Anomaly detection in sensor data

By- Priyanka Gangwar

Understanding of problem statement

- Object – **To prepare a model that can classify the coming data set has anomaly or not**
- Data preprocessing –
  - Duplicity of the time stamp indexes has been checked – Not found
  - Missing values – Availability of the data has been computed corresponding to each attribute. It is found that sensor_50 has 34% data unavailability. This problem can be handled in two ways as given below.
    1. Sensor_50 can be dropped from the analysis
    2. As in sensor 50 we can see in the data after row number 143327 is not available. We can drop all the rows after that and can have sensor 50 in the analysis. However, in this case we will loose more information.
    3. I have dropped sensor_50 column from the analysis. However, I have tried both ways, and as I though of first technique was giving more promising results.
    4. Even after dropping sensor 50 column from the analysis, still data has missing values. These missing values were filled by median. I have observed in many columns the difference between max and average was very large. Mean wont be a good approach so I have filled with median values.
    5. One trivial value has been observed in few columns that was 1000. The rows consisting of this value have been deleted because in the scalarization it will create a impact.

Problem solving approach

- **Problem with Dataset –**
  - This dataset is unbalanced, i.e., most of the given examples are from normal class and only 7% are from anomaly class. Any simple classifier can not handle all the scenarios.
- **Tried approach -**
  - First I tried interquartile approach, then clustering and after that Isolation Forest algorithm
  - Before approaching towards model building in this case, few other steps are followed those are listed below:
    1. Time series analysis has been done, i.e, time series is stationary or not. – Most of the features were following this assumption.
    2. Then PCA has been formed for dimensionality reduction and it is found only 2 principle components are significant.
    3. Now, again the stationarity and autocorrelation of these two principal components has been checked that they are stationary and not autocorrelated.

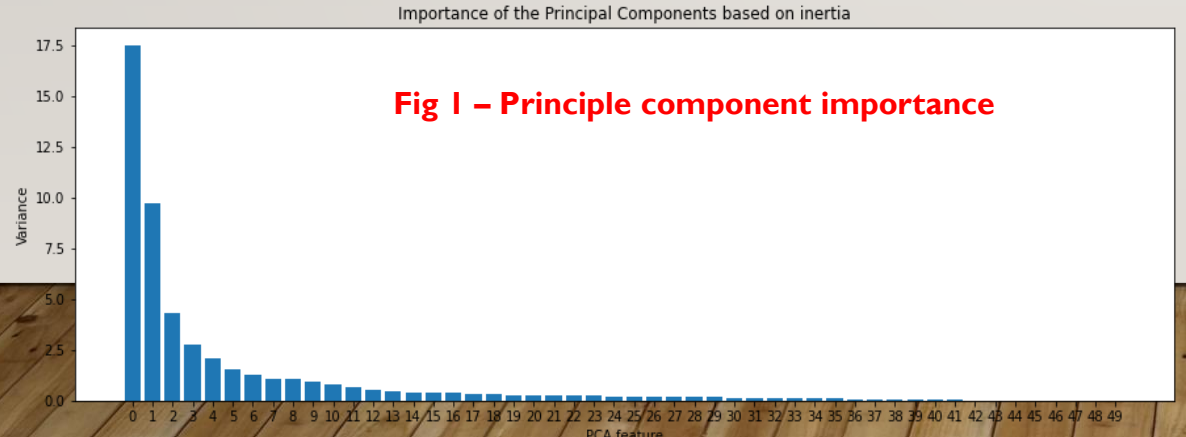
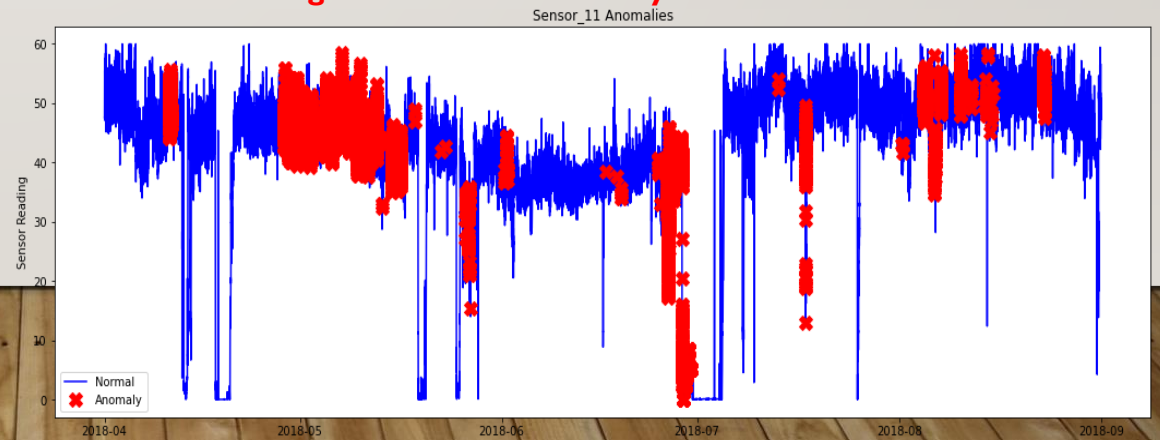


Fig 1 – Principle component importance

Fig 2 -Anomalies found by Isolation forest



Autoencoder based approach

- From the other tried approaches only Isolation Forest could work. However because of unbalanced dataset we can not trust it for prediction.
- In autoencoder based approach as we feed it only normal transactions, which it will learn to reproduce with high fidelity. So unbalanced data set wont create much problem. However, autoencoder works well in this kind of problem statement.
- Similar kind of data preprocessing has been done before model building.
- **Data Visualization** - Data visualization has been done using t-SNE. t-SNE is a dimensionality reduction technique used for visualizations of complex datasets. It maps clusters in high-dimensional data to a two- or three dimensional plane so we can get an idea of how easy it will be to discriminate between classes. It does this by trying to keep the distance between data points in lower dimensions proportional to the probability that these data points are neighbors in the higher dimensions.

Model building

- Training: only Normal data
  - Split into:
    1. Actual training of our autoencoder
    2. Validation of the neural network's ability to generalize
- Testing : mix of normal and anomaly data
  - Treated like new data
  - Attempt to locate outliers
    1. Compute reconstruction loss
    2. Apply threshold
- Normalising & Standardising – To reduce the impact the different range of multiple features and as well as it will make model more generalize for unseen data set.
- Visualization has been done to see the impact of normalization.
- Architecture of the autoencoder is defined- This architecture has been further improved using grid search and other techniques.
- Apply the transformation pipeline to test set. Then, pass the data through the trained autoencoder. Calculate the reconstruction loss for every transaction and draw a sample.

Model building contd…

- From the graph we can see that autoencoder identifying the abnormal cases however some transactions seems to be wrongly captured by autoencoder. (Fig 3)
- Latent space – Visualization to at the compressed representation our neural network devised. We can see normal data is closely grouped together however anomaly is scattered all over the space. We can see still model needs calibration or improved model architecture.
- Precision – 88.65%
- Recall – 11%

Note – As recall of this model is very low. This model architecture needs more hyperparameter tuning.

Fig 3 – Loss distribution

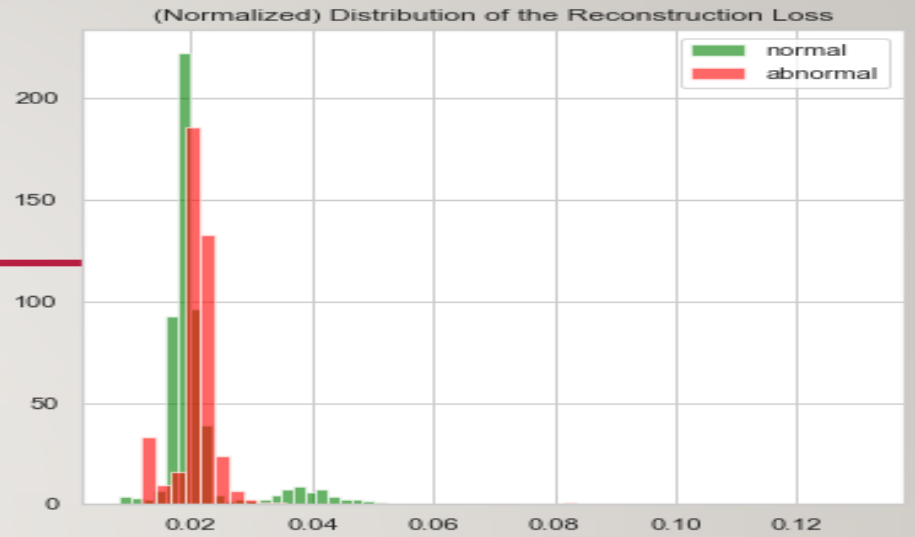


Fig 4 – Latent space presentation