ANOMALY DETECTION IN TIME-SERIES DATA

(DATA ANALYTICS - Final Project Report)

Sri Lakshmi Prasanna Dwarampudi
U81031815

INTRODUCTION:

This project's goal is to find irregularities in a time-series dataset. Anomalies in the dataset, which was gathered from a city infrastructure system, correspond to cyberattacks. We need to create an algorithm that can learn some nominal patterns and identify deviations from the nominal patterns (i.e., anomalous instances) in the testing dataset.

The training dataset only contains nominal instances; thus, your training dataset is entirely made up of nominal examples. Both nominal and abnormal cases are present in the testing dataset. The test labels with a value of 0 or 1 denote nominal or anomalous data instances, respectively. You do not require a training labels document because the entire training dataset is nominal (i.e., all training labels are zero).

PERFORMED ANALYSIS:

I trained various algorithms to comprehensive analysis by using multiple algorithms and predicted the test data to evaluate them.

Models used:

Isolation Forest - acc
One-Class SVM
Local Outlier Factor
Auto-Encoder
Variational Auto-Encoder
Gaussian Mixture Model – f1

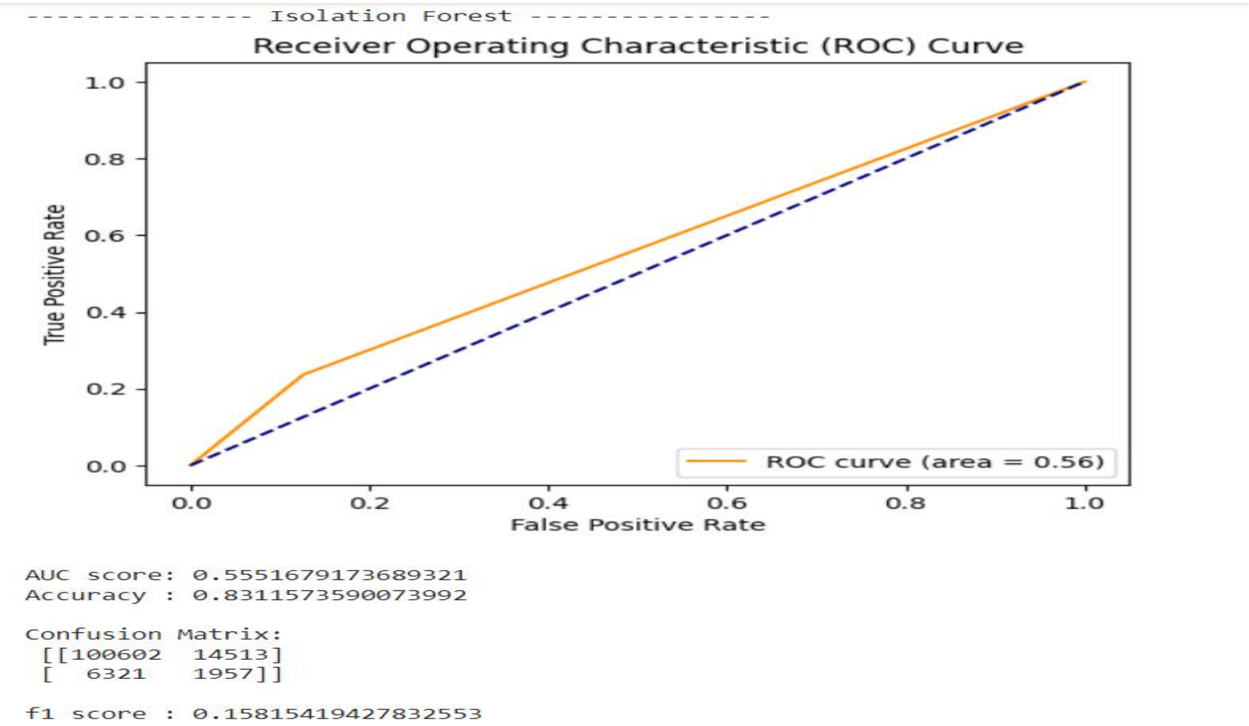
Isolation Forest:

Isolation Forest is an unsupervised machine learning system for finding anomalies. It is based on the idea of isolating anomalies rather than locating normal places.

Isolation Forest's core concept is to use binary trees to distinguish between abnormal and normal points. The method divides the data into subsets repeatedly until the anomalies are isolated in the leaf nodes. The segmentation of the data is based on a randomly chosen feature and a randomly chosen threshold value for that feature.

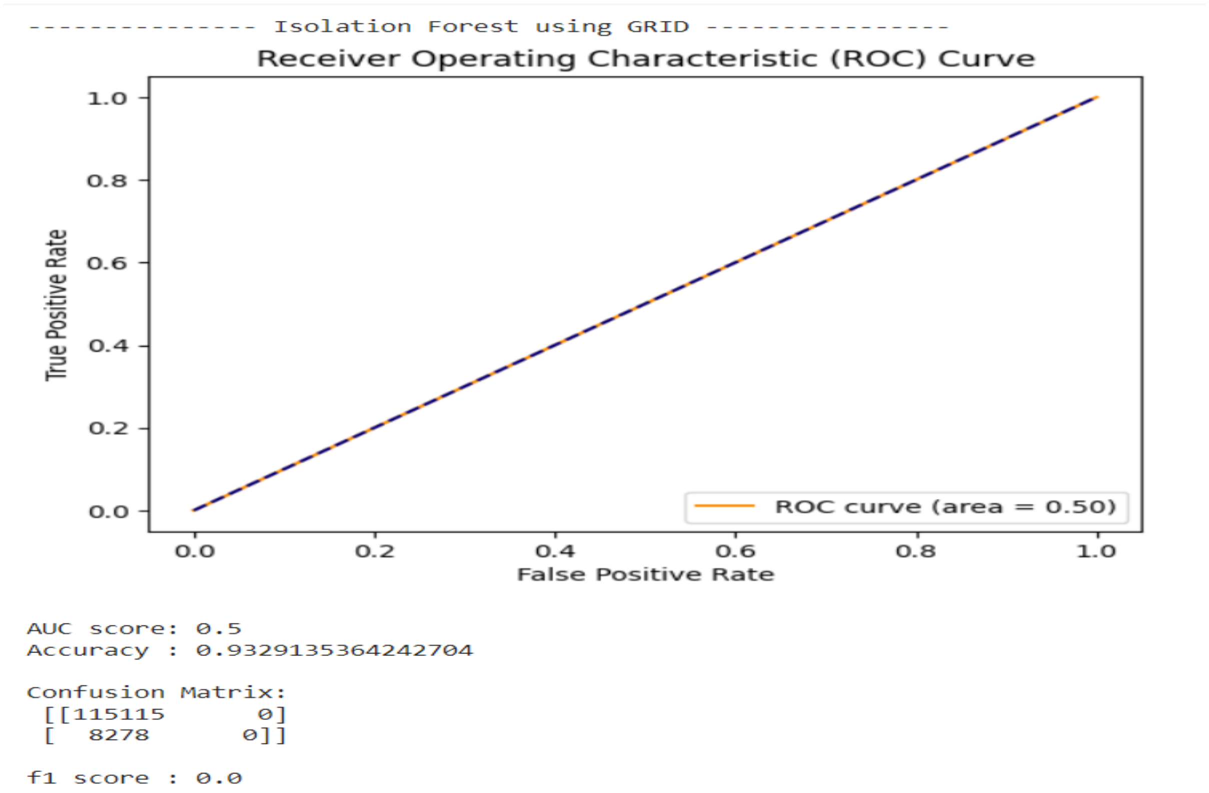
Results:

I used grid search also to train the model with best parameters, but the f1-score is lower than the model with default hyper-parameters as shown in the below screenshots. However, misclassification rate is high and accuracy is also low.



```
--------------- Isolation Forest ----------------
```

```
AUC score: 0.5551679173689321
Accuracy : 0.8311573590073992

Confusion Matrix:
 [[100602  14513]
 [   6321   1957]]

f1 score : 0.15815419427832553
```



```
--------------- Isolation Forest using GRID ----------------
```

```
AUC score: 0.5
Accuracy : 0.9329135364242704

Confusion Matrix:
 [[115115       0]
 [   8278       0]]

f1 score : 0.0
```

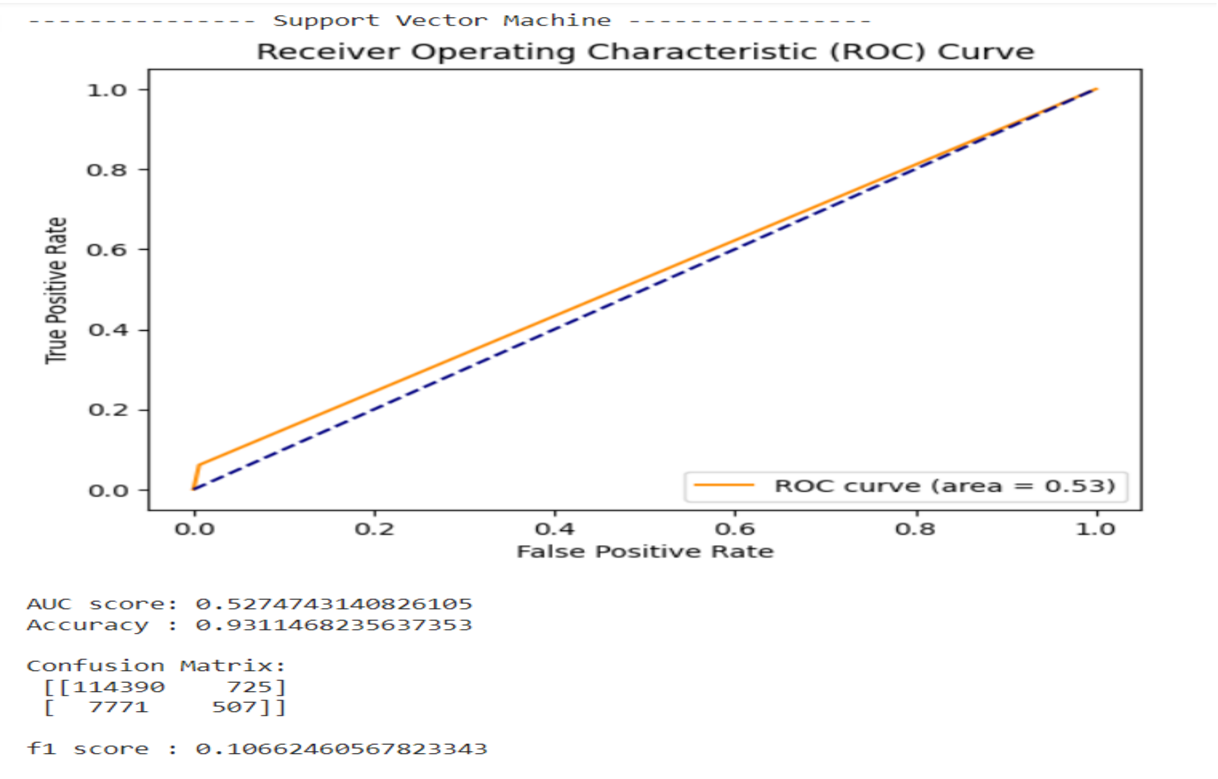
One-Class SVM:

OneClass SVM is an unsupervised machine learning technique for anomaly detection. It is based on the idea of finding a hyperplane in a high-dimensional space that divides regular points from anomalies.

The method involves fitting a hyperplane to the data in a way that maximizes the separation between it and the nearest data points (support vectors). The normal data points and the anomalous ones are divided using this hyperplane. The OneClass SVM algorithm is comparable to the conventional SVM algorithm with the exception that it only utilizes one class of data points for training.

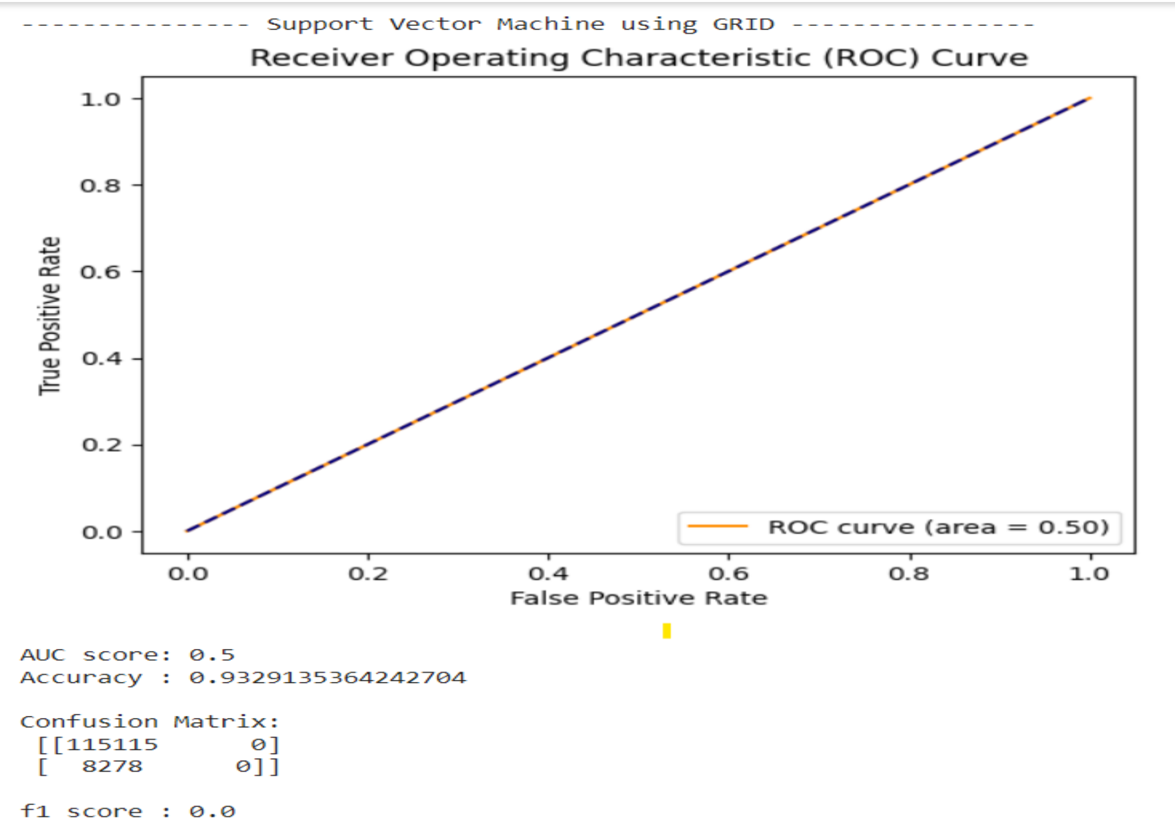
Results:

I used grid search also to train the model with best parameters, but the f1-score is lower than the model with default hyper-parameters as shown in the below screenshots. However, misclassification rate is high and accuracy is also low.



```
--------------- Support Vector Machine ----------------
```

```
AUC score: 0.5274743140826105
Accuracy : 0.9311468235637353

Confusion Matrix:
 [[114390    725]
 [  7771    507]]

f1 score : 0.10662460567823343
```



```
--------------- Support Vector Machine using GRID ----------------
```

```
AUC score: 0.5
Accuracy : 0.9329135364242704

Confusion Matrix:
 [[115115      0]
 [  8278      0]]

f1 score : 0.0
```

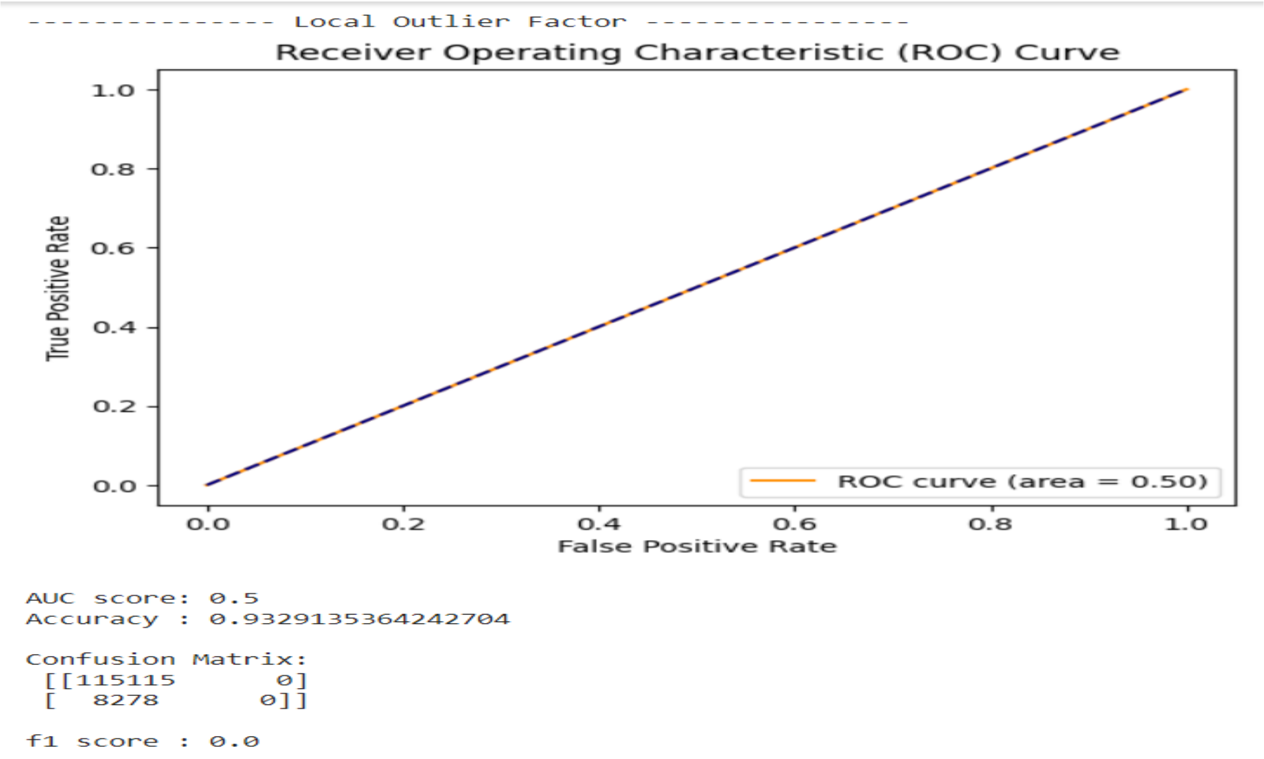
Local Outlier Factor:

A machine learning approach called Local Outlier Factor (LOF) for detecting anomalies is unsupervised. The idea behind it, which was first put forth by Breunig et al. in 2000, is to locate data points that differ noticeably from their immediate surroundings.

By calculating the density of the nearby data points around each position, the algorithm operates. An anomaly is most likely a site that has a denser population than the surrounding areas. The ratio between the average densities of a data point's k-nearest neighbors and its own density is known as the LOF score. If the point has a low LOF score, it is likely to be comparable to its neighbors, but a high LOF value suggests a major departure from those neighbors and is likely to be an anomaly.

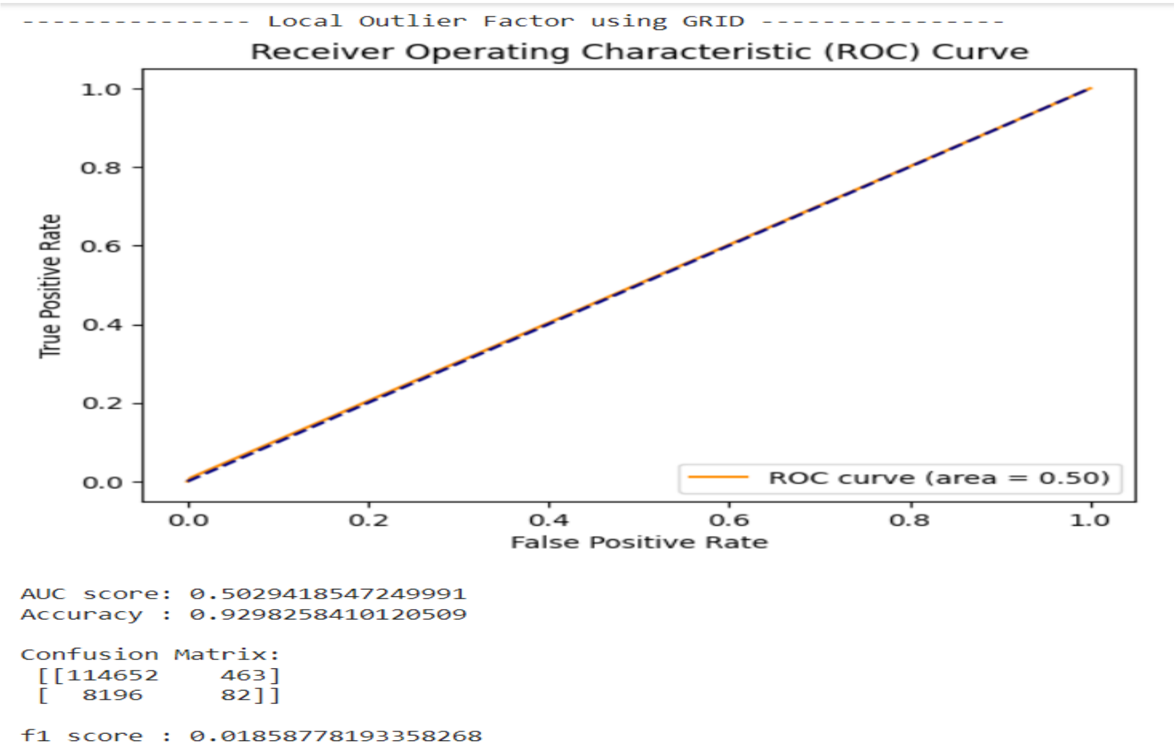
Results:

I used grid search also to train the model with best parameters, but the f1-score is lower than the model with default hyper-parameters as shown in the below screenshots. However, misclassification rate is high and accuracy is also low.



```
--------------- Local Outlier Factor ----------------
```

```
AUC score: 0.5
Accuracy : 0.9329135364242704

Confusion Matrix:
 [[115115      0]
 [  8278      0]]

f1 score : 0.0
```



```
--------------- Local Outlier Factor using GRID ----------------
```

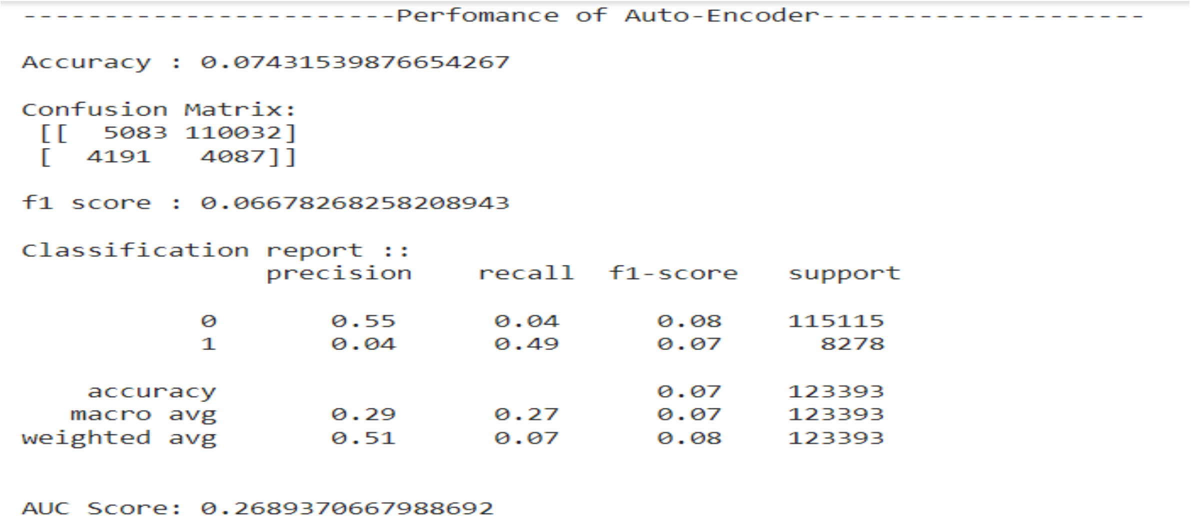
```
AUC score: 0.5029418547249991
Accuracy : 0.9298258410120509

Confusion Matrix:
 [[114652    463]
 [  8196      82]]

f1 score : 0.01858778193358268
```

Auto-Encoder:

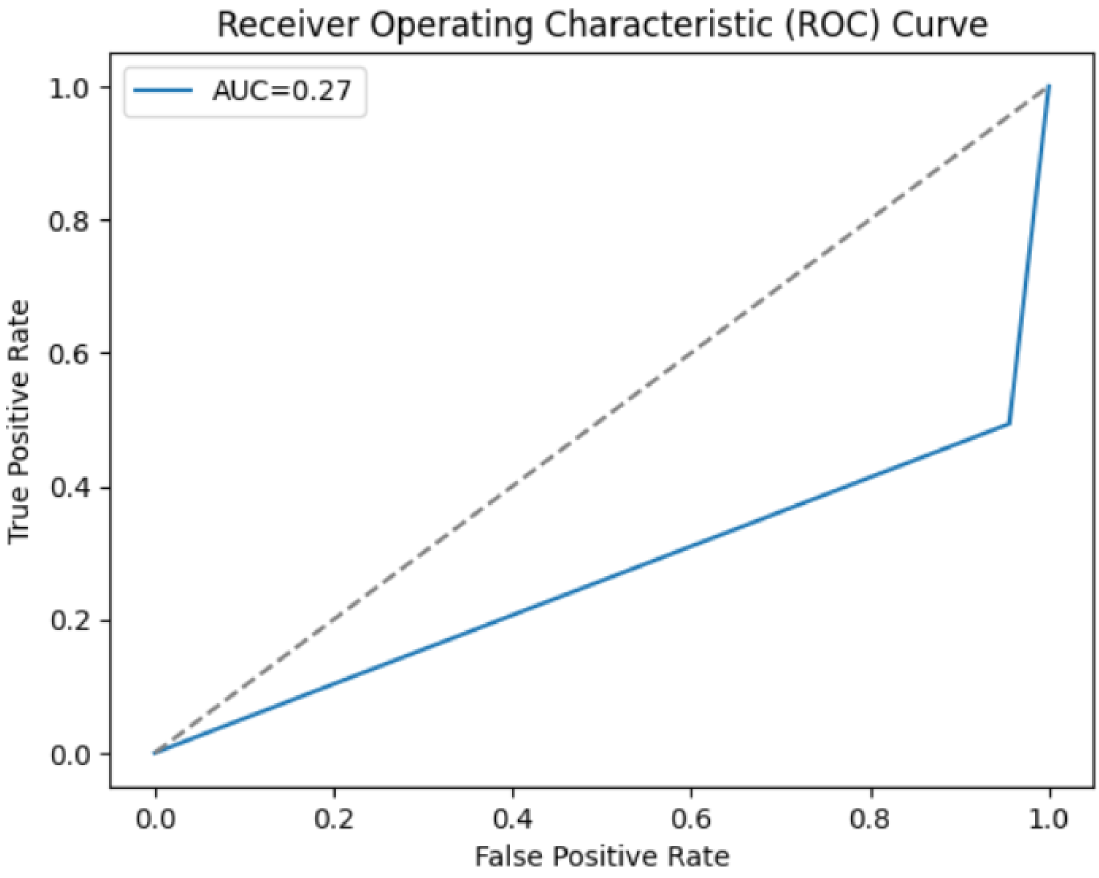
A popular neural network design for feature extraction and unsupervised learning is the auto-encoder. The fundamental concept underlying auto-encoders is to train a neural network to reconstruct the original input from a lower-dimensional representation in order to learn a compressed version of the input data.

Encoder and decoder are the two components that make up an auto-encoder. The input data is mapped by the encoder into a lower-dimensional representation, which is then mapped by the decoder back into the original data space. Reconstruction error, or the difference between the original input and the decoder's output, is what we want to reduce when we train an auto-encoder.

Results:



```
-----------------------Perfomance of Auto-Encoder--------------------

Accuracy : 0.07431539876654267

Confusion Matrix:
 [[  5083 110032]
 [  4191   4087]]

f1 score : 0.06678268258208943

Classification report ::
               precision    recall  f1-score   support

           0       0.55      0.04      0.08    115115
           1       0.04      0.49      0.07      8278

    accuracy                           0.07    123393
   macro avg       0.29      0.27      0.07    123393
weighted avg       0.51      0.07      0.08    123393


AUC Score: 0.2689370667988692
```

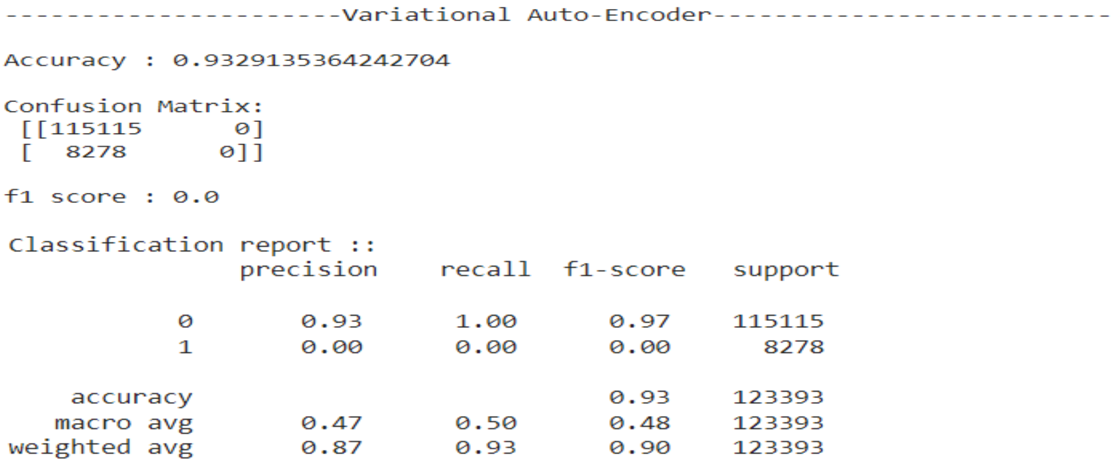


Variational Auto-Encoder:

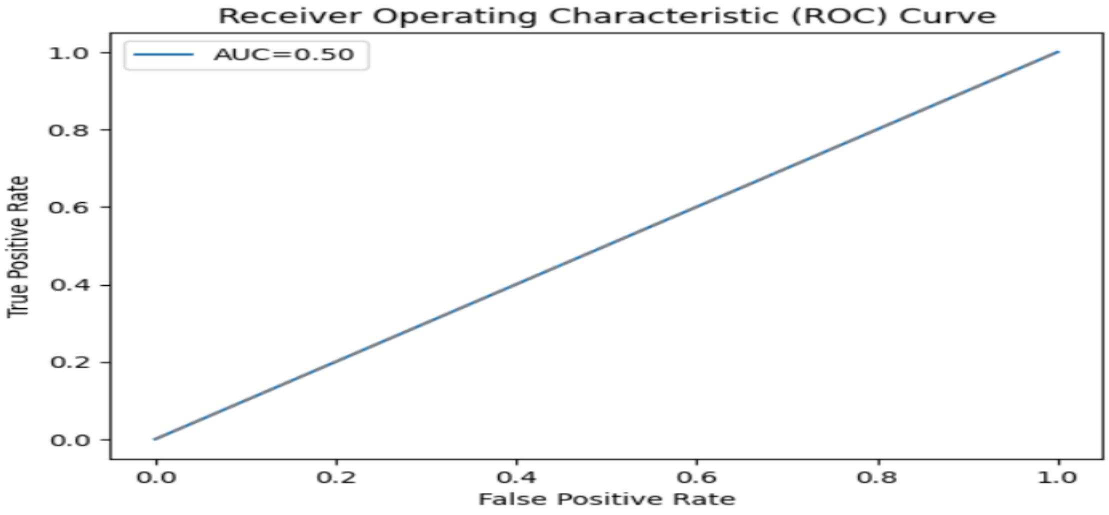
Variational Auto-Encoders (VAE), a subset of auto-encoders, have a probabilistic meaning. A VAE is composed of an encoder and a decoder, just like a regular auto-encoder, but it also learns a compressed representation of the input data that adheres to a particular probability distribution, often a Gaussian distribution.

Within a VAE, the encoder converts the input data into two vectors: the mean and variance of the latent space distribution. The decoder then relates the latent space to the initial input data. The VAE learns to minimize the difference between the learnt latent space distribution and a prior distribution, which is commonly a standard Gaussian distribution, and the reconstruction error during training.

Results:



```
----------------------Variational Auto-Encoder--------------------------

Accuracy : 0.9329135364242704

Confusion Matrix:
 [[115115      0]
 [  8278      0]]

f1 score : 0.0

 Classification report ::
               precision    recall  f1-score   support

           0       0.93      1.00      0.97    115115
           1       0.00      0.00      0.00      8278

    accuracy                           0.93    123393
   macro avg       0.47      0.50      0.48    123393
weighted avg       0.87      0.93      0.90    123393
```

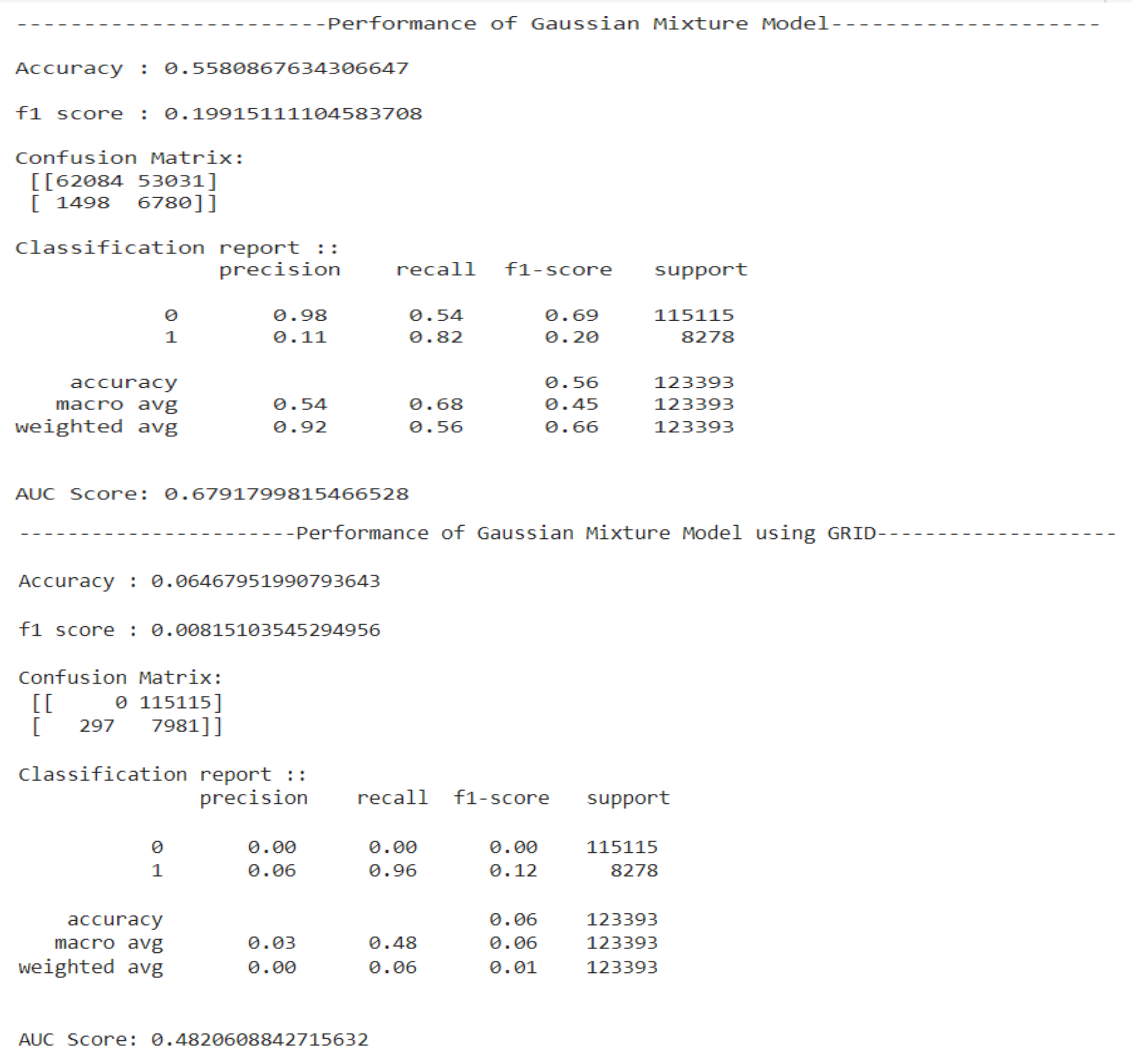


Gaussian Mixture Model:

An algorithm for clustering data in machine learning is called a Gaussian mixture model (GMM). It is a form of unsupervised learning technique that assumes the data is produced from a combination of several Gaussian distributions, each with its own mean and covariance matrix. The aim of the GMM is to estimate the parameters of these underlying Gaussian distributions as well as the likelihood that each data point belongs to each distribution.

Results:

I used grid search also to train the model with best parameters, but the f1-score is lower than the model with default hyper-parameters as shown in the below screenshots. However, misclassification rate is high and accuracy is also low.



```
-----------------------Performance of Gaussian Mixture Model--------------------

Accuracy : 0.5580867634306647

f1 score : 0.19915111104583708

Confusion Matrix:
 [[62084 53031]
 [ 1498  6780]]

Classification report ::
               precision    recall  f1-score   support

           0       0.98      0.54      0.69    115115
           1       0.11      0.82      0.20      8278

    accuracy                           0.56    123393
   macro avg       0.54      0.68      0.45    123393
weighted avg       0.92      0.56      0.66    123393


AUC Score: 0.6791799815466528
-----------------------Performance of Gaussian Mixture Model using GRID--------------------

Accuracy : 0.06467951990793643

f1 score : 0.00815103545294956

Confusion Matrix:
 [[     0 115115]
 [   297   7981]]

Classification report ::
               precision    recall  f1-score   support

           0       0.00      0.00      0.00    115115
           1       0.06      0.96      0.12      8278

    accuracy                           0.06    123393
   macro avg       0.03      0.48      0.06    123393
weighted avg       0.00      0.06      0.01    123393


AUC Score: 0.4820608842715632
```

RESULTS AND DISCUSSIONS:

As per the results of various models shown in the above screenshots and after comparing them with each other for comprehensive analysis.

Isolation Forest:

The model's AUC score is 0.555, which indicates that it performs somewhat better than a random guess. The model's accuracy is 0.831, meaning that it correctly categorizes 83.1% of the data points. However, the f1 score is low (0.158), indicating poor performance in identifying true anomalies. The confusion matrix demonstrates that the model properly recognizes 100602 normal data points and 1957 anomalies, but wrongly recognizes 6321 normal data points as anomalies and misses 14513 actual anomalies. The classification report reveals that the model has weak precision and recall for anomalies, with a precision of only 0.12 and a recall of 0.24. The weighted average f1 score is 0.86, suggesting high performance on the typical data points but low performance on the anomalies.

Isolation Forest using GRID Search:

It is clear from the study given that the isolation forest's AUC score is 0.5551679173689321, which is comparatively low. This demonstrates that the Isolation Forest struggles to discern between typical and unusual occurrences. The Isolation Forest is able to accurately identify 83.12% of cases as either normal or anomalous, according to the accuracy score of 0.8311573590073992.

One-Class SVM:

The low AUC score, low f1 score, and low recall for the anomalous class demonstrate the poor performance of the One-class SVM algorithm on this dataset. This shows that other anomaly detection techniques might work better, and that the algorithm might not be appropriate for this specific dataset.

Local Outlier Factor:

The LOF method performed poorly in locating outliers, with an AUC score of 0.5 and an accuracy of 0.9329. The confusion matrix demonstrates that the algorithm correctly categorised every event as negative and failed to identify any genuine positives. As a result, the f1 score is 0.0 and the classification report indicates subpar performance. In conclusion, LOF struggled to find outliers in this specific dataset.

Auto-Encoder:

In comparison to the other models, the Auto-Encoder model performs very poorly. It only has an F1 score of 0.067 and an accuracy of 0.074. The confusion matrix demonstrates that the model consistently predicts fewer instances of the minority class (1) and more instances of the majority class (0). The minority class's extremely low recall score illustrates this.

The model cannot efficiently distinguish between the two groups, as shown by the AUC value of 0.269. Overall, it appears that the Auto-Encoder model is not a suitable match for this anomaly detection task.

Variational Auto-Encoder:

With an accuracy of 0.93 and an AUC score of 0.5, the Variational Autoencoder performed similarly to the Local Outlier Factor. Once more, the model was unable to identify any anomalies, and as a result, the classification report's precision, recall, and F1-score for the anomalous class are all zero. Its F1 score, however, is still somewhat low, indicating that it would have trouble accurately selecting the positive class.

Gaussian Mixture Model:

According to the GMM model's accuracy score of 0.558, 55.8% of the cases in the dataset are properly classified. With recall being much greater than accuracy, the model's precision and recall appear to be out of balance, according to the f1 score of 0.199. The model's moderate ability to distinguish between legitimate and fraudulent transactions is indicated by the AUC score of 0.679.

CONCLUSION:

Overall, based on the given outputs, the GMM seems to be the best performing model among the three in terms of identifying fraudulent transactions, as it has the highest f1 score and AUC score. However, it should be noted that these results are based on a single evaluation and may not be representative of the overall performance of the models.

However, in terms of precision and recall, Isolation Forest has higher precision than the Gaussian Mixture Model and One-Class SVM but lower than the Variational Auto-Encoder. However, its recall is lower than all models.

Overall, the Isolation Forest model seems to have moderate performance compared to the other models, with lower AUC and f1-score, but higher accuracy and precision.