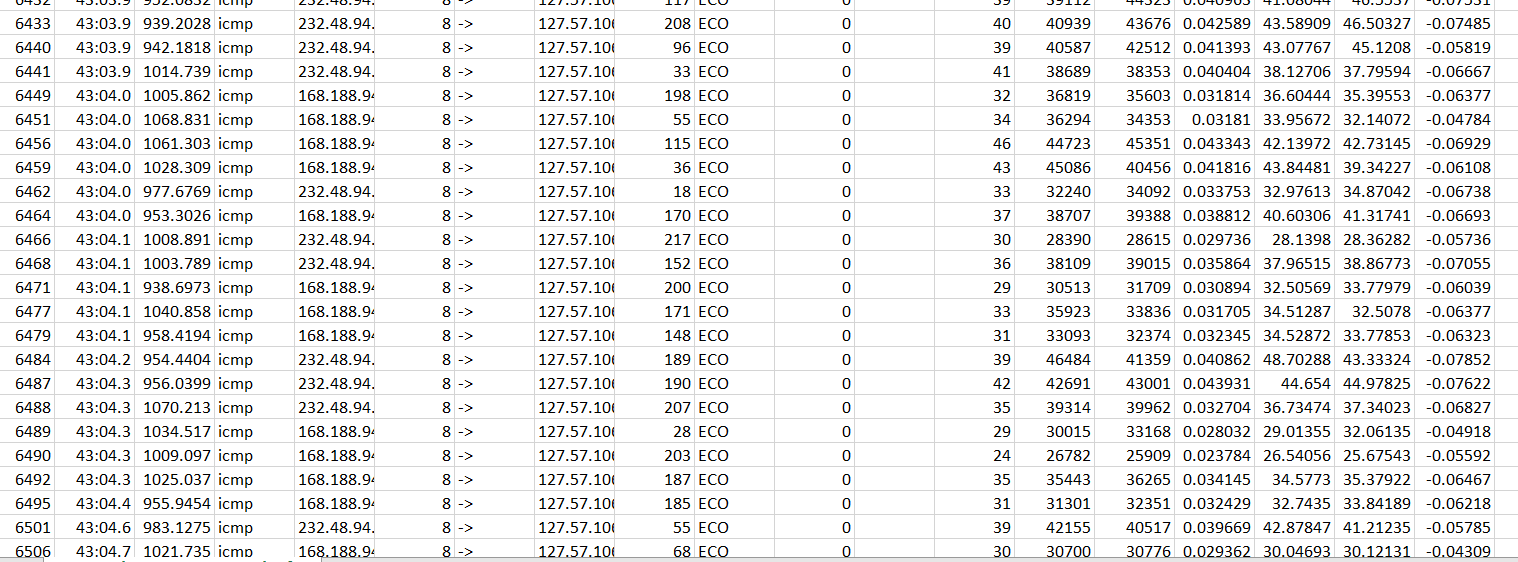


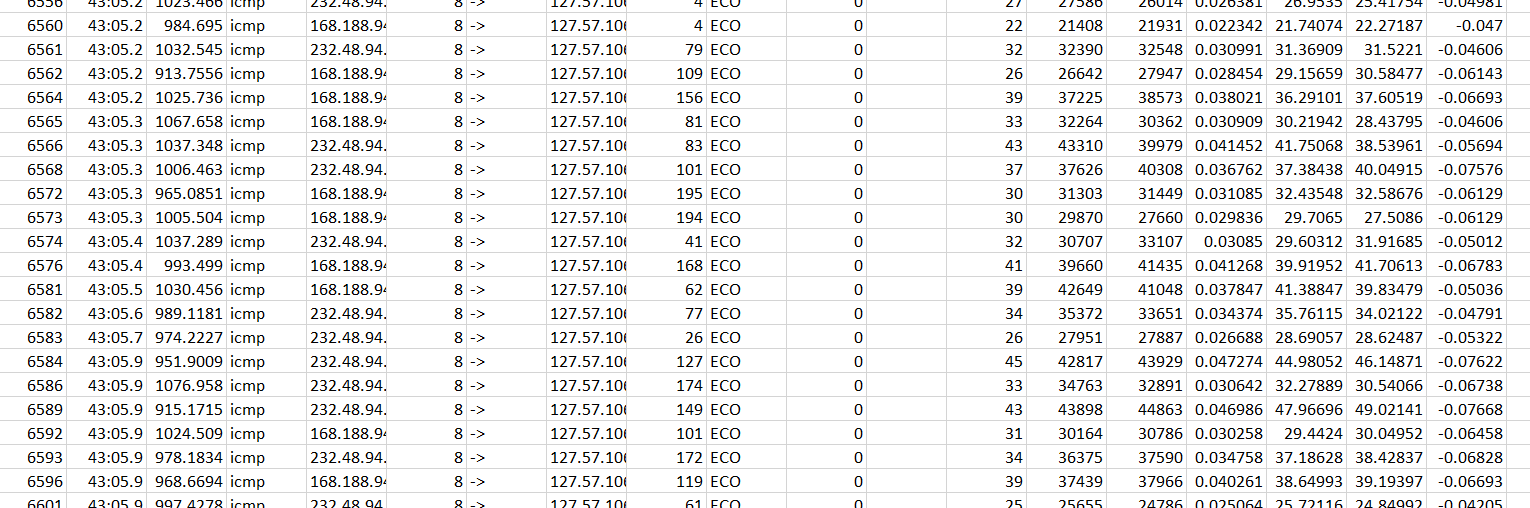


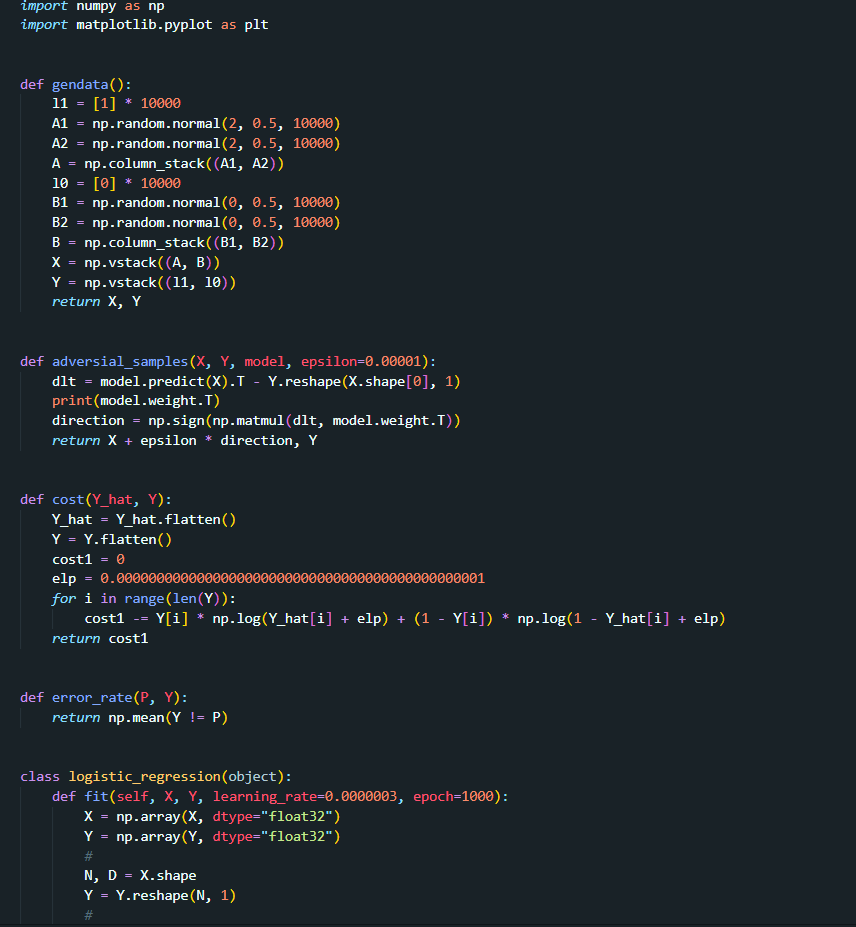




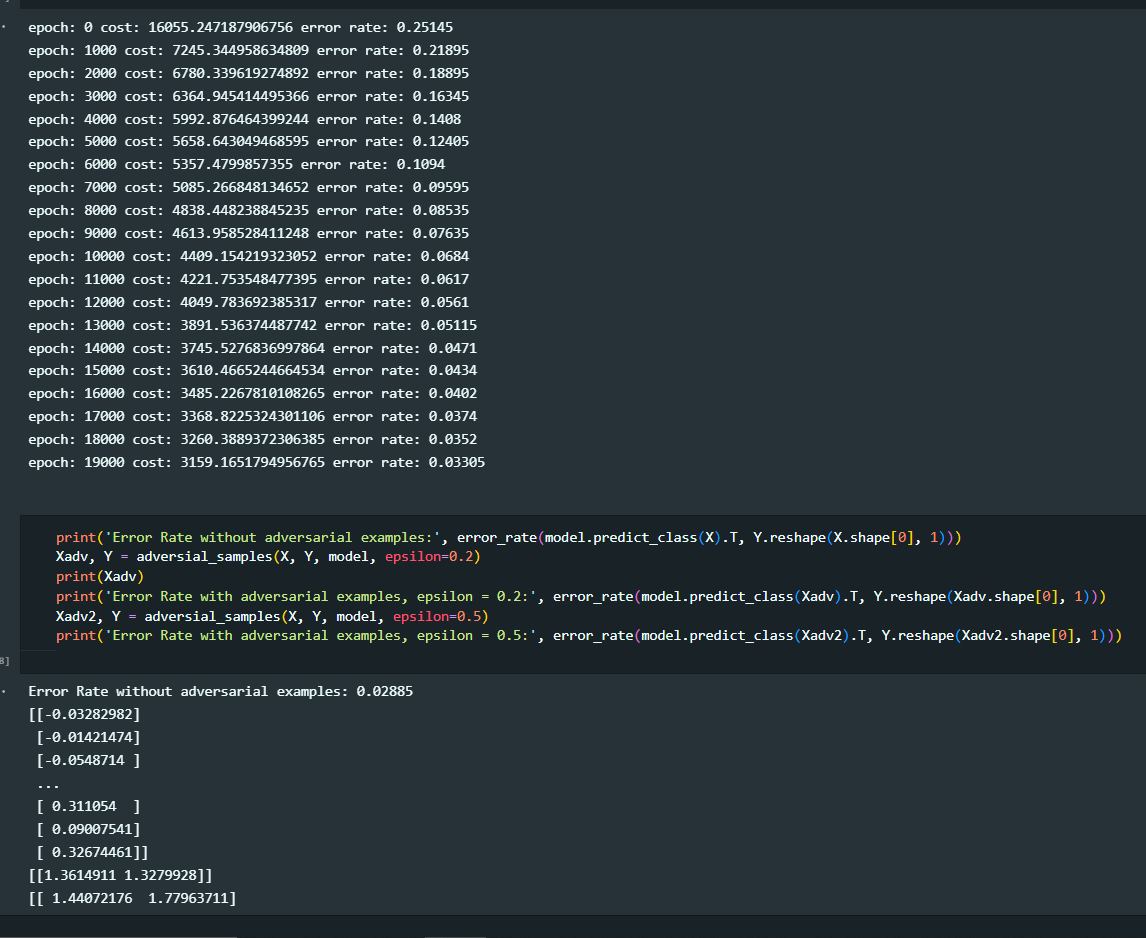


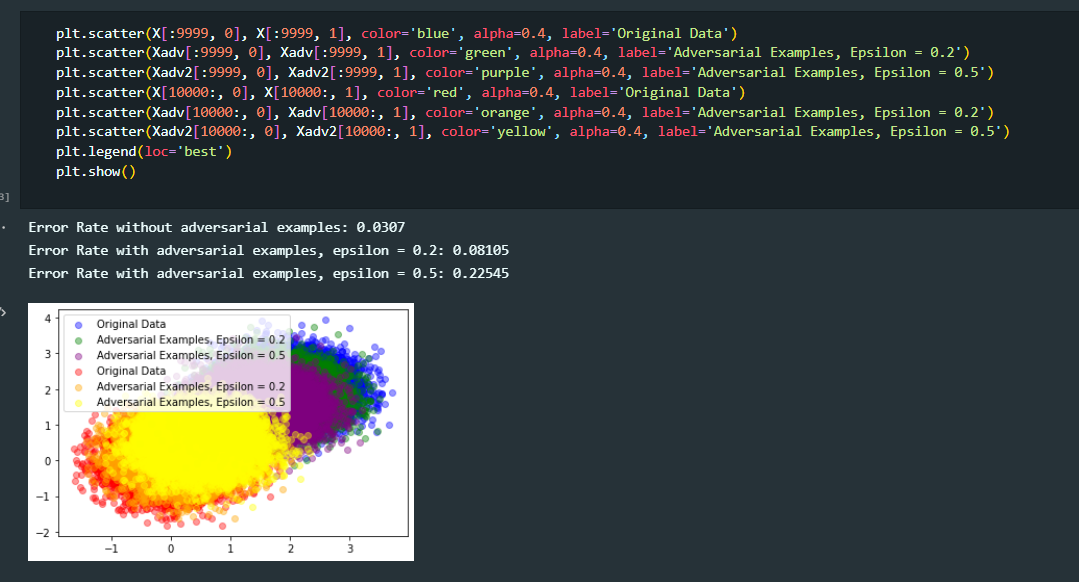












* **Conclusion**
* The method worked best was iForest under Variance Threshold. But not in principle component analysis even with its high dimension reduction, is generating less confident results. The reason for this is that the reduction operation it is not suitable for an anomaly detection task because they tend to absorb the specific anomaly variability. Regarding k-means, the performance tends to be very rigid, means it have same results under PCA and VA. This lack of flexibility difficult to tune the model to address the task.
* Respect to iForest model, this method worked better in this context because allow some grade of flexibility to search the anomalies in the data. Even though, the results seem not to be perfectly accurate, as the set of anomalies detected has different metrics compared to validation set. It is considered that this issue is more related with the feature generation task rather than the model’s performance itself.
* **References**
* Mrs. Ashwini Sheth Research Paper on Cyber Security
* International **Journal** of Information Management Data Insights
* A **comprehensive review** study of cyber-attacks and cyber security;   
  Emerging trends and recent developments.
* **A**rtificial **N**eural **N**etwork for Cybersecurity: A Comprehensive Review
* <https://www.nist.gov/media/0>
* <http://164.100.47.193/Refinput/New_Reference_Notes/English/Cyber-Security-FINAL.pdf>
* <https://www.dhs.gov/>
* <https://www.researchgate.net/publication/260126665_A_Study_Of_Cyber_Security>