Building an algorithm using K-Means and DBSCAN for Anomaly Detection

By:Michael Lively-Scholz

In this program we implemented 2 unsupervised learning techniques to detect anomalies in networks. We used the subsample_95_ormal_5_anomaly.csv dataset to implement the learning methods. The dataset we utilized contains data on network traffic, represented with numerical and categorical features. The dataset also contained a binary label class that indicates normal traffic with the binary representation, “0”, and anomalistic or attack traffic with the binary representation of “1”.

In the algorithm we preprocessed the dataset. We used StandardScaler to standardize the numerical features to ensure we had equal weighting across our dimensions. We also utilized one-hot encoding from the scikit-learn library to encode our categorical features and make them numerical. This gives a complete set of numeric features matrices that are ready to use for the K-Means and DBSCAN unsupervised learning techniques.

When we conducted K-means for anomaly detection we applied K-means clustering with k=2. To conduct anomaly scoring we computed the Euclidean distance from each point to its assigned cluster center. Then we flagged any points with distances above the 95th percentile in their cluster as anomalies. When conducting the DBSCAN learning technique we trained with eps=2 and min-samples = 20. We used DBSCAN to identify points with a label of -1 as anomalies and instructed the model to consider all other points normal.

When evaluated the K-Means unsupervised learning algorithm had the following metrics for the anomaly class, with the binary classification of “1”.

- Precision: 0.64
- Recall: 0.64
- F-1 Score: 0.64

The DBSCAN model obtained these metrics for the anomaly class “1”.

- Precision: 0.12
- Recall: 0.38
- F-1 Score: 0.18

It is very clear that the K-Means learning technique was more effective than DBSCAN at detecting isolated anomalies with this dataset. The higher F-1 and precision scores suggest that it flagged less positives. It had an accuracy score of 0.96, making it a better choice to identify atypical patterns in networks.