Artifical Intelligence In industry
Project Report

Anomaly detection using Unsupervised learning

Jyoti Yadav

March 2022

Contents

1 Introduction

1.1 What is Anomaly Detection?

Anomaly detection is about finding the odd one out where we analyze the data patterns in a dataset to find data points that are outliers and do not comply with the normal data patterns of the dataset.

1.2 what is semi-Supervised Anomaly Detection?

They are also referred to as Novelty Detection Algorithms. Outliers that do not pollute the training data and the anomaly detection algorithm only detects if the new observation is an inlier or an outlier.

1.3 Problem Statement

Anomaly detection in supercomputers is a very difficult problem due to big scale of the systems and high number of components. **High Performance Computing (HPC)** systems are complex machines with many components that must operate concurrently at the best of their theoretical performance.In reality many factors can degrade the performance of a HPC system. hardware can break the applications may enter undesirable and unexpected states, components can be wrongly configured.A critical aspect of modern and future supercomputers is the capability of detecting faulty conditions stemming from the improper behaviour of one or multiple parts. This issue is revelant not only for scientific computing systems but also in data centers and clouds providers whose business strongly relies on the availability of their web services. For instance ,Amazon in 2016 would have lost 15M dollar for just an hour of out of service (Hennessy and Patterson 2011). An automated process for anomaly detection would be a great improvement for current HPC systems, and it will probably be a necessity for future Exascale supercomputers.

Nowadays,monitoring infrastructures are available in many HPC systems and data centers, used to gather data about the state of the systems and their components thanks to a large variety of measurement sensors. Given the deluge of data originating from a monitoring framework, real time identification of problems and undesired situations is a daunting task for system administrators. The growing scale of HPC systems will only make this task even more difficult.

The key idea here is to use autoencoders which will learn the normal (healthy) behaviour of the supercomputer nodes and after training, use them to identify the abnormal conditions.

2 Dataset

The HPC dataset was used to perform anomaly detection.The data was collected from a monitored supercomputer hosted at CINECA and called "Marconi100".The data was collected with a tool called Examon, developed together with other Unibo colleagues. The dataset is composed of several folders, a folder for each selected node (there are not all the hundreds of nodes present on Marconi100, but some nodes with periods that also contained failures). The information monitored on Marconi100's nodes is varied, ranging from the load of the different cores, to the temperature of the room where the nodes are located, the speed of the fans, details on memory accesses in writing / reading, etc.

The column called "New Label" column indicates the presence or absence of a failure on the node.I have dropped the timestamp Label column from the dataset.

3 Download Dataset

The dataset composed of several files more than 150+ and each dataset correspond to one node.At first I also noticed the file was in gzip format it has to be unzip it. I tried several ways to unzip it and also read through several pandas functions.After reading about several file formats i came across the file format parquet. At first I tried to read single file using pandas read_parquet function.Then i read how to read multiple files using read_parquet function.

Parquet is a columnar storage format that supports nested data.It is designed to support fast data processing for complex data with several notable characteristics.

Columnar: Unlike row-based formats such as CSV or Avro, Apache Parquet is column-oriented – meaning the values of each table column are stored next to each other, rather than those of each record.

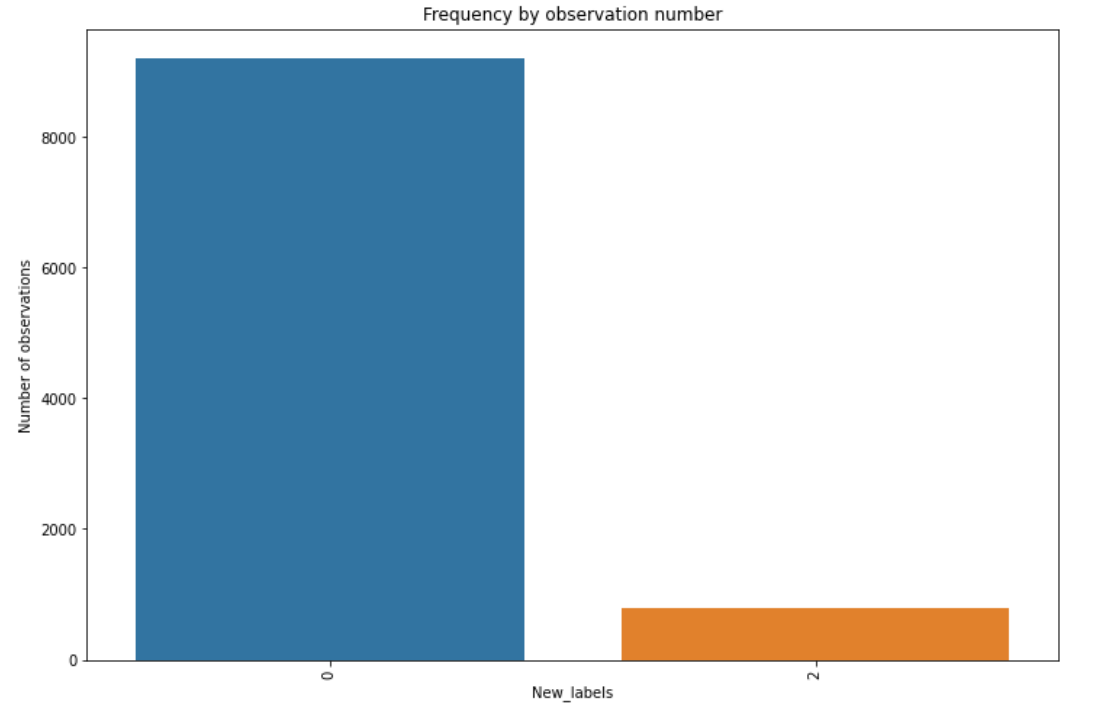
Open-source: Parquet is free to use and open source under the Apache Hadoop license and is compatible with most Hadoop data processing frameworks.

Self-describing: In addition to data, a Parquet file contains metadata including schema and structure.

4 Data Analysis

After reading the dataset let's look at the dataset.

1. The shape of the dataset.
2. Finding the missing or null values.
3. Finding the unique number of labels.
4. Finding the count and percentage of each New_label.



As we can see from the above the sample points for class 1 is around 96% and for the class 2 it was 4%.It was observed the dataset was imbalanced.

5 Train Test Split

We divide the dataset into training and testing with the help of function called train_test_split and we passed the parameters as test size of 0.2 and random_state for the reproducibility of the results.

80% - Training
20% - Testing

We have to train the autoencoder only with the normal behaviour of the classes and we do not inject any anomalies at the time of training.If our model can learn the correlations between the set of features that describe the state of a supercomputer then it can consequently notice changes in the correlations that indicate an abnormal state.After the training phase our model is capable of detecting the anomalies that have never been seen before.

6 MinMax Scaling

I have Normalize the dataset as the features vary with different scales.Therefore, in order for Deep learning models to interpret these features on the same scale, we need to perform feature scaling. MinMaxScaler has managed to rescale those features so that their values are bounded between 0 and 1.

Fit the normalizer to the training set (X_train) using the fit_transform() function and then apply the scale to the test set(X_test)and also to the y_test.

7 Model

AutoEncoder is a semi supervised deep learning algorithm which is used for reconstructing high dimensional input data using neural network with a narrow bottleneck layer in the middle which contains the latent representation of the input data.

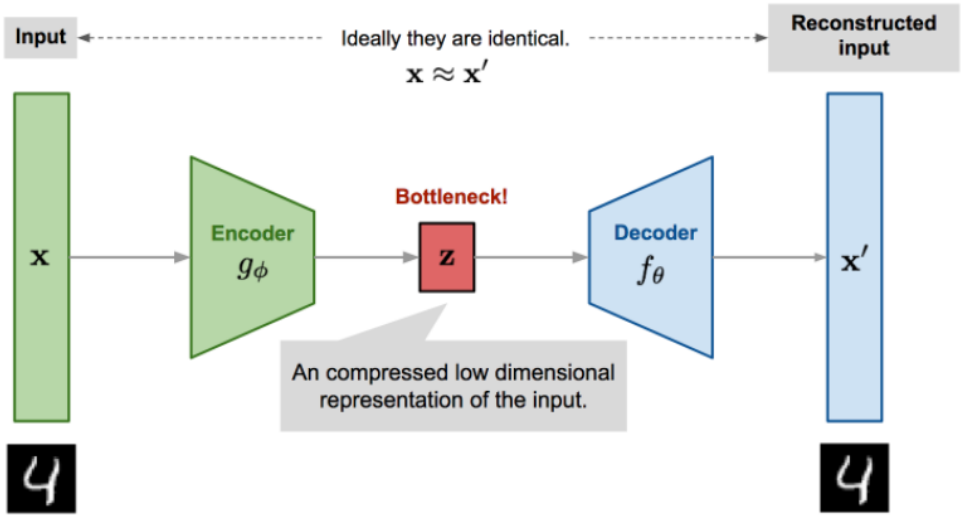


Fig. 1. Illustration of autoencoder model architecture.

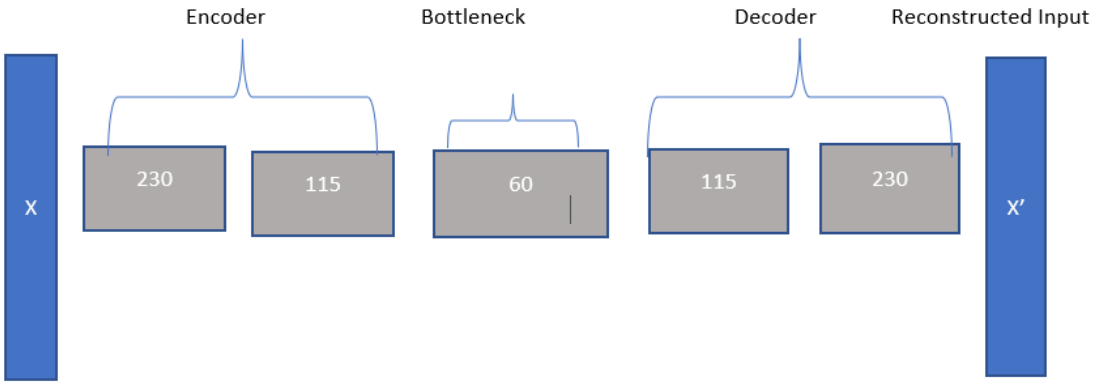
Autoencoder consists of an Encoder and a Decoder.

Encoder : Accepts High dimensional input data and translates it to latent low-dimensional data.The input size to an Encoder is larger than its output size.
Decoder : It receives the input from the Encoder output.It's objective is to reconstruct the input data.The output size of a decoder is larger than its input size.

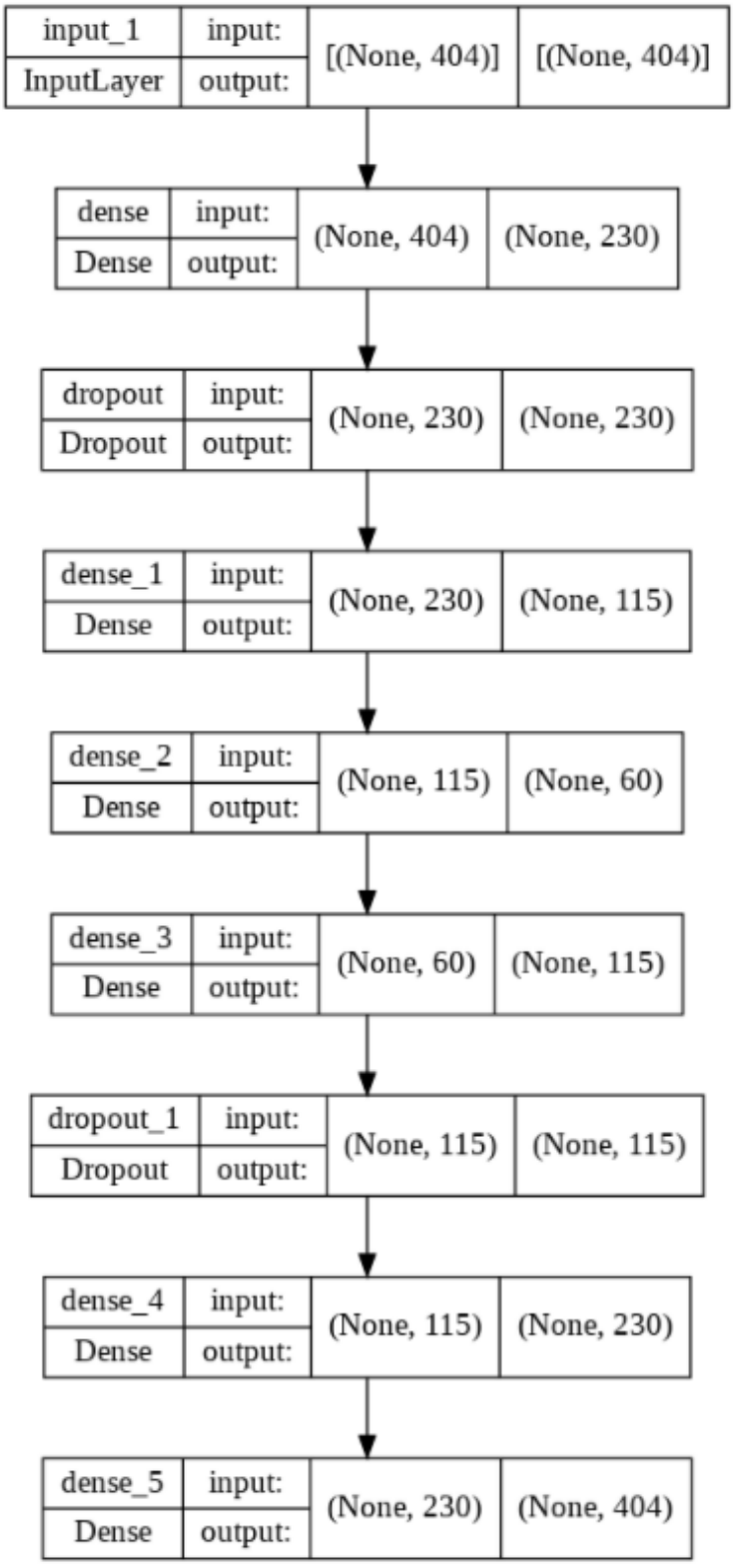
The Autoencoder accepts high-dimensional input data, compress it down to the latent-space representation in the bottleneck hidden layer; the Decoder takes the latent representation of the data as an input to reconstruct the original input data.

Advantages of Autoencoders:
1. Dimensionality Reduction
2. Recommendation Engines
3. Denoising Images
4. Image recognition
5. Image generation

The architecture of the autoencoder is shown below. :

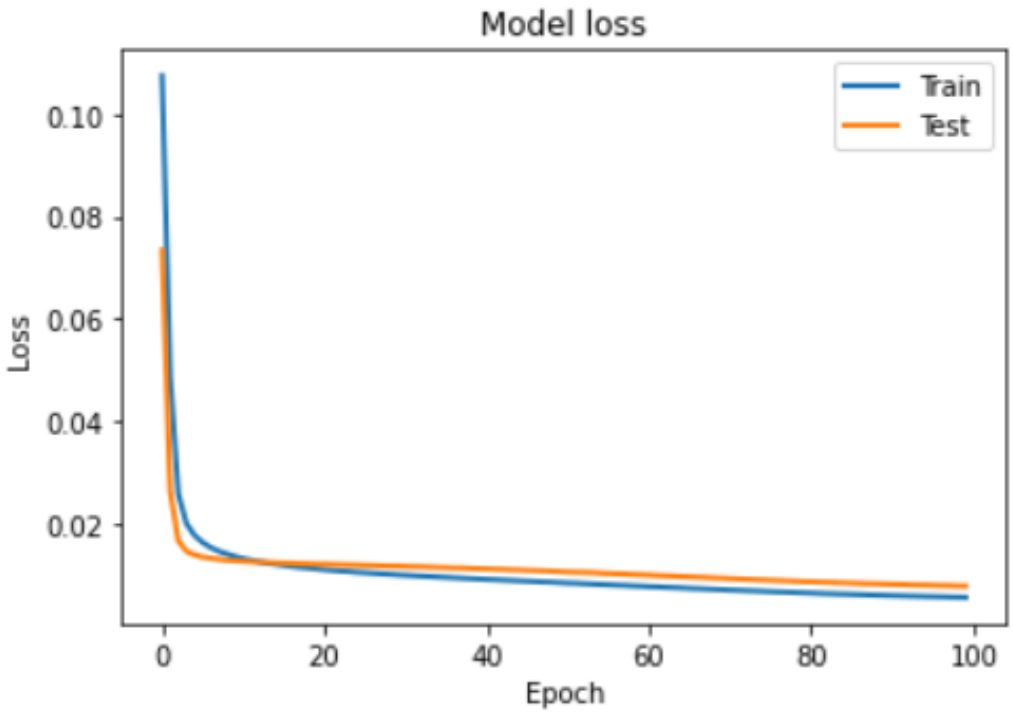


All the layers are densly connected.The input layer is of the same size as the number of features .The Dense layer is connected with the encoding dimension of 230 followed by another Dense layer of dimension 115. The bottleneck layer which has a dimension of 60.The bottleneck layer will learn the latent representation of the normal input data.The decoder will use bottleneck layers 60 and output to reconstruct the normal behaviour of the original input data.An Anomaly will be different from the normal behaviour.The Autoencoder will have trouble reconstructing the anomaly and hence the reconstruction error will be high.


input_1
InputLayer
input:
output:
[(None, 404)]
[(None, 404)]
dense
Dense
input:
output:
(None, 404)
(None, 230)
dropout
Dropout
input:
output:
(None, 230)
(None, 230)
dense_1
Dense
input:
output:
(None, 230)
(None, 115)
dense_2
Dense
input:
output:
(None, 115)
(None, 60)
dense_3
Dense
input:
output:
(None, 60)
(None, 115)
dropout_1
Dropout
input:
output:
(None, 115)
(None, 115)
dense_4
Dense
input:
output:
(None, 115)
(None, 230)
dense_5
Dense
input:
output:
(None, 230)
(None, 404)

The main objective is to minimize the mean squared error and the model was trained with the below constant hyperparameters :

1. **batch_size** =32
2. **epoch** = 100
3. **Optimizer** =SGD
4. **learning rate** = 1e-5
5. **activations**= relu /tanh



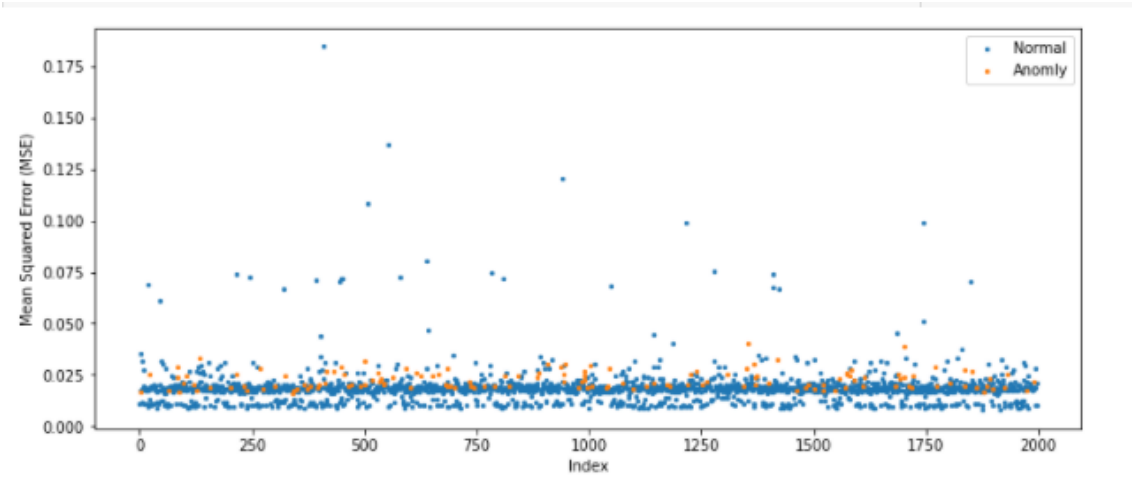
The loss decreases and it is steady for both Training and testing. After training the model the weight of the model were saved using the save_weights function.

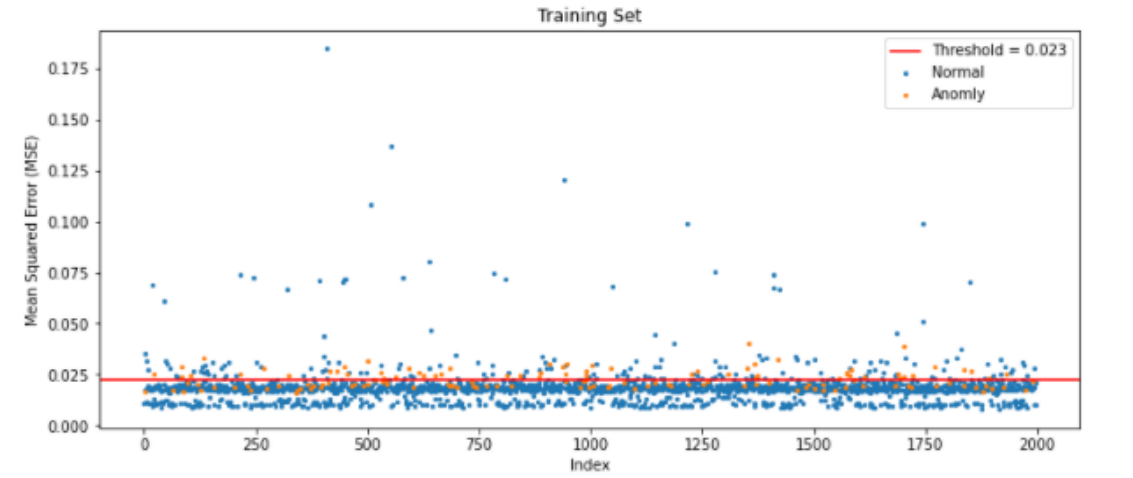
At First I decided to do GridSearchCV hyperparameter tuning using kerasClassifier but it was very time consuming hence i decided to test manually with some parameters I used below epochs - [80, 100,128] with batch size of [32,64,100] and then i decided to check different hyperparameters optimiziers= ['SGD', 'RMSprop', 'Adagrad', 'Adadelta', 'Adam', 'Adamax', 'Nadam'], learning_rate = [1e-5,1e-7,1] and with the different combinations of the activation functions =[relu,tanh,leaky_relu].After making so many experiments I got the best final results with the above mentioned hyperparameters.

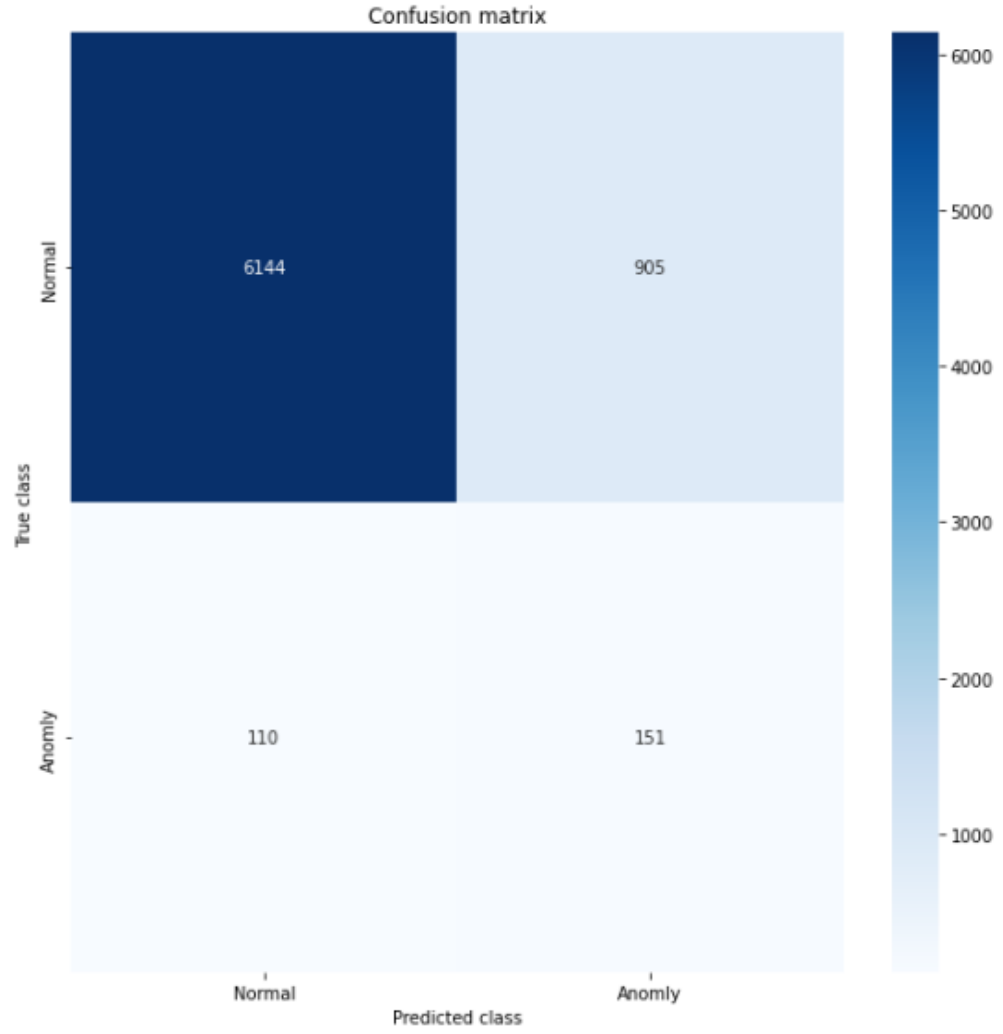
8 Reconstruction error

The reconstruction error is the element we use to detect the anomalies.To calculate the reconstruction loss we predict on the test data and calculate the mean squared error between the test data and reconstructured test data.

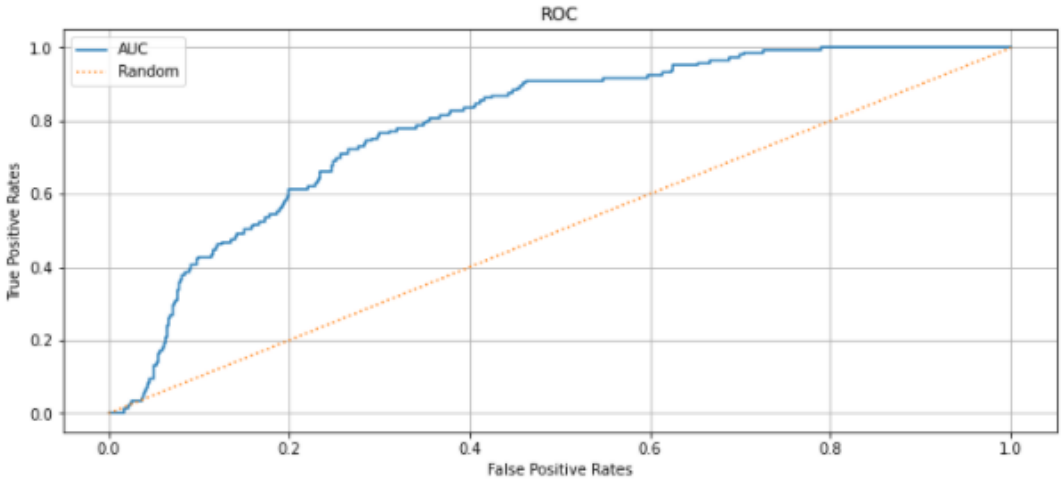
let's have a look at the distribution of Normal vs anomaly







True Positive - 6144 False Positive - 905 False Negative - 110 True Negative -151



9 Unsupervised Learning

9.1 What is Anomaly detection in Unsupervised learning ?

Unsupervised anomaly detection involves an unlabeled dataset. It assumes that the majority data points in the unlabeled dataset are “normal” and it looks for data points that differs from the “normal” data points. Here we feed both the Normal as well as Anomaly data during the training phase.

Let’s have a look at the four types of unsupervised algorithms :

9.2 One Class SVM

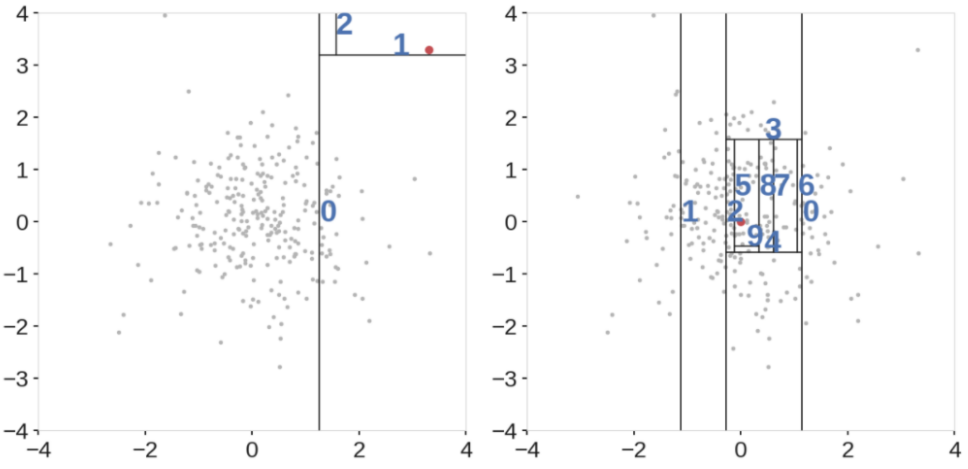
One Class SVM refers to the one class problem where all data belongs to single class.In this case the algorithm is trained to learn both on ”normal” as well as ”anomaly”.When modeling one class, the algorithm captures the density of the majority class and classifies examples on the extremes of the density function as outliers. This modification of SVM is referred to as One-Class SVM.It is mainly used for Imbalanced Classification.

Instead of hyperparameter tuning in other classification algorithms,one class SVM uses nu as a hyper-parameter which is used to define what portion of data should be classified as outliers.I have used nu=0.05 which means algorithm designate 5% data as outliers.I have used the default value of kernel as Radial Basis Function kernel which is used to find a regression line

The predicted dataset will have -1 or 1 values.I have Reshaped the values in which -1 corresponds to 1 Which refers to the anomaly and 1 corresponds to 0 which refers to Normal.

9.3 Isolation Forest

Isolation Forest is an unsupervised machine learning algorithm for identifying anomalies within a dataset by isolating anomalies as they are few and different. It is based on the Decision Tree algorithm. It isolates the outliers by randomly selecting a feature from the given set of features.Presumably the anomalies need fewer random partitions to be isolated compared to ”normal” points in the dataset, so the anomalies will be the points which have a smaller path length in the tree.Using Isolation Forest, we can not only detect anomalies faster but we also require less memory compared to other algorithms.The parameters used here is Contamination which refers to the proportion of outliers in the data set.The range is between (0, 0.5).



It is much easier to isolate an anomaly compared to normal observation

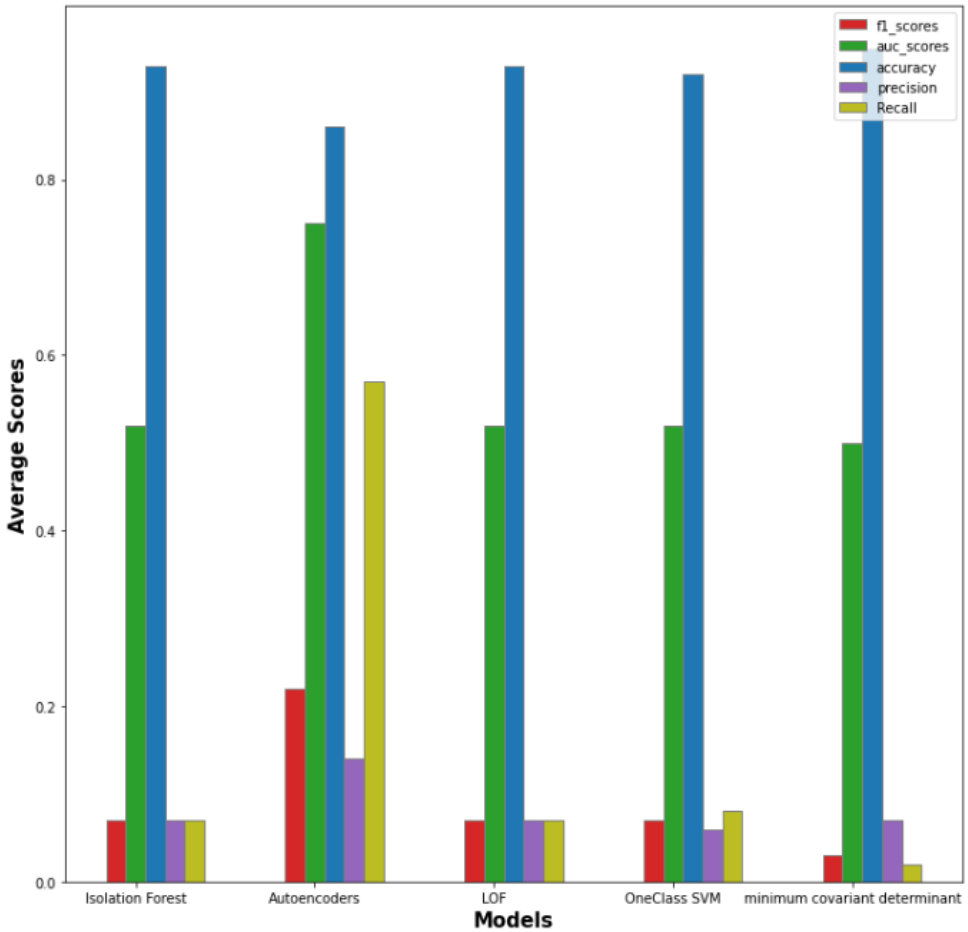
9.4 Local Outlier Factor

Local Outlier Factor is a density-based clustering to identify outliers in a multi-dimensional dataset by indicating the degree of outlierness quantifying how outlying the object is.
1. LOF identifies local outliers by comparing the local density of a point with the local densities of its k neighbors.
2. If the point has a substantially lower density than its k neighbors local density, it is considered an outlier.
3. Every data point in the dataset is assigned an outlier factor which is the degree of outlierness.

9.5 Elliptic Envelope

The Elliptic Envelope is a supervised as well as unsupervised algorithm to model the data as a high dimensional data Gaussian distribution with possible covariance between features. It then finds an Elliptical boundary that contains most of the data points and anything outside the elliptical boundary is an Outlier.For the contamination I experimented by using the default value of 0.1.

10 Conclusion



From the above diagram we see clearly Autoencoders is the winner based on auc_score f1_score.We can take the accuracy as a measure because it has imbalanced dataset. The accuracy will be always high for Imabalanced datasets.