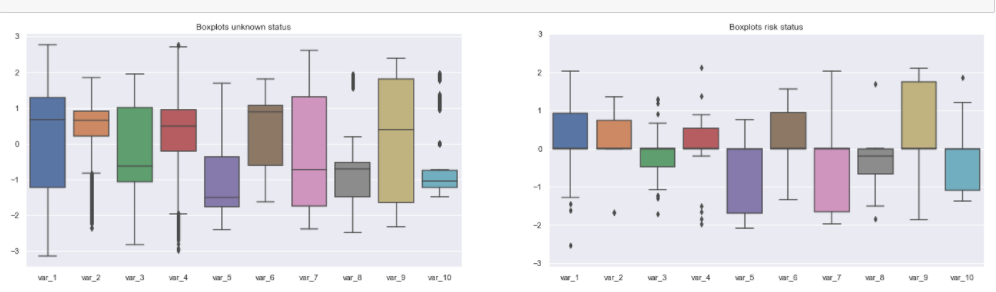
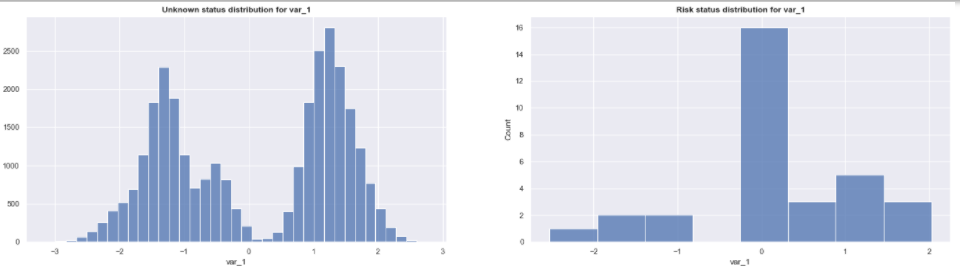
**Case 2 – Identify cyber attacks**

In this case we have a dataset of 30k cyber events and we only know the labels for 32 of those that were identified as cyber attacks. Therefore, we have an anomaly detection problem (unsupervised learning), since we only know a fraction of the labels as anomalous 32/30000 = 0.107%. Furthermore it can be assumed that cyber attacks are no more than 1% of the total data points.

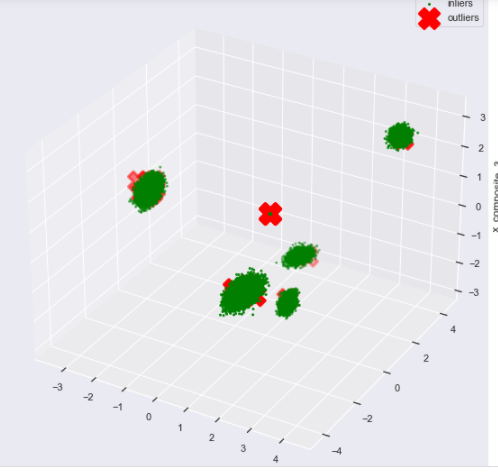
The data was read, checked for null values which the 10 variables/features did not have any. Some exploratory graphs were created to see if the distribution of the variables in the total dataset was different from that of the known 32 anomalous data points.





What could be noticed is that there were more points than normal around 0 for all the variables in the anomalous datapoints. A correlation matrix was also created to see the correlations between features for the unknown status data and anomalous data. A discrepancy between the correlations of var\_2 and var\_4 was identified which was later used to create a feature.

Principal Component Analysis (PCA) was used to visualize the data in the 3D space. Selecting 3 components maintained more than 90% of the variance.



In the image above we can see 5 distinct clusters. There are many outliers gathered at the (0,0,0) point away from the clusters. The rest of the outliers seem to be close to the borders of the 5 clusters

We have an anomaly detection problem since we only know a fraction of the labels as anomalous 32/30000 = 0.107%. The data with unknown status was used for training an Isolation Forest algorithm. The 32 known anomalous events were split between the validation and the test set, 16 each. The metric that was used is recall, since we want to capture as many of the anomalies as possible because they represent cyber attacks and the cost of not diagnosing them is high. The validation set was used for tuning the parameters and the test set is the real world expected performance of the algorithm. Of course our known anomalies are few so the real recall rate may differ from the observed one. I assume that the total number of anomalies does not exceed 1% of the data points.

The Isolation Forest algorithm was used with gridsearch for parameter optimization. The aim was to select the algorithm that finds the most anomalous of the 16 points in the validation set while keeping the number of anomalous predictions low (lowest contamination) since by making more anomalous predictions would cost more investigating time from the corresponding departments and we would have many false positives. In the end the contamination was selected at 1%, in line with our assumption that the total anomalous points do not exceed 1% of the dataset. By checking which points the algorithm classified correctly as anomalous in the validation and test set, we see that it identified correctly the points that had values around 0, which is half of the anomalous points. The rest of the other points were close to the 5 clusters as shown in the PCA making their identification extremely difficult.

As mentioned earlier in the 2 correlation plots, for non-known and anomalous data, what we could observe is that in the case of anomalous data there was lower negative correlation between var\_2 and var\_4 than in the non-anomalous data. Therefore, I engineered an extra feature, trying to give high values to anomalies. For that reason, var\_2 was divided with the square of var\_4 and var\_4 was subsequently removed from the features. This approach enabled the algorithm to identify one extra event as anomalous correctly. However, the threshold of contamination was increased to 3% of the total data.

In general, for the evaluation of the algorithm, it was attempted to have a low enough contamination, so that the algorithm does not flag too many anomalies resulting in low precision and many false positives, and try to identify as many as possible of the known 32 anomalies.

The model is also deployed as an api with Flask and can be run locally via the command

> python app.py

from a terminal. Then it can be accessed in a browser at localhost:5000 or some other configured port. Values can be given for each of the 10 variables and a prediction will be given regarding if the point is predicted as anomalous or not.