Machine Learning

Fall semester 2020-2021

Final project – Task1 – Report

Anastasia Foudouli

Intrusion Detection

Dataset: NSS-KDD

Goal: Use unsupervised techniques to find if an entry is intrusion or normal

Security is an important issue in today's environments.

An intrusion detection system (IDS) is a model that can be used to analyze anomalous behavior in a network. NSL-KDD is the updated version of KDD cup 99 dataset. It is used as a benchmark for researchers to compare different types of Intrusion detection system (IDS) methods or build an intrusion detection system.

Since the test set is labeled and there is approximately the same amount of intrusion and normal entries, we can induce that this also stands for the training set, so as to be representative of the test set as well. Therefore, unsupervised clustering techniques were preferred for this project instead of using unsupervised algorithms for outlier detection such as autoencoders.

Preprocessing

Not much processing was used on the data. The whole dataset was used and data were scaled as z-scores (~N(0,1)).

Modelling

KMeans is one of the best know clustering techniques.

Kmeans algorithm is an iterative algorithm that tries to split the dataset into K pre-determined (user defined) distinct non-overlapping number of clusters (subgroups) where each data point belongs to only one group.

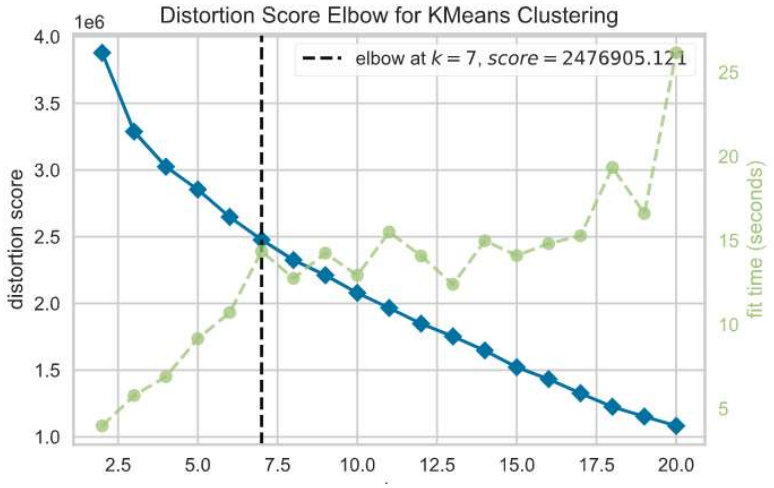
It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible.

It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

Initially a clusterer that would classify the data into two classes (intrusion vs normal) was selected. The accuracy of the clustering was tested on the labeled test set and it achieved 73% accuracy.

However, since it is an unsupervised technique, we usually are not aware of the actual number of classes. There are various techniques that can be used to find the optimal number of clusters K bases on different criterions. In this project the elbow method was used, because other methods such as silhouette score for example are more computationally expensive.

The elbow method is used to determine the number of clusters K. The clustering algorithm is run multiple times, for an increasing number of cluster choice. The clustering score is then plotted as a function of the number of clusters. If the line chart resembles an arm, then the "elbow" (the point of inflection on the curve) is a good indication that the underlying model fits best at that point.



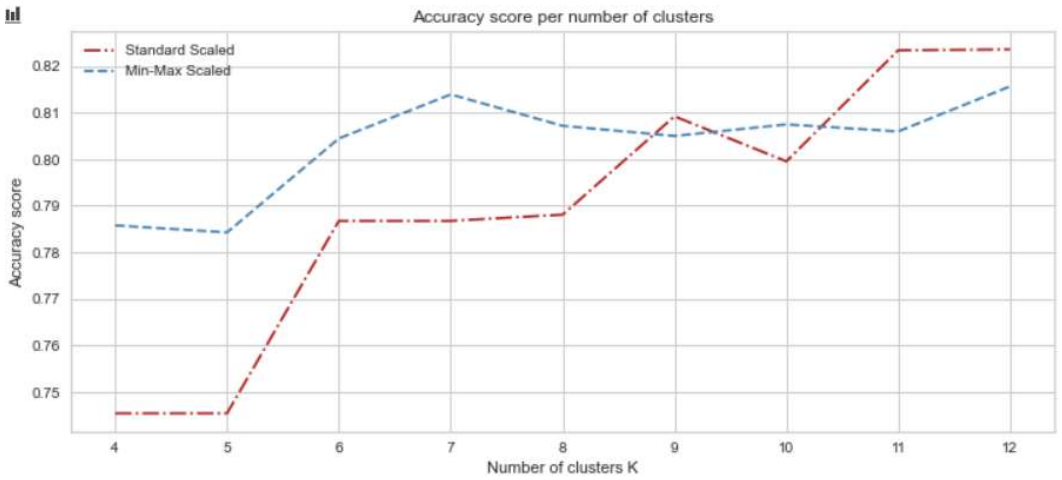
In the graph, 7 is shown as the optimal number of clusters (first break point). However, it is not very clear, as 11 could be considered another good K (based on distortion score).

But, since the test set is labeled, results of clusters around 7 (range 4-12) were plotted against the resulting accuracy on the test set.

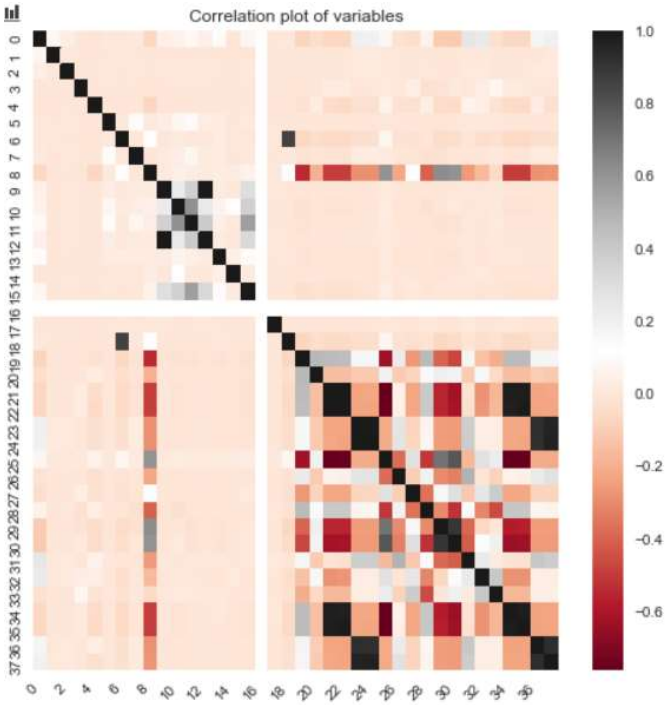
Since the test set has data classified into two classes, normal and attack an approach was needed to be followed on how a cluster is considered to represent the attack or normal class. For the purpose of this project, since I have no domain expertise to be able to interpret the clusters, a cluster is considered to represent the attack class, if most entries of the cluster from the test set are in attack class and normal otherwise.

Different trials and results

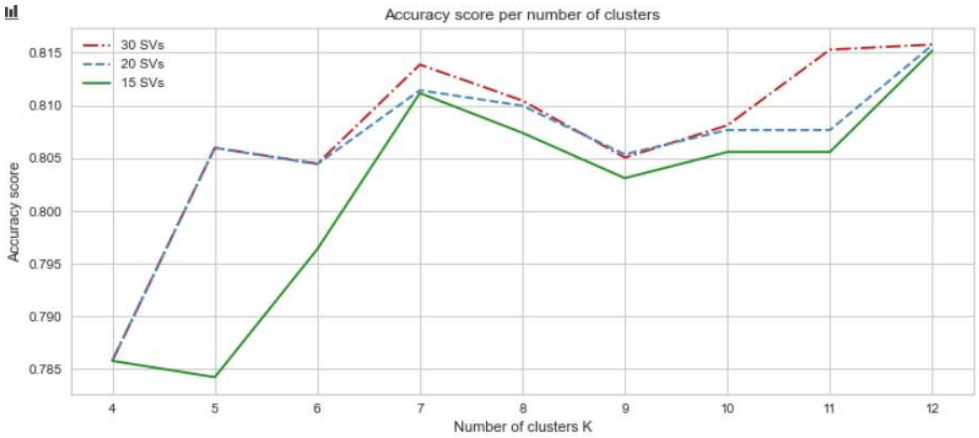
1. Scaling input dataset into [0,1] or transforming it into z-scores. Use of only numeric features.



Since there is collinearity in the features, reducing the dimensions of the input dataset using a dimensionality reduction technique such as PCA or SVD would lead to a new dataset with minimal loss of information.



2. Dimensionality reduction on the numeric features via singular value decomposition.



Although none of the trained models was able to achieve 82% accuracy, the resulting score is still very close.

3. Numeric features scaled into [0,1] and one-hot encoded categorical features.

Here the resulting accuracy did increase over 86%

Since the dataset size is pretty big and there are enough entries to represent both classes, maybe using less data would not lead to a decrease in the resulting scores. Reducing the size of the training data also decreases the training times of the algorithms. The tests were conducted on the best preprocessing of the data, were all variables are used with one hot encoding on the categorical variable set.

4. Using 50% and 25% of the available training data.

Even with 25% of the training set, accuracy of 86% can be reached for 8 number of clusters.

It needs to be noted that KMeans is not a clustering algorithm but rather a partitioning one. More elegant clustering techniques like HDBSCAN are also able to find the optimal number of clusters.

HDBSCAN was used on 25% of the train set and it predicted 180 clusters in the data. However, it did cluster most points as noise.

Not much testing was conducted with this technique as it is very computationally expensive.