IDS 575 Project Report (SP22) - Group 10

Anomaly Detection in Time Series Sensor Data

Karishma Mulchandani | Sravya Tangirala | Vennam Prahasith | Zohaib Sheikh

---

Objective

Anomaly Detection involves identifying the differences, deviations, and exceptions from the norm in the dataset. Our main goal is to detect anomaly in a time series data in manufacturing industry. Manufacturing industry is a heavy industry which requires large amount of capital investment on heavy machinery assets which are most critical for manufacturing. The ability to detect any anomaly in advance would result in mitigating the risk of an equipment failure.

Related Work/Current Industry Applications

Anomaly detection is a great application of machine learning. Few of the most widely used applications of anomaly detection include fraud detection, cyber-attack detection, and equipment monitoring. It is also commonly used to detect intrusions to a computer network and to detect the risk of medical problems in health data. Related work includes statistical frameworks for detecting Latent faults - Performance anomalies, that indicate a fault, or that could eventually result in a fault.

Application of Machine Learning and Statistics

There are many ways in which statistics and machine learning techniques can be leveraged for anomaly detection. We have used a few of them as part of our project and presented a comprehensive comparison between them in Python using Scikit-Learn. In our project we have explored both supervised and unsupervised techniques for anomaly detection. Specifically, we have implemented the below mentioned techniques:

- Interquartile range
- K means Clustering
- Gaussian Distribution
- Gaussian Mixture Model

For evaluation of these techniques, we will use different classifier evaluation metrics such as Number of Outliers detected, Confusion Matrix, Accuracy, etc.

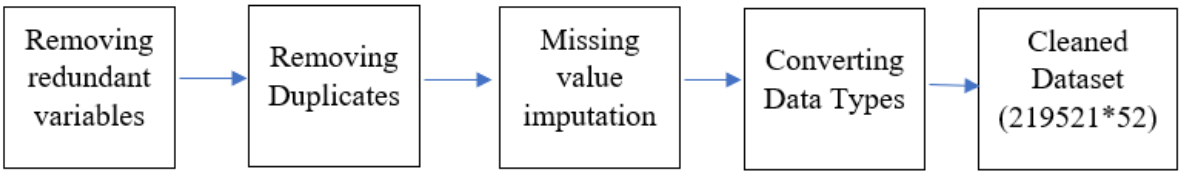
Understanding the Dataset and Dataset Ingestion

Our dataset contains sensor readings from 52 sensors installed on a pump which is a part of a manufacturing setup. These different sensors measure various behaviours of the pump. The dataset contains reading for one full year at different timestamps. We also have a ‘machine status’ column which represents different working conditions of the pump as ‘normal operating ‘broken’, and ‘recovering. Our dataset has 220k different data points each of which represents a reading of the 53 different sensors at a given timestamp. After downloading the dataset, we ingested it into our project using a Pandas dataframe.

Data Science Pipeline

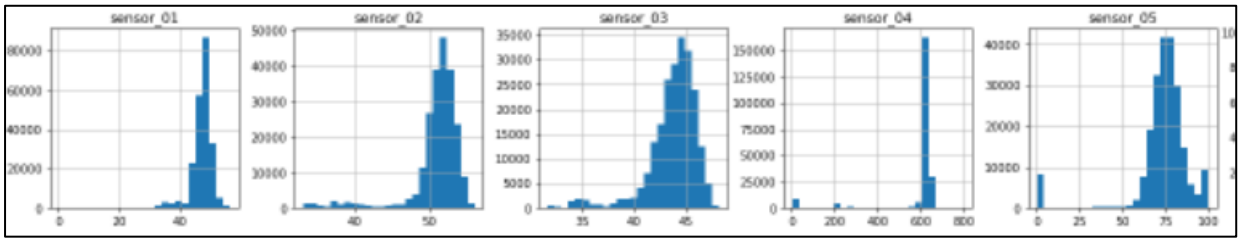
1. Data Cleaning and pre-processing

The dataset we used had a few abnormalities, like empty columns, repetitive entries, missing values, and more. So, we had to clean the dataset. We have also utilized an interesting concept/technique called pickling for this project. Pickling is basically a technique of converting a python object into a byte stream to be stored in the file and maintain program state across all sessions in the project. Our cleaned dataset has a shape of (219521, 52).



2. Exploratory Data Analysis

Below we visualize the distribution of some of the sensors.



We can see most of the sensor readings follow a normal distribution which is intuitive and this observation lets us use multivariate gaussian techniques.

We also visualized the correlation of all the sensors (below) with each other to better understand any underlying connection between them. The numbers highlighted in shades of green, blue, and purple indicate that the sensors they represent are highly correlated. This means that they have a strong relationship with each other. If a sensor detects an anomaly, it is likely that the remaining sensors with which it is strongly correlated will also detect the anomaly.

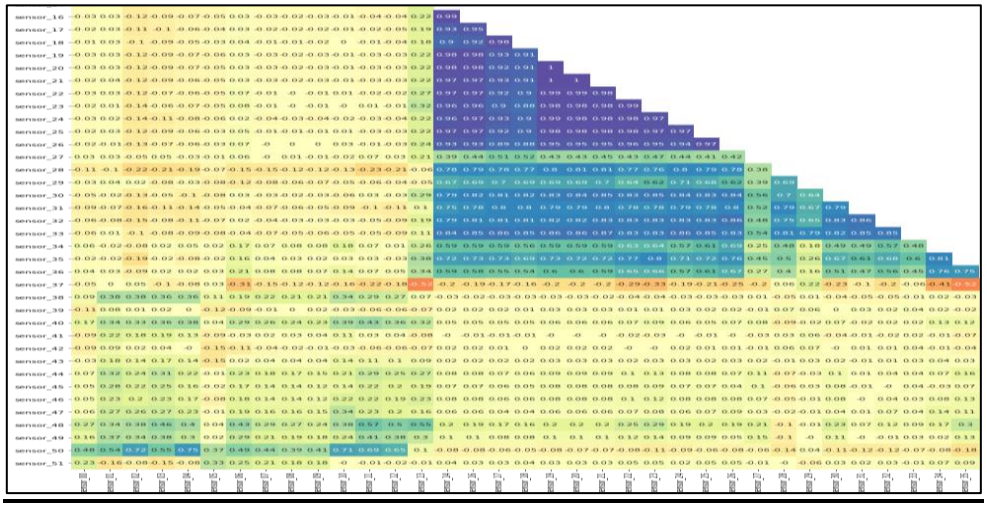


Fig 6.1 Excerpt from Correlation Matrix Heatmap

Sensors 2-12 seem to have high positive correlation with values ranging from around 0.2 to 0.9 whereas sensor 37 has high negative correlation with sensors 13, 35, 36 respectively with correlation values around -0.5.

As we aim to find anomalies in the machinery, our focus will be on the detection of the machines which have a predicted status of "Broken" and "Recovering". Below is an excerpt from the plots of the readings of 2 of the sensors with respect to time. The 'Broken' readings are marked with a red cross and the ' Recovering ' readings are highlighted in yellow to indicate the anomaly.

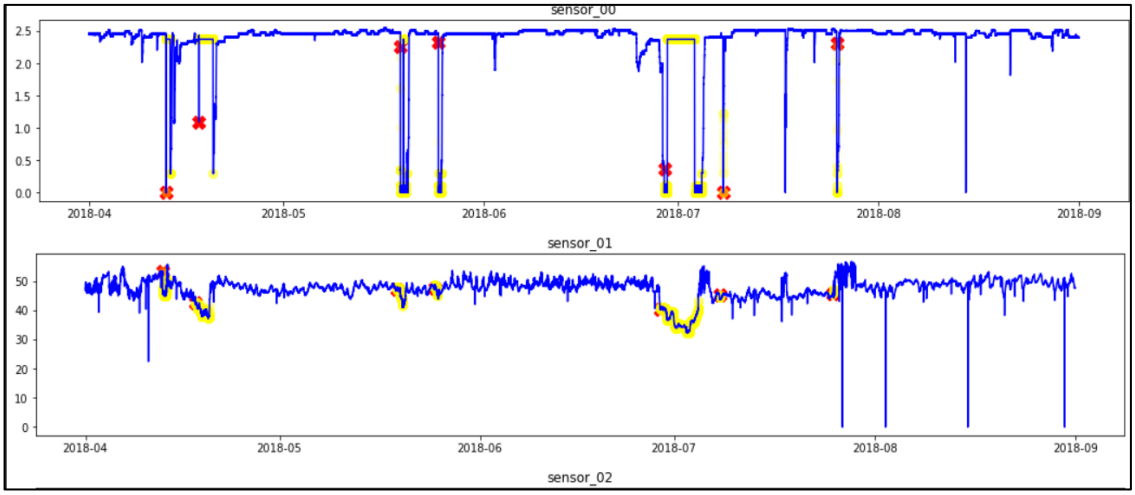
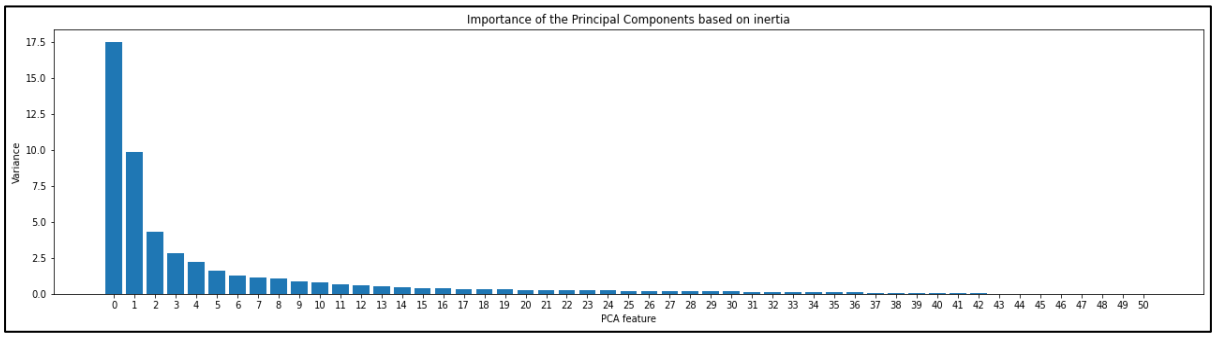


Fig 6.2 Sensor Readings with Detected Anomalies

As seen clearly from the above plots, the red marks, which represent the broken state of the pump, perfectly overlaps with the observed disturbances of the sensor reading. Now we have a pretty good intuition about how each of the sensor reading behaves when the pump is broken vs operating normally.

3. Feature Engineering

Feature Engineering is a machine learning technique that takes advantage of data to generate new variables that are originally not a part of the training dataset. Feature Engineering can be used to produce new features for supervised learning as well as unsupervised learning. The primary purpose of feature engineering is to simplify as well as speed up data transformations while improving model accuracy. In this step, we will scale the data and apply Principal Component Analysis (PCA) to extract the most important features to be further used in training models. It is computationally quite expensive to process the data of this size, (219521, 53), hence the reason for reducing the dimensionality with PCA.

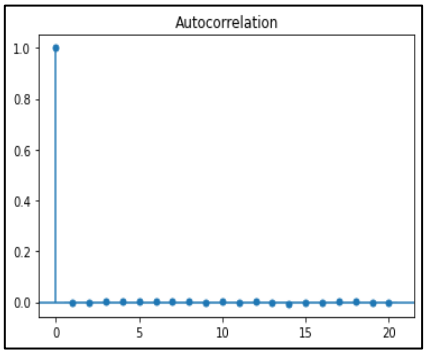


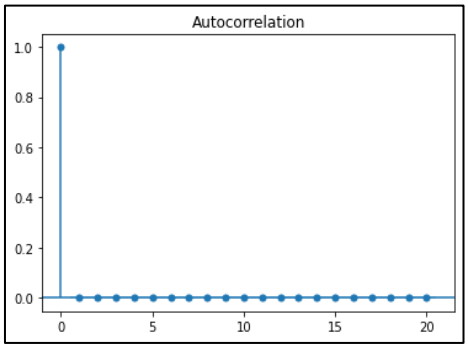
It appears that the first two principal components are the most important as per the features extracted by the PCA in above importance plot. So as the next step, I will perform PCA with 2 components which will be my features to be used in the training of the base model and the unsupervised model.

Next, we checked the stationarity and autocorrelation of these two principal components just to be sure they are stationary and not autocorrelated.

Stationarity Test : For this we used the Augmented Dickey Fuller Test. Running the Dickey Fuller test on the 1st principal component, we got a p-value of 5.4536849418486247e-05 which is very small number (much smaller than 0.05). Thus, we rejected the Null Hypothesis and say the data is stationary. We performed the same on the 2nd component and got a similar result. So both principal components are stationary.

Autocorrelation Test : For this test, we used the ACF plot to visually verify that there is no autocorrelation for the two principal components. As evident from the below plot our two components have no autocorrelation.





4. Statistical Modelling

Statistical modelling can be defined as the process of applying statistical analysis to a dataset. It uses mathematical models and statistical assumptions to generate sample data and make predictions. We have used the following statistical modelling techniques –

4.1 BASE Model : Inter Quartile Range

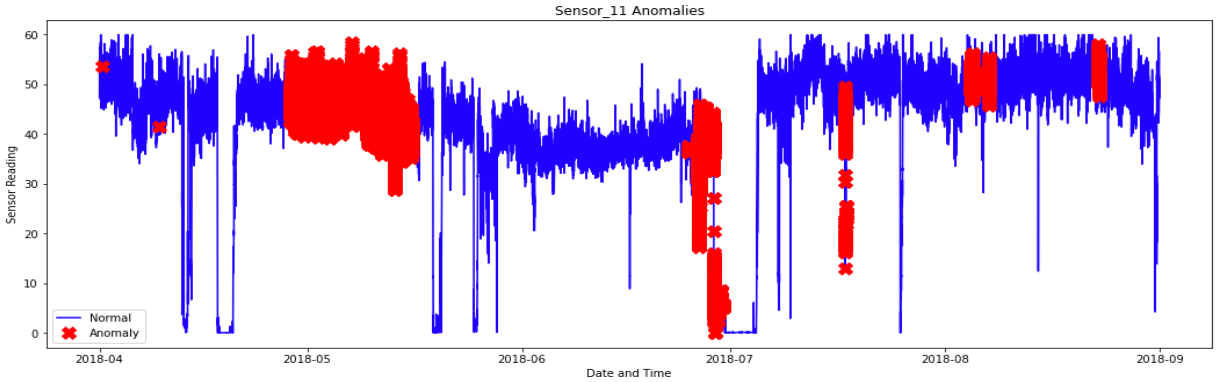
The interquartile range is the difference between the third quartile and the first quartile of the distribution, and it is used to measure variability. It helps us get an estimate of how wide our distribution is.

Now, let us walk through how we used Inter Quartile range to our advantage in this project. First, we calculated the inter quartile range which is the difference between Q3 and Q1. We then calculated the upper bound and the lower bound i.e. 1.5 times of IQR to mark the outliers. As we have now created the upper and lower bounds, we are able to classify any data points out of the bounds as outliers/anomalies. So, any data points that fell outside the upper and lower bounds were flagged and marked as anomalies/outliers.

Next, we implemented Univariate feature selection to select k most important features. We used the Chi-Square test to identify the 3 most important sensors in our dataset. These are as below:

```
     Feature          Score
11   sensor_11   10106.761967
12   sensor_12    9879.052739
4    sensor_04    8167.176442
```

Lastly, we plotted the abnormalities on the time series data as seen in the figure below for sensor 11.

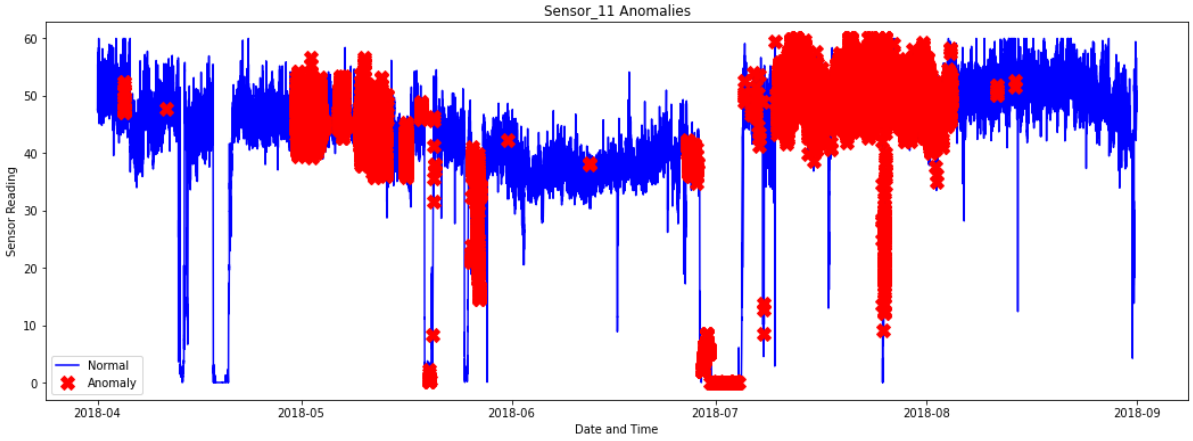


As seen from plots, there are a lot more outliers in pc1 (1st principal component) than that from pc2. The outliers in pc1 represent approximately 14% of the data set. Also the outliers in pc1 seem to better explain the failures in the sensor readings from one of the sensors, sensor_00 is used in this case.

4.2 K-means Clustering

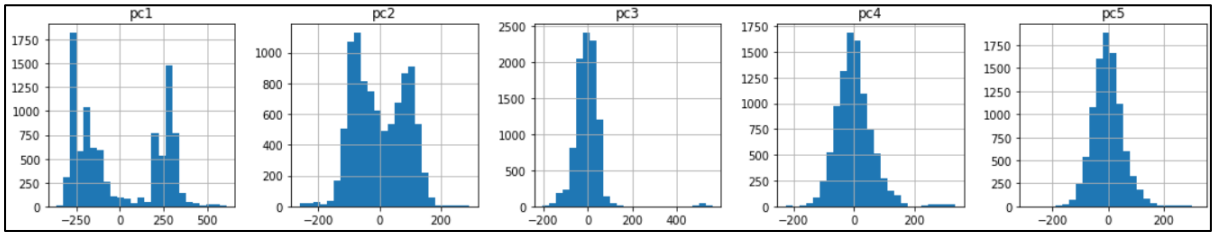
The underline assumption in the clustering-based anomaly detection is that if we cluster the data, normal data will belong to clusters while anomalies will not belong to any clusters or belong to small clusters. We used the following steps to find and visualize anomalies.

- Calculate the distance between each point and its nearest centroid. The biggest distances are considered as anomaly.
- We use outliers_fraction to provide information to the algorithm about the proportion of the outliers present in our data set. Situations may vary from data set to data set. However, as a starting figure, I estimate outliers_fraction=0.14 (14% of dataset are outliers from base model)
- Calculate number_of_outliers using outliers_fraction.
- Set threshold as the minimum distance of these outliers.
- The anomaly result of anomaly1 contains the above method Cluster (0:normal, 1:anomaly).
- Visualize anomalies with cluster view.
- Visualize anomalies with Time Series view.



4.3 Multivariate Gaussian Distribution

Multivariate Gaussian Distributions (multi-variate) are high dimensional normal distributions (univariate). A vector is said to be multi-variate normally distributed if all the linear combinations of its components follow normal distributions. We implemented this model on 5 of the principal components. The model learns by estimating the parameters of the distribution and assigning observations to the distribution based on the probabilities. Below is the distribution of our first 5 principal components:



The 5 principal components chosen have near to normal distributions, all centred around 0. Each distribution is defined by its own set of parameters-mean and covariance matrices.

After we calculated the mean and co-variance matrix of the five principal components, we applied a multivariate gaussian distribution on the components collectively. We defined a threshold called “epsilon” which we use to determine if an observation should be flagged as an anomaly or not. After we have set the multivariate distribution, we implemented a search algorithm using the F1 score to pick the best threshold for flagging an observation as an anomaly.

The best value for “epsilon” is determined using a stepwise iterative process. The size of the step is determined from the maximum and minimum values of the probabilities of the observations (which are determined by the multivariate normal random variable). This step is used to iterate through the range of probabilities generated, each of which is a potential epsilon. Predictions are made at every stage on the test data for every potential value of the epsilon for which the F-score is calculated. The epsilon with the highest F-score is chosen as the “best” epsilon.

Next, we will flag an observation as an anomaly if the probability of that observation to be a part of the dataset is less than the determined threshold. We got the best threshold value as 1.7111345161379106e-17.

4.4 Gaussian Mixture Model

Gaussian Mixture Model can be considered a "generalized" version of k-means algorithm where clustering is done using probability measures. The Gaussian Mixture Model used applies an iterative EM (Expectation Maximization) algorithm to fit a mixture of Gaussian models where the following steps are repeated until convergence occurs:

- **E-Step-** "Soft clustering" is performed. This means that there are no restrictions on the number of clusters a datapoint may belong to (unlike k-means which performs hard clustering and assigns a datapoint to only one cluster) and considers the possibility that clusters may overlap (mixed membership). This step returns the probability of each point belonging in a certain cluster (in our case, anomaly or not)
- **M-Step-** updates the membership and parameters of the clusters

We used the model for anomaly detection in 2 ways:

1) Predicting the labels from the model generated

2) Using the predicted probabilities from the model to detect anomalies. This was implemented using the same threshold mechanism as used in the Multivariate Gaussian Distribution model.

5. Evaluation and Comparison of Models

For this project as seen above, we have used 4 different models namely Interquartile Range, K-Means clustering, Multivariate Gaussian Distribution and Gaussian Mixture Model. Now, let us compare the results that we received from these three models. We have indexed 0: as a normal and well-functioning machine and 1: as an anomaly.

Model	Accuracy	F-Score
K-means Clustering	82%	N/A
Multivariate Gaussian	94%	0.602
Gaussian Mixture Model (using labels)	80%	0.07
Gaussian Mixture Model (using probabilities)	94%	0.076

Model	Result Returned	Count
Interquartile Range (IQR) for 'Anomaly_pc1'	0 : Normal functioning	189,644
	1: Anomaly / Abnormal functioning	29,877
K-means clustering model	0 : Normal functioning	190,984
	1: Anomaly / Abnormal functioning	28,537
Multivariate Gaussian model	0 : Normal functioning	202,033
	1: Anomaly / Abnormal functioning	17,488
Gaussian Mixture Model (using labels)	0 : Normal functioning	205,067
	1: Anomaly / Abnormal functioning	14,454
Gaussian Mixture Model (using probabilities)	0 : Normal functioning	202,033
	1: Anomaly / Abnormal functioning	17,488

∴ we choose the Multivariate Gaussian Distribution model as our best model.

Conclusion

So far, we have done anomaly detection with four different methods. In doing so, we went through most of the steps of the commonly applied Data Science Process which includes the following steps:

1. Problem Identification
2. Data Wrangling
3. Exploratory Data Analysis
4. Pre-processing and training data development
5. Modeling
6. Documentation

One of the challenges we faced during this project is that training anomaly detection models with unsupervised learning algorithms with such a large data set can be computationally very expensive. This limited us from implementing SVM (Support Vector Machine modelling) on this data as it was taking a very long time to train the model with no success. We suggest the following next steps on improving the model:

1. Feature selection using advanced techniques
2. Advanced hyperparameter tuning
3. Implementing other learning algorithms such as SVM, DBSCAN, etc.

References

- https://www.kaggle.com/datasets/nphantawee/pump-sensor-data
- https://towardsdatascience.com/feature-engineering-for-machine-learning-3a5e293a5114
- https://scikit-learn.org/
- https://stackoverflow.com/
- https://www.kaggle.com/datasets/nphantawee/pump-sensor-data
- http://node101.psych.cornell.edu/Darlington/series/series1.htm
- https://iwringer.wordpress.com/2015/11/17/anomaly-detection-concepts-and-techniques/#:~:text=Among%20them%2C%20Anomaly%20detection%20detects,data%20cleanup%2C%20and%20predictive%20maintenance