Anomaly detection in Sensor Data

Group # 2
Team member 1 (**Alok Kumar Mishra** , **811191814**, **amishra2@kent.edu**)
Team member 2 (**Yuan Chen , 811074030**, **ychen135@kent.edu**)
Team member 3 (**Chhavi Vishnoi**, **811169000**, **cvishnoi@kent.edu**)
Team member 4 (**Priyanka Vyas** , **811188035**, **pvyas2@kent.edu**)

1. Introduction

The project focuses on the application of anomaly detection on a machinery dataset that pertains to Manufacturing Industry. There are several real applications that have been implemented on anomaly detection concepts like detecting fraud transactions, fraudulent insurance claims, cyber attacks and detecting abnormality in the machinery behavior etc. It is a technique that helps identify abnormal behavior as compared to an established pattern defined by attributes in a dataset. Any data points that deviate from an established baseline pattern is called an anomaly. Anomaly detection helps detect and alert abnormal equipment behavior because anomalies show something different is happening than expected.

Detecting and responding to some type of input from the physical environment through the output of a device is sensor data. Identifying the differences, deviations and exceptions from the norm in a dataset is anomaly detection. It is also known as outlier detection. It is an old technique which is used as a common application of Machine learning. Detecting fraud transactions, fraudulent insurance claims, cyber attacks to detect abnormal behaviors around the internet are some of the real world examples of anomaly detection. In the case of industrial anomaly detection, devices get damaged from constant, rigorous use. To stop further losses and escalation, the damages must be found as soon as feasible. Because they are acquired using various sensors and gathered for analysis, the data in this field are referred to as sensor data. Therefore, it can be said that in this context, anomaly detection techniques monitor the performance of industrial components, such as motors, turbines, oil flow in pipelines, or other mechanical components, and detect problems that might arise due to wear and tear or other unforeseen events.

Depending on the interpretation and context, anomalies themselves can be good or detrimental. Anomalies in data can be translated into substantial actionable information in a wide range of application fields, which has an impact on the significance of anomaly detection. The ability to act on the system to appropriately respond to, avoid, or remedy the circumstances they relate to is empowered by the decision-accurate maker's detection of these kinds of anomalous information. We are trying to implement the application of anomaly detection in the manufacturing industry. Manufacturing industry is one of the industries which is lagging behind in using machine learning techniques effectively. The detection of anomalies in sensor data from manufacturing industries will be implemented in the project.

2. Project Description

2.1 Brief descriptions of your project

The goal in this project is to apply unsupervised learning techniques to search for anomalies in the time series sensor readings from a pump. For a manufacturing industry such equipment are

the most critical assets for the plant operations. The main factor driving a huge concern for these assets is that failure of the equipment frequently results in production loss, which, depending on the size and scale of the operations, might result in losses of hundreds of thousands or even millions of dollars. Hence, anomaly detection can help identify the equipment's abnormal behavior to mitigate the risk and reduce future production losses.

2.2 Challenges and technical contributions (new problems or new solutions?) in your project
Finding publicly available data from the industrial sector for the project idea was difficult. However, we could find a dataset from 'kaggle' that refers to 53 sensors that were mounted on a pump to evaluate a variety of pump behaviors, and the data collection includes sensor values from those sensors.

2.3 The workload distribution for each member in your team
The workload is distributed uniformly amongst the team, following a discussion over the ideation and implementation of the project, every member has worked and contributed on respective assigned report sections and aspects of the project.

3. Background

3.1 Literature Review

PCA (Principal component analysis) [1] is a conventional method to reduce the dimension of high - dimensional data. It projects the high-dimensional data into the low-dimensional space by the method of linear transformation, so as to obtain a relatively small number of unrelated attribute subsets in the original attribute set. PCA method requires that the data after dimensionality reduction should not be distorted, that is, the dimensionality dropped should be noisy or redundant.

Interquartile range [2][3] is a widely accepted method of finding outliers in data set. When the interquartile or IQR is used, the complete data set is divided into four equal parts or quartiles. The distance between the quartiles is used to determine the IQR.

K-means algorithm [4] is a kind of clustering method based on division proposed by J.B.MacQueen. The algorithm is simple, efficient, and suitable for the processing of large-scale data sets, so it is widely used in various fields. At the end of the algorithm, the data set X{} in space can be divided into K clusters of different classes, so that the similarity between the classes is as small as possible and the similarity within the classes is as large as possible.

Isolation Forest (i_Forest) [5]is a fast outlier detection method based on Ensemble. With linear time complexity and high precision, i_Forest is a State-of-the-art algorithm that meets the requirements of big data processing. It was first proposed by Professor Zhou Zhihua and others in 2008, It is suitable for anomaly detection of Continuous numerical data. Different from other anomaly detection algorithms, which describe the degree of alienation between samples by quantitative indicators such as distance and density, isolated forest algorithm detects outliers by isolating sample points, Compared with traditional algorithms such as LOF and K-means, the isolated forest algorithm has better robustness to high-dimensional data.

3.2 Our workflow and related tools and skills

Anomaly detection (or outlier detection) is the identification of rare items, events, or observations that raise suspicion by being different from most data. Often, abnormal data can be associated with some kind of problem or rare event, for example, malfunctioning equipment, structural defects, bank fraud, medical problems, and so on. This connection makes it interesting to be able to pick out which data points can be considered exceptions, because identifying these events is often interesting from a business perspective.

In this section, our experimental process and related details will be introduced.

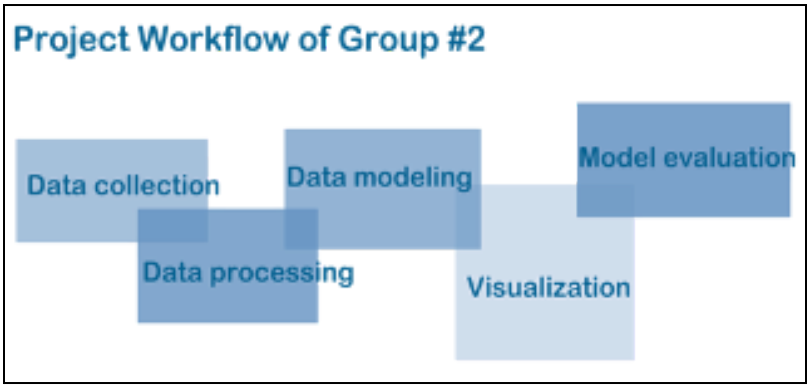


Fig 1 shows our workflow.

3.2.1 Data processing

The status information of water pump equipment is characterized by multiple sources, heterogeneous information, large quantity and various attributes, and those data are uncertain, noisy and inconsistent. The quality of the original data often cannot meet the requirements of the subsequent state evaluation model, so it is essential to carry out data cleaning before the state evaluation or diagnostic analysis. Data cleaning can improve data quality by filling in missing values, smoothing noise data and identifying outliers, which is helpful to improve the accuracy and efficiency of the data mining process.

According to the operation characteristics of the pump equipment, the abnormal state data usually manifests in two forms:

1. Anomalies that can be used for data cleaning, namely noise points and missing values;
2. Data is abnormal due to interference in the running status of the device.

The time series of equipment status data often contains multiple abnormal data. The goal of data cleaning is to repair all the noise points and missing values, and to achieve the effective acquisition of sudden fault information, rather than the elimination of abnormal data.

3.2.2 Data modeling

The most common way to perform condition monitoring is to look at each sensor measurement of the machine and impose minimum and maximum limits on it. If the current value is within the bounds, then the machine is healthy. If the current value is out of range, the machine is not operating properly, and an alert is sent.

In our project, we will use the interquartile range [3] to determine the computational threshold and calculate the outlier data, the interquartile range is the difference between the values ranking

in 25% and 75% in a data set, the values are denoted as Q1 and Q3 respectively. In general, IQR thresholding strategy calculates the threshold as follows:

$$Tmin = Q1 - c * ICR; Tmax = Q3 + c * IQR$$
$$IQR = Q1 - Q3$$

Where c is set to 1.5 usually.

We also tried the simple and efficient k-mean clustering algorithm and optimized the algorithm. The following are our optimization steps:

1. For each data point *x*% in the data set X$\{x_1, x_2, ...x_n\}$ on R^d, calculate its compactness,$Tigh(x_i) = \frac{1}{\sum_{x_j \in Gt(xi)} D(x_i, x_j)}$, the set of t nearest neighbor data points of the G_t data point x_i.

2. All the sparse data points of compact $Tigh < \frac{1}{n} \sum_{x \in X} Tigh(x)$ in X are deleted to obtain the set X' of dense data points.

3. In set X', x of $Tign_{max}(x)$, which is the most compact, is taken as the first initial cluster center c_1. The data point farthest from *c*!is taken as the second initial clustering center c_2; The *m* ($3 \leq m \leq$ K) initial cluster center c_m is the data point x_i, that meets the following conditions ($x_i \in$ X' :max($D_{min}(x_i, c_1)$, $D_{min}(x_i, c_2)$, ..., $D_{min}(x_i, c_{m=1})$,), i - 1,2,...,n) until the final K initial cluster centers are obtained.

3.2.3 Model evaluation

To measure the accuracy of the algorithm model, we introduces the performance index Detection Rate TPR(True Positive Rate), false positive Rate FPR(False Positive Rate) and running time T. TPR is used to measure the probability rate that outliers are correctly labeled, and FPR is used to measure the probability rate that positive and constant values are incorrectly labeled. The calculation is as follows:

$$TPR = \frac{TP}{TP + FN} \times 100\%$$

$$FPR = \frac{FP}{FP + TN} \times 100\%$$

In the equation above, TP represents the correctly detected outliers; TN represents the undetected outlier. FP indicates that normal values are detected as outliers; FN indicates that the normal value is detected as normal.

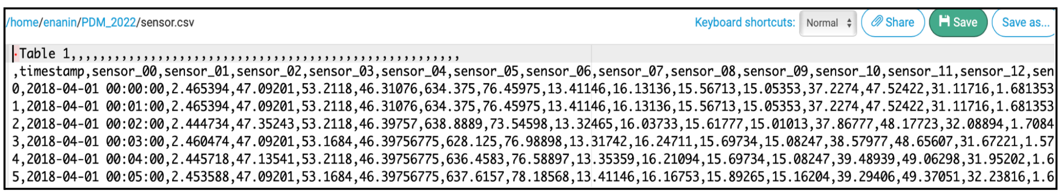
3.3 Experimental environment and data set

3.3.1 Experimental environment

In our project, the water pump sensor data set is used for experiments. Experiment on the Intel (R) Core (TM) i5-6400 CPU, Windows 10 operating system environment.

3.3.2 Experimental Data set

In our project, we analyzed the time series data from 56 different sensors, among which different sensors detected different types of data, including but not limited to: 1. The pH probe was used to collect the pipeline pH value; 2. The water pressure probe is used to collect the pressure value of the pipeline; 3, the water temperature probe is used to collect the water temperature of the pipeline, using the PTC temperature sensor; 4, simulate the water pressure value detected by the water pressure sensor; 5. Electromagnetic flowmeter is used to detect the flow rate of pipeline liquid; 6, the environment detector is used to detect the environment temperature and humidity; 7. Power parameter collector Used to collect the voltage and current of single-phase or three-phase electricity.



```
/home/enanin/PDM_2022/sensor.csv                Keyboard shortcuts: Normal  Share  Save  Save as...
Table 1,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
,timestamp,sensor_00,sensor_01,sensor_02,sensor_03,sensor_04,sensor_05,sensor_06,sensor_07,sensor_08,sensor_09,sensor_10,sensor_11,sensor_12,sen
0,2018-04-01 00:00:00,2.465394,47.09201,53.2118,46.31076,634.375,76.45975,13.41146,16.13136,15.56713,15.05353,37.2274,47.52422,31.11716,1.681353
1,2018-04-01 00:01:00,2.465394,47.09201,53.2118,46.31076,634.375,76.45975,13.41146,16.13136,15.56713,15.05353,37.2274,47.52422,31.11716,1.681353
2,2018-04-01 00:02:00,2.444734,47.35243,53.2118,46.39757,638.8889,73.54598,13.32465,16.03733,15.61777,15.01013,37.86777,48.17723,32.08894,1.7084
3,2018-04-01 00:03:00,2.460474,47.09201,53.1684,46.39756775,628.125,76.98898,13.31742,16.24711,15.69734,15.08247,38.57977,48.65607,31.67221,1.57
4,2018-04-01 00:04:00,2.445718,47.13541,53.2118,46.39756775,636.4583,76.58897,13.35359,16.21094,15.69734,15.08247,39.48939,49.06298,31.95202,1.6
5,2018-04-01 00:05:00,2.453588,47.09201,53.1684,46.39756775,637.6157,78.18568,13.41146,16.16753,15.89265,15.16204,39.29406,49.37051,32.23816,1.6
```

Fig 2 shows the partial data.

Finding patterns in data that do not match an a priori expected behavior is known as anomaly detection. This is related to the issue when some samples are different from the rest of the dataset in terms of a particular statistic, and these anomalous samples are labeled as outliers. Due to its importance in practical applications including intrusion detection, fraud detection, defect detection, and system health monitoring, among many others, anomaly detection has recently caught the interest of the industry. The literature has suggested methods for detecting anomalies based on distribution, distance, density, grouping, and classification. Their uses differ according to the user, the domains of the problems, and even the dataset. Anomaly detection frequently has a connection to outlier detection.

Outliers in statistics are instances of data that depart from the sample in which they occur. Through literature [6] it is learned that Anomaly detection techniques are implemented using following types of approaches:

1. Distribution-based approaches where the data points are modeled using a specific statistical distribution. Anomalies or outliers are then noted for points that depart from the model.
2. Depth-based approaches calculate the various convex hull layers and mark any anomalous or outlier objects in the outer layer.
3. Clustering approaches where anomalies or outliers are defined as elements that do not belong to or are close to any cluster and can be found using a variety of clustering algorithms.
4. Distance-based approaches where the distance between an element and a subset of the elements that are closest to it is indicated by distance-based anomalies or outlier identification.

5. Density-based approaches, in which through the use of the local outlier factor, density-based anomalies or outlier identification have been presented as a solution to the multi-density problem.
6. Spectral decomposition where the data is embedded in a lower-dimensional subspace using spectral decomposition so that data instances may be easily distinguished from one another.
7. Classification approaches identifies the categories to which an observation belongs is the difficulty that is put forth in this instance

In the project we are using K-means clustering techniques of vector quantization that aims to split n observations into k clusters, where each observation is assigned to the cluster that has the closest mean (also known as the cluster centroid or cluster centers), acting as a model for the cluster. The solution is proposed to be implemented using Python language and designed a model including libraries like Tensorflow, Numpy, Pandas, Kaggle, matplotlib, seaborn alongwith Scikit learn i.e. sklearn.

4. Problem Definition

As illustrated in Section 2, the goal for the project is to identify the abnormal behavior of the pump using anomaly detection technique as any abnormality in the system or equipment can lead to huge financial losses for the respective industry. Therefore, in order to reduce unplanned downtime, unnecessary maintenance and to manage important components for these assets more effectively, it is crucial to be able to recognize anomalies in advance and be able to mitigate risks. Moreover, the production loss due to unscheduled downtime, the expense of unnecessary maintenance, and the excess or shortage of critical components all have significant financial implications.

The data used to study the problem definition is a Time series sensor data. It is a collection of observations that have been organized and are connected over time. The challenge while working on the dataset is that the dataset has missing values, empty columns and timestamp with incorrect data type. Hence, the data needs to be cleaned, preprocessed and checked for stationarity and autocorrelation before feeding them into a clustering algorithm to detect anomalies.

The technique proposed for implementation is K means clustering technique such that the data instances outside of established clusters may be designated as anomalies. It is the process of organizing a collection of items so that objects in one group (referred to as a cluster) resemble one another more closely than those in other groups (clusters). In many domains, including pattern recognition, image analysis, information retrieval, bioinformatics, data compression, computer graphics, and machine learning, it is a frequent statistical data analysis technique and a key goal of exploratory data analysis.

The iterative Kmeans algorithm tries to divide the dataset into K pre-defined unique, non-overlapping subgroups (clusters), each of which contains only one group to which each data point belongs. While keeping the clusters as distinct (far) apart as possible, it aims to make the intra-cluster data points as comparable as possible. It distributes data points to clusters in a way that minimizes the sum of the squared distances between the data points and the cluster centroid, which is the average value of all the data points in the cluster. The more similar the data points inside a cluster are, the less variation there is between clusters. Mathematically the objective function can be defined as:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \|x^i - \mu_k\|^2$$

where wik=1 for data point xi if it belongs to cluster k; otherwise, wik=0. Also, μk is the centroid of xi's cluster.

On a technical level, we will first differentiate J w.r.t. wik and update cluster assignments (C-step). Following the cluster assignments from the previous step (P-step), we differentiate J w.r.t. μk and recompute the centroids. Therefore C-step can be stated as:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} \|x^i - \mu_k\|^2$$

$$\Rightarrow w_{ik} = \begin{cases} 1 & \text{if } k = argmin_j \|x^i - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

In other words, place the data point xi in the nearest cluster determined by its sum of squared distance from the cluster's centroid. Therefore, P-step can be stated as:

$$\frac{\partial J}{\partial \mu_k} = 2 \sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0$$

$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}}$$

which equates to recalculating each cluster's centroid to account for the new assignments.

As mentioned earlier, in order to design the model to detect anomalies from this uncertain data, first the data will be preprocessed, cleaned to impute the missing information with their mean and drop the rest. The respective attributes required for data analysis to understand the instances where the pump has broken down and how that reflects in the sensor readings. Finally, to evaluate the model K means clustering technique will be implemented and we will analyze the anomaly result as 0 for 'normal' and 1 for 'anomaly' and visualize the resulting anomalies.

Additionally we have referred to the paper [7] where Bin Jiang et al has proposed to implement interval skyline queries as a novel technique for time series analysis queries. The algorithm proposed by the researcher is as follows: for time series s, let s.max denote the maximum value of s in the base interval W, i.e., s.max = maxk∈W {s[k]}. Let s.min[i : j] denote the minimum value of s in interval [i : j] ⊆ W, i.e., s.min[i : j] = mink∈[i:j]{s[k]}. Using the maximum values and the minimum values in intervals of the time series, we can determine the domination relation between some time series without checking the details. We are exploring this approach in line with the dataset we have considered for the problem statement and project.

5. The Proposed Techniques

The dataset in this study refers to sensor readings of 53 sensors that were mounted on a pump to evaluate a variety of pump behaviors, and the data collection includes readings from those sensors. The data used to study the problem definition is a Time series sensor data with 200 thousand entries and three classes: Normal, Broken and Recovering conditions of the pump at different timestamps. It is a collection of observations that have been organized and are connected over time. The challenge while working on the dataset is that the dataset has missing values, empty columns and timestamp with incorrect data type. Hence, the data needs to be cleaned, preprocessed and checked for stationarity and autocorrelation before feeding them into the proposed algorithm to detect anomalies.

The literature referred to scope the implementation of is [7] B Jiang et al, Online Interval Skyline Queries on Time Series for probabilistic range query implementation,[8] Jacob, V Song et al, A Benchmark for Explainable Anomaly Detection over Time Series., this study is about interquartile range implementation approach and [9] Guansong Pang et al, Deep Learning methods for Anomaly Detection for understanding anomaly detection techniques.

The techniques proposed in this study to search anomalies in sensor data is as follows:

- Interquartile range is used to determine the computational threshold and calculate the outlier data. In this approach the complete data set is divided into four equal parts or quartiles. The distance between the quartiles is used to determine the IQR.
- K means clustering techniques aim to split n observations into k clusters, where each observation is assigned to the cluster that has the closest mean (also known as the cluster centroid or cluster centers), acting as a model for the cluster.
- Isolation Forest is an unsupervised detection method based on the isolation of outliers. It does not calculate either distance or density and therefore significantly reduces execution time and memory requirement. The isolation forest needs an Anomaly Score to have an idea of how anomalous a data point is. Its values lie between 0 and 1.

To design the model and detect anomalies from this uncertain data, first the data will be preprocessed, cleaned to impute the missing information with their mean and drop the rest. The respective attributes required for data analysis to understand the instances where the pump has broken down and how that reflects in the sensor readings. Finally, to evaluate the implemented models the anomaly results are analyzed as 0 for ‘normal’ and 1 for ‘anomaly’ and visualize the resulting anomalies.

The solution is proposed to be implemented using Python language and designed using a model including Python libraries. The aim is to detect anomalies right before the pump breaks down. This could be very valuable information for an operator to see and be able to shut down the pump properly before it actually goes down. The detailed approach regarding model implementation, prediction, evaluation and data-result visualization using tools and libraries is elaborated in this Section, Section 6, 7, and Section 8 respectively.

5.1 Scope of the Solution Space:

The scope of this project is limited to detecting anomalies in the 53 sensors of the selected pump and excludes other pumps and predicting the failures of the pump.

5.2 Constraints:

The Data set contains readings from a single pump hence the proposed may not be the best representation of all the pumps. Also we tried to include all the possible pumps that would require lots of computing power and will become a big constraint for visualization of 53 features at one instance.

5.3 Data:

The data set is loaded from https://www.kaggle.com/nphantawee/pump-sensor-data and it consists of 51 numerical features and categorical labels. Numerical features contain readings from used 51 different sensors that are used to monitor the condition of pumps. The label feature contains string values that represent normal, broken and recovering operational conditions of the pump.The data set represents 219,521 readings from 51 sensors.

5.4 Solution Approach:

We have first built a base model using IQR technique and later we have implemented two other unsupervised algorithms to compare their performance and accuracy. Please see the steps we have follow:

5.4.1 Data wrangling

We at first cleaned the data to make a process designed to transform raw data into more readily used formats. Such as removing duplicates, converting data to correct data types.

5.4.2 Exploratory Data Analysis (EDA)

After loading the data we perform Exploratory data analysis, we perform two types of analysis. Quantitative EDA fig (3) and Graphical EDA fig (4) as shown below:

```
[11]: df.describe().T
```

[11]:

	count	mean	std	min	25%	50%	75%	max
sensor_00	219521.0	2.371961	0.403223	0.000000	2.417187	2.455556	2.499826	2.549016
sensor_01	219521.0	47.590015	3.299447	0.000000	46.310760	48.133678	49.479160	56.727430
sensor_02	219521.0	50.863387	3.668355	33.159720	50.390620	51.649300	52.777770	56.032990
sensor_03	219521.0	43.746718	2.417856	31.640620	42.838539	44.227428	45.225693	48.220490
sensor_04	219521.0	590.629672	144.127709	2.798032	626.620400	632.754600	637.615723	800.000000
sensor_05	219521.0	73.397642	17.304487	0.000000	69.982320	75.578420	80.918750	99.999880
sensor_06	219521.0	13.500351	2.143252	0.014468	13.346350	13.628470	14.539930	22.251160
sensor_07	219521.0	15.841237	2.176726	0.000000	15.856480	16.167530	16.427950	23.596640
sensor_08	219521.0	15.198273	2.015901	0.028935	15.183740	15.451390	15.697340	24.348960

Fig:(3) Quantitative EDA

Gaphical EDA

Let's visualize the sensor readings across the entire 52 sensors and mark the pump's broken state in red color on the same graph

```
# Vizualize time series and the BROKEN state (red dots) in the same graph for each sensor
import warnings
# Extract the readings from BROKEN state and resample by daily average
broken = df[df['machine_status']=='BROKEN']
# Extract the names of the numerical columns
df2 = df.drop(['machine_status'], axis=1)
names=df2.columns
# Plot time series for each sensor with BROKEN state marked with X in red color
for name in names:
    sns.set_context('talk')
    _ = plt.figure(figsize=(18,3))
    _ = plt.plot(broken[name], linestyle='none', marker='X', color='red', markersize=12)
    _ = plt.plot(df[name], color='blue')
    _ = plt.title(name)
    plt.show()
```

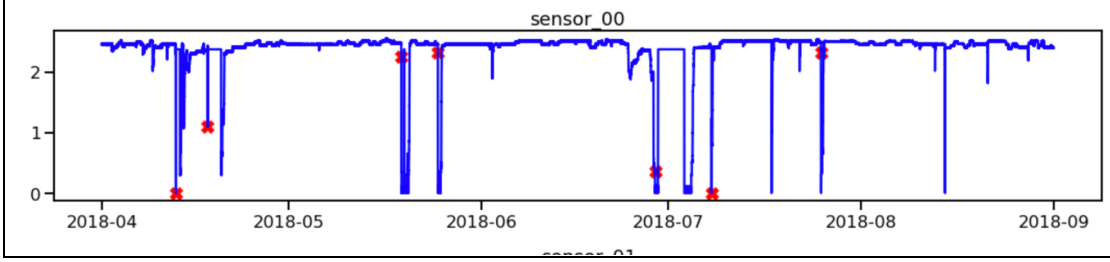
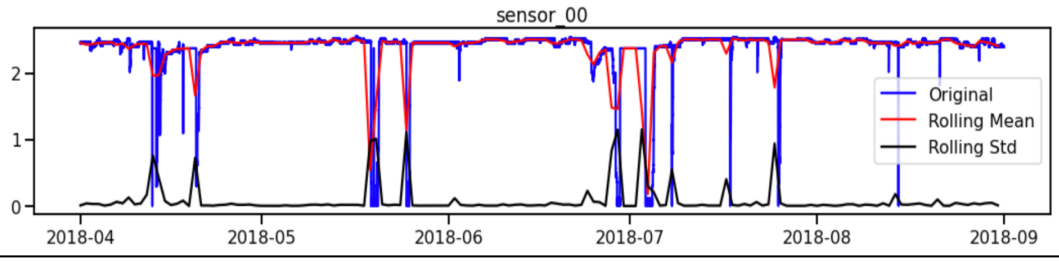


Fig:(4) Graphical EDA

More graphical representation of the above method is shared in the below experimental section. We perform two additional methods for EDA, Stationary and Rolling statistics fig (5).

```
In [14]:
# Resample the entire dataset by daily average
rollmean = df.resample(rule='D').mean()
rollstd = df.resample(rule='D').std()
```

```
In [15]:
# Plot time series for each sensor with BROKEN state marked with X in red color
for name in names:
    _ = plt.figure(figsize=(18,3))
    _ = plt.plot(df[name], color='blue', label='Original')
    _ = plt.plot(rollmean[name], color='red', label='Rolling Mean')
    _ = plt.plot(rollstd[name], color='black', label='Rolling Std' )
    _ = plt.legend(loc='best')
    _ = plt.title(name)
    plt.show()
```



Fig(5): Stationary and Rolling statistics

5.4.2 Pre-Processing and Feature Engineering

In the pre-processing phase we perform PCA for extracting more important features from the dataset for further use in training the model. Training the dataset with the current dataset would be very computationally expensive. hence the reason for reducing the dimensionality with PCA.

5.4.4 Modeling

- **Initially we created a base model (IQR) as mentioned above. As shown below in fig(6).**

Base Model: Detect Outliers Using the Interquartile Range (IQR)

Anomalies are defined as rare events that could be represented by the outliers in the data set. As an initial step, I want to apply a basic statistics technique to get the feel of the outliers present in this data set. Later, I will compare the results of the other models to the results from the Base Model for further model evaluation.

0: normal

1: anomaly

```
# outlier_lower = Q1 - (1.5*IQR)
# outlier_upper = Q3 + (1.5*IQR)
# Calculate outlier bounds for pc1
q1_pc1, q3_pc1 = df['pc1'].quantile([0.25, 0.75])
iqr_pc1 = q3_pc1 - q1_pc1
lower_pc1 = q1_pc1 - (1.5*iqr_pc1)
upper_pc1 = q3_pc1 + (1.5*iqr_pc1)
# Calculate outlier bounds for pc2
q1_pc2, q3_pc2 = df['pc2'].quantile([0.25, 0.75])
iqr_pc2 = q3_pc2 - q1_pc2
lower_pc2 = q1_pc2 - (1.5*iqr_pc2)
upper_pc2 = q3_pc2 + (1.5*iqr_pc2)
```

```
lower_pc1, upper_pc1
```

```
(-736.3329510286096, 222.73590494562214)
```

```
lower_pc2, upper_pc2
```

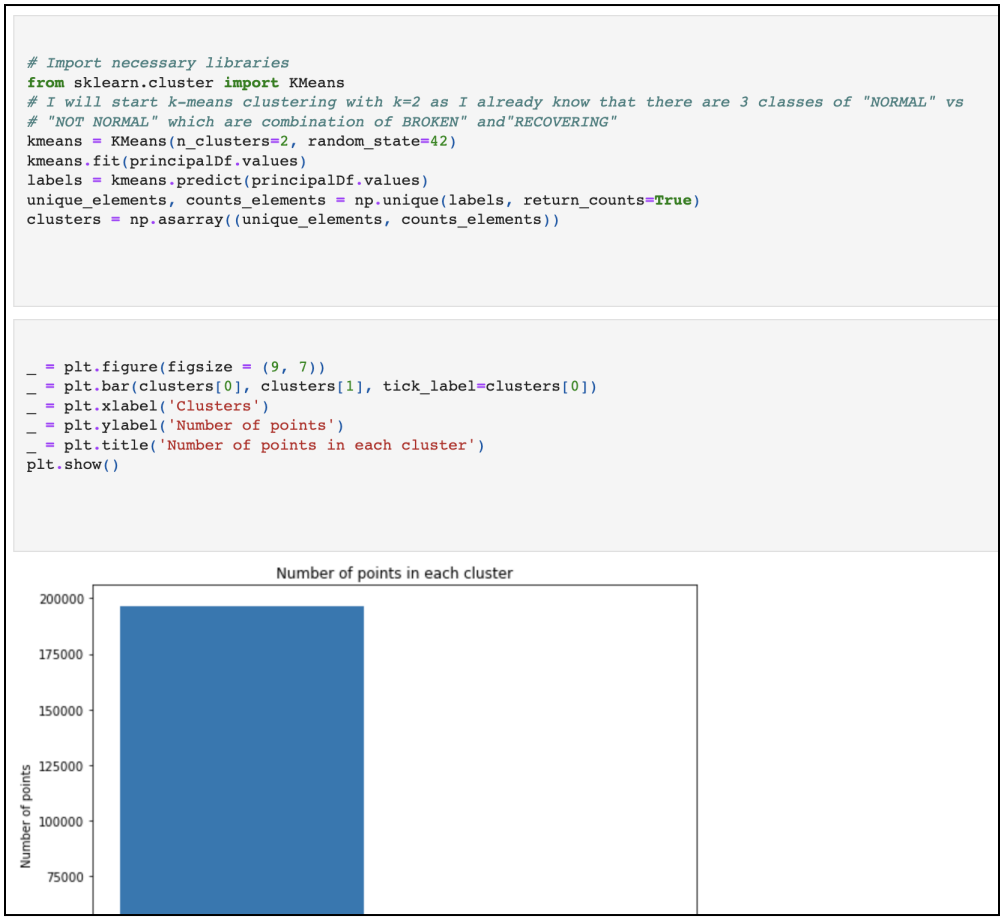
```
(-738.2364730735547, 683.9742726790346)
```

Fig(6): Code implementation of Base model (IQR)

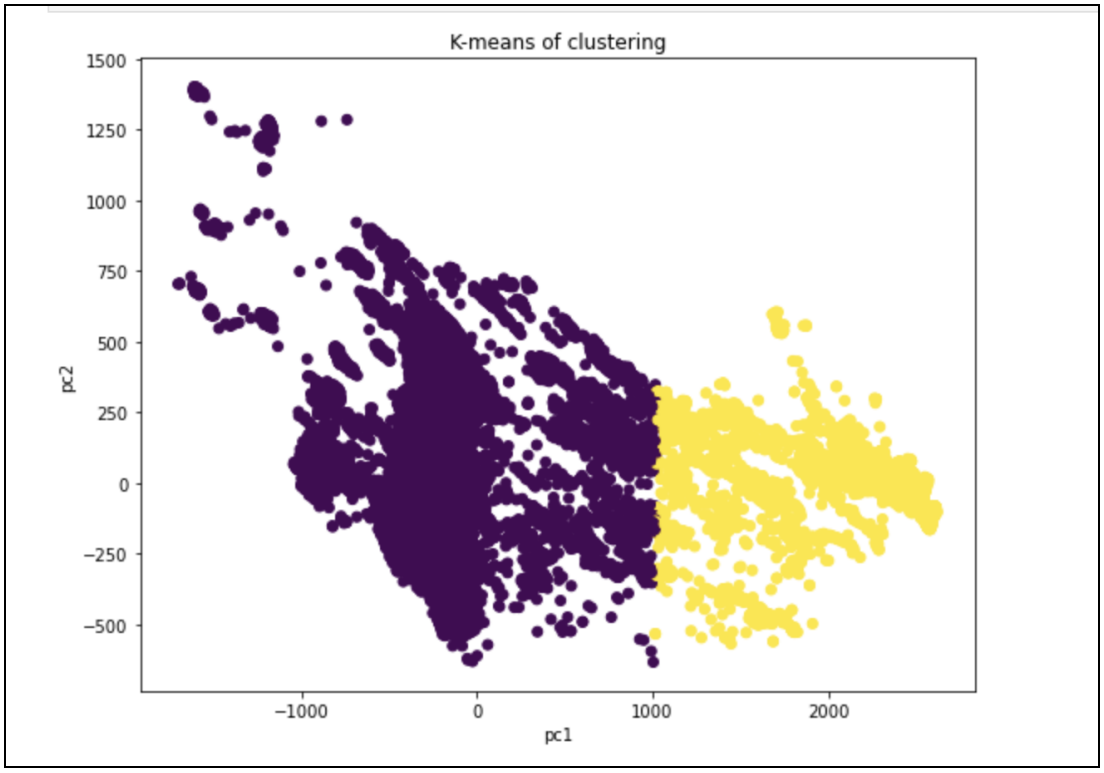
- **K-means clustering**
  - We calculate the distance between each point and its nearest centroid and the largest distance is considered to be the anomaly as shown in Fig(7), and Fig (8)
  - We set the minimum distance threshold to get the outlier**s.**
  - It contains 2 cluster 1 and 0 (0: normal, 1: anomaly)

```
# Import necessary libraries
from sklearn.cluster import KMeans
# I will start k-means clustering with k=2 as I already know that there are 3 classes of "NORMAL" vs
# "NOT NORMAL" which are combination of BROKEN" and"RECOVERING"
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans.fit(principalDf.values)
labels = kmeans.predict(principalDf.values)
unique_elements, counts_elements = np.unique(labels, return_counts=True)
clusters = np.asarray((unique_elements, counts_elements))
```

```
_ = plt.figure(figsize = (9, 7))
_ = plt.bar(clusters[0], clusters[1], tick_label=clusters[0])
_ = plt.xlabel('Clusters')
_ = plt.ylabel('Number of points')
_ = plt.title('Number of points in each cluster')
plt.show()
```



Fig(7): Code Implementation of K-means clustering



Fig(8): Visualization of outliers using K-means clustering

- **Isolation Forest**

Isolating the anomalies from the dataset. We have used the IsolationForest library of Python as shown below in Fig (9).

```
# Import IsolationForest
from sklearn.ensemble import IsolationForest
# Assume that 13% of the entire data set are anomalies
outliers_fraction = 0.13
model =  IsolationForest(contamination=outliers_fraction)
model.fit(principalDf.values)
principalDf['anomaly2'] = pd.Series(model.predict(principalDf.values))
```

Fig(9): Code implementation for Isolation forest

6. Visual Applications

We used public data in this project and it required some cleaning. So, to clean it and make it tidy we removed redundant columns, duplicates, handled missing values and converted data types to the correct ones. Fig 10 below has the first 10 rows of the data after cleaning-

date	sensor_00	sensor_01	sensor_02	sensor_03	sensor_04	sensor_05	sensor_06	sensor_07	sensor_08	sensor_09	...	sensor_47	sensor_48
2018-04-01 00:00:00	2.465394	47.09201	53.2118	46.310760	634.3750	76.45975	13.41146	16.13136	15.56713	15.05353	...	38.194440	157.9861
2018-04-01 00:01:00	2.465394	47.09201	53.2118	46.310760	634.3750	76.45975	13.41146	16.13136	15.56713	15.05353	...	38.194440	157.9861
2018-04-01 00:02:00	2.444734	47.35243	53.2118	46.397570	638.8889	73.54598	13.32465	16.03733	15.61777	15.01013	...	38.194443	155.9606
2018-04-01 00:03:00	2.460474	47.09201	53.1684	46.397568	628.1250	76.98898	13.31742	16.24711	15.69734	15.08247	...	38.194440	155.9606
2018-04-01 00:04:00	2.445718	47.13541	53.2118	46.397568	636.4583	76.58897	13.35359	16.21094	15.69734	15.08247	...	38.773150	158.2755

Fig(10): Sensor dataset readings

After data cleaning the data analysis has been done and a graph was formed to visualize the details for the sensors. Fig 11 below is the graph for sensor_00.

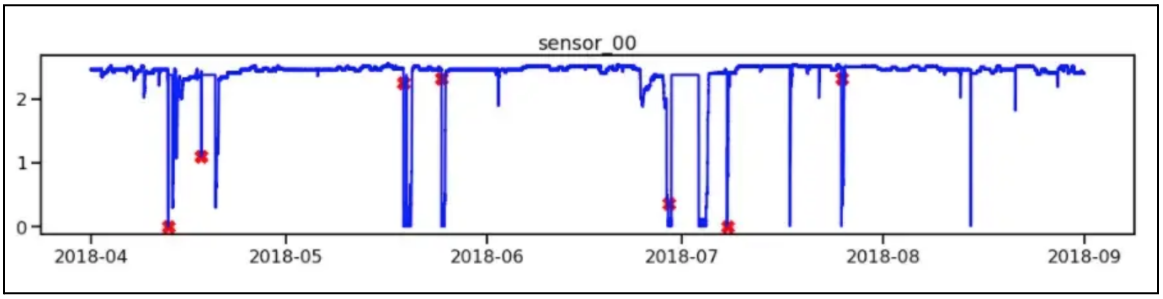


Fig (11): Graph for sensor_00

In this graph, the red marks show the broken state of the pump and it's overlapping with disturbances of the sensors.

In the next step the visualized inspection of the stationarity of each feature in the data set has been done. Stationarity means the behavior where the mean and standard deviation of the data changes over time, so the data without that behavior is considered as stationary. Fig (12) below is the graph showing the **stationarity** in the time series data-

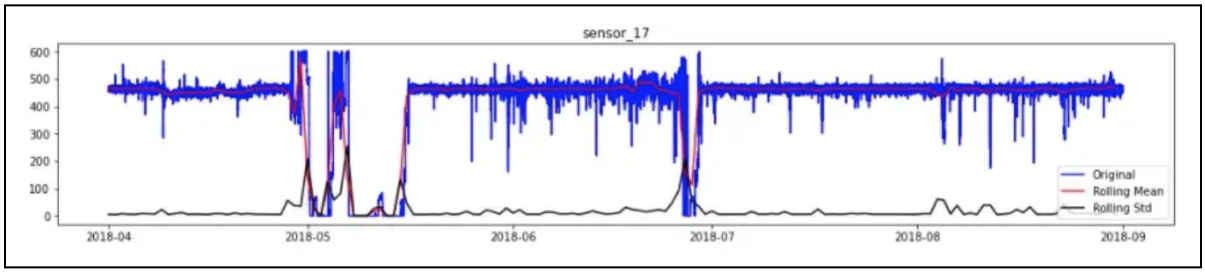


Fig (12): Graph showing stationarity in time series data

This graph shows the readings from sensor_17 and the data is looking pretty stationary which means that the standard deviation will not change over time except the downtime of the pump.

After this, pre-processing and dimensionality reduction of the data has been done and for that **Principal Component Analysis(PCA)** technique has been used. Firstly, we scaled the data and then PCA was performed on it to find the most important principal components based on the inertia. Fig 13 below is the graph showing the most important two components-

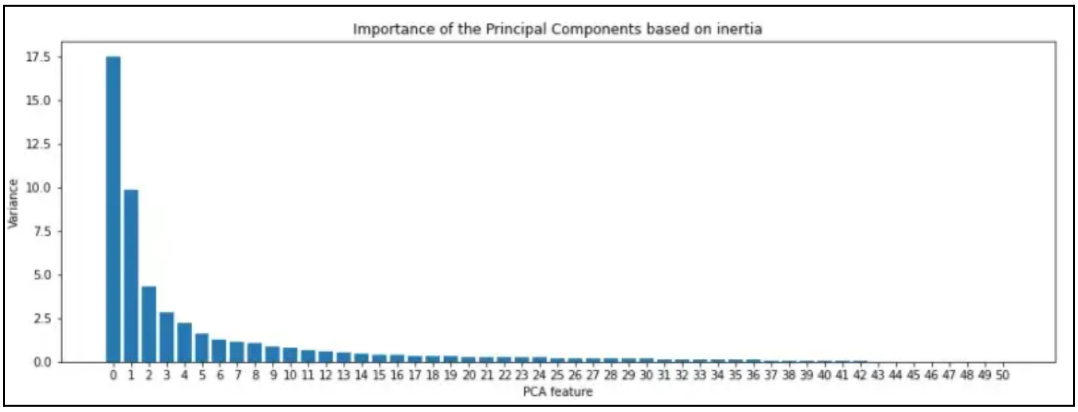


Fig 13: PCA feature graph

After that, we visually verified that there is no **autocorrelation**. Fig 14 below is the graph which shows that it is not autocorrelated-

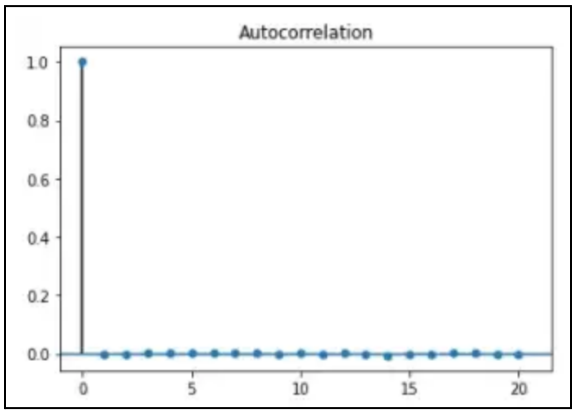


Fig 14:Autocorrelation graph

In the next step, the modeling of the data has been done by applying the algorithms like Interquartile Range(IQR), K-means clustering, Isolation forest to detect the anomalies.

After applying the technique of **Interquartile Range(IQR)** to find the anomalies, Fig 15 the graph will look like as follows and the anomalies will be marked in red color-

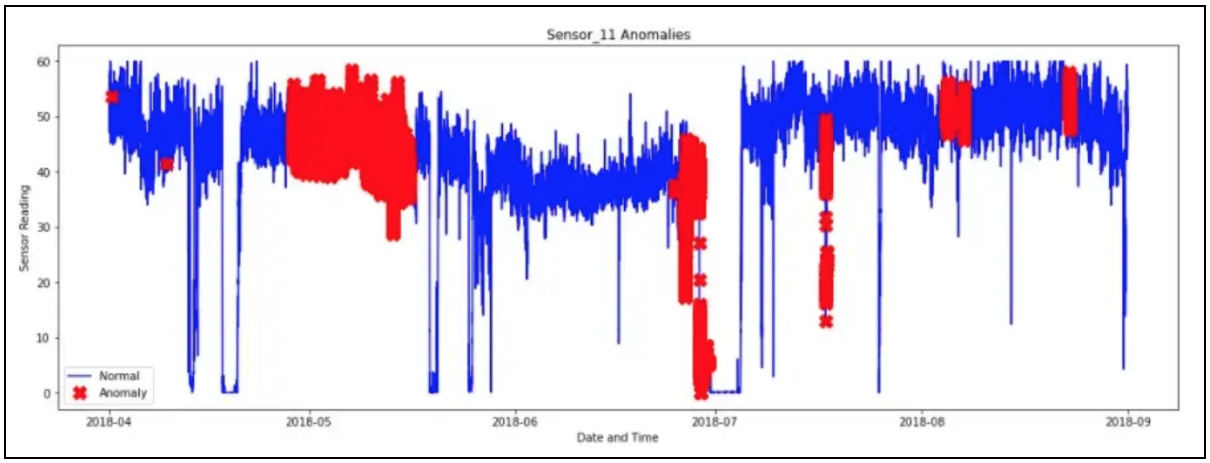


Fig 15: Anomalies visualization using Interquartile range

After applying the technique of **K-means clustering** to find the anomalies, Fig 16 the graph will look like as follows and the anomalies will be marked in red color-

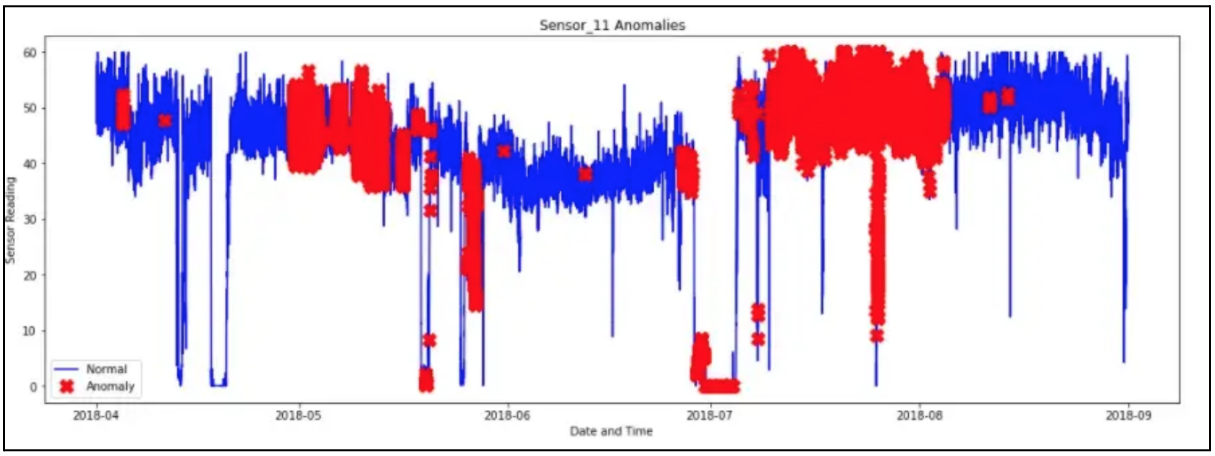


Fig 16: Anomalies visualization using K means Clustering

After applying the technique of **Isolation forest** to find the anomalies, Fig 17 the graph will look like as follows and the anomalies will be marked in red color-

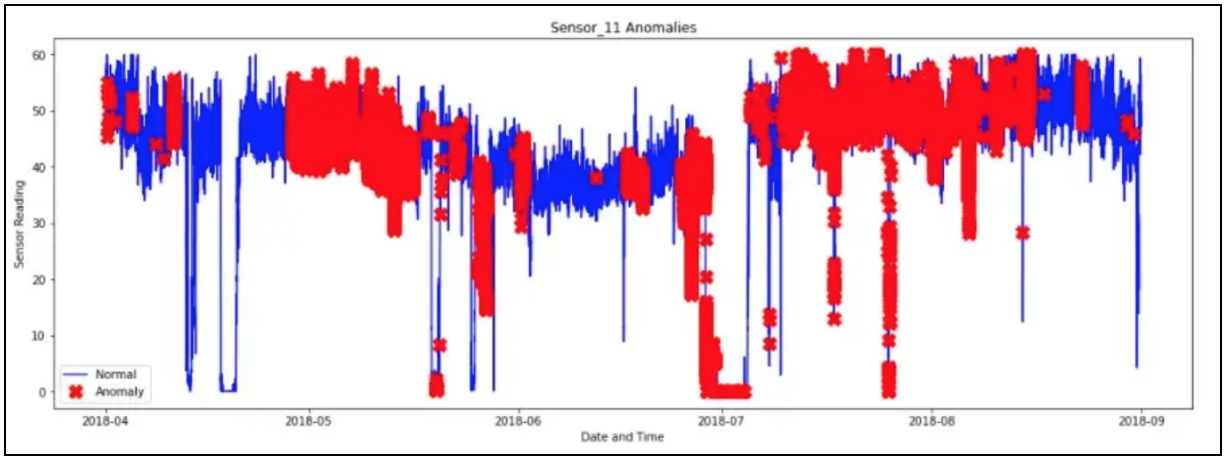


Fig 17: Anomalies visualization using Isolation Forest

7. Experimental Evaluation

7.1 Experimental settings

7.1.1 Descriptions of real data sets

In our project, we analyzed 220320 sets of time series data from 56 different sensors, among which different sensors detected different types of data, including but not limited to: 1. The pH probe was used to collect the pipeline pH value; 2. The water pressure probe is used to collect the pressure value of the pipeline; 3, the water temperature probe is used to collect the water temperature of the pipeline, using the PTC temperature sensor; 4, simulate the water pressure value detected by the water pressure sensor; 5. Electromagnetic flowmeter is used to detect the flow rate of pipeline liquid; 6, the environment detector is used to detect the environment temperature and humidity; 7. Power parameter collector Used to collect the voltage and current of single-phase or three-phase electricity.

7.1.2 Competitors of the model

In order to test the advantages and disadvantages of different algorithms for anomaly detection, the following four algorithms are selected for comparison in the experiment, Interquartile range（which is IQR）, original K-mean clustering, Isolation Forest and improved K-mean clustering.

We used the data from sensor 11, trained the data using the four algorithms mentioned earlier, and visualized the results. The blue part is considered as normal data by the algorithm, while the red part is detected as abnormal data by the algorithm. Below are the visual comparison results we

produced using the above four algorithms. (fig.18 Visual comparison results)

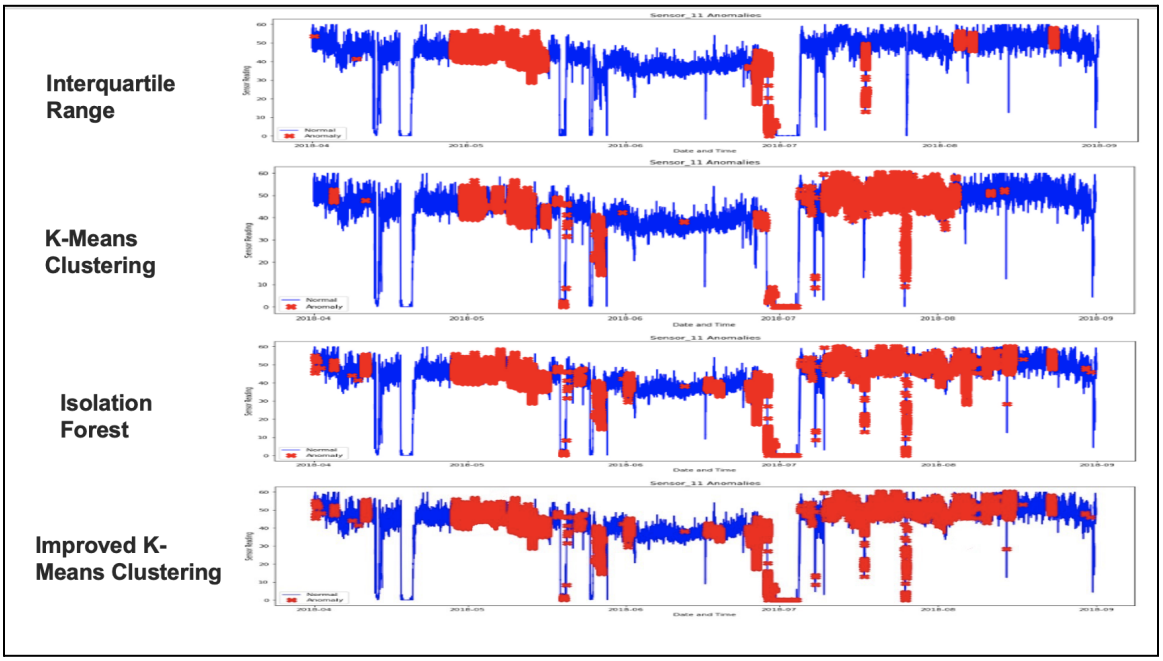


Fig 18. Visual comparison results

7.2 The performance report

7.2.1 Comparison of quantity of abnormal data detection

We counted the number of normal and abnormal points based on these four algorithms, It is interesting to see that all four models detected a lot of the similar anomalies. Just by visually looking at the above graphs in the last slide, one could easily conclude that the Isolation Forest might be detecting a lot more anomalies than the other three. However, the following tables show, on the contrary, that IQR is detecting far more anomalies than others. (table. Comparison of quantity of abnormal data detection)

	Normal	Anomaly
IQR	189,644	29,877
K-means Clustering	190.984	28,537
Isolation Forest	190,983	28,538
Improved K-means Clustering	189,688	29833

7.2.2 Comparison of model accuracy

	TPR	FPR
IQR	82.5%	11.5%
K-means Clustering	78.9%	15.5%
Isolation Forest	75.8%	14.2%
Improved K_means Clustering	83.8%	10.2%

When we calculate the equation using accuracy (Mentioned in Chapter 3), the comparison results are shown in the table below. In addition to counting the amount of abnormal data,, We also have to consider whether the model calculation is accurate. We introduced the formula on the left to calculate the accuracy of the model. TPR is used to measure the probability rate that outliers are correctly labeled, and FPR is used to measure the probability rate that positive and constant values are incorrectly labeled. TP represents the correctly detected outliers; TN

represents the undetected outlier. FP indicates that normal values are detected as outliers; FN indicates that the normal value is detected as normal.

After comparing the data, we found that the TPR of the improved k algorithm was higher than that of the other three algorithms, and the FPR was also lower than that of the other three algorithms.

By improving the initial clustering center random selection process of the traditional K-means algorithm (Mentioned in Chapter 3), an anomaly detection algorithm based on improved K-means clustering was proposed. When selecting the initial clustering center, firstly calculate the tightness of all data points, exclude the outlier regions, and evenly select K initial centers in the place of tight data, so as to avoid the defect that random selection is easy to lead to local optimization. By optimizing the selection process, the algorithm is closer to the real cluster center before iteration, reducing the number of iterations and improving the clustering quality and anomaly detection rate. Experimental results show that the improved algorithm is better than the traditional K-means algorithm in clustering performance and anomaly detection.

8. Future Work

8.1 Test the model with more data sets

The generality of the model needs to be tested in more data sets. Only through continuous research and comprehensive optimization of the model can we further improve the accuracy and calculation speed of the model and realize the accurate detection of abnormal data. Therefore, we will try to use the above model to train uncertain data in different fields.

8.2 Optimize the models

The uncertainty of sensor data is caused by a variety of reasons. In future work, we will try to introduce more parameters to optimize our detection model.

9.Conclusion

As a conclusion, In our projects, we compares three original anomaly data detection algorithms, optimizes the selection process of the initial clustering center of the K-means algorithm by evenly selecting the data compact region and avoiding the outlier region, and proposes an anomaly detection algorithm based on the improved K-means clustering. Experiments show that the improved K-means clustering algorithm has more advantages than other algorithms in anomaly detection.

10. References

[1] Y. Jin, C. Qiu, L. Sun, X. Peng and J. Zhou, "Anomaly detection in time series via robust PCA," 2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE), 2017, pp. 352-355, doi: 10.1109/ICITE.2017.8056937.

[2] Jiawei Yang, Susanto Rahardja, and Pasi Fränti. 2019. Outlier detection: how to threshold outlier scores? In Proceedings of the International Conference on Artificial Intelligence,

Information Processing and Cloud Computing (AIIPCC '19). Association for Computing Machinery, New York, NY, USA, Article 37, 1–6. https://doi.org/10.1145/3371425.3371427

[3] Jacob, V., Song, F., Stiegler, A., Rad, B., Diao, Y., & Tatbul, N. (2021). Exathlon: A Benchmark for Explainable Anomaly Detection over Time Series. Proc. VLDB Endow., 14, 2613-2626.

[4] S Cherednichenko, V Hautamäki, T Kinnunen, I Kärkkäinen and P Fränti (2005). Improving k-means by outlier removal. Scandinavian Conf. on Image Analysis (SCIA'05), 978-987.

[5] Zhangyu Cheng, Chengming Zou, and Jianwei Dong. 2019. Outlier detection using isolation forest and local outlier factor. In Proceedings of the Conference on Research in Adaptive and Convergent Systems (RACS '19). Association for Computing Machinery, New York, NY, USA, 161–168. https://doi.org/10.1145/3338840.3355641

[6] Martí, L.; Sanchez-Pi, N.; Molina, J.M.; Garcia, A.C.B. Anomaly Detection Based on Sensor Data in Petroleum Industry Applications. Sensors 2015, 15, 2774-2797. https://doi.org/10.3390/s150202774

[7] B. Jiang and J. Pei, "Online Interval Skyline Queries on Time Series," 2009 IEEE 25th International Conference on Data Engineering, 2009, pp. 1036-1047, doi: 10.1109/ICDE.2009.70.

[8] Jacob, V., Song, F., Stiegler, A., Rad, B., Diao, Y., & Tatbul, N. (2021). Exathlon: A Benchmark for Explainable Anomaly Detection over Time Series. Proc. VLDB Endow., 14, 2613-2626.

[9] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. 2021. Deep Learning for Anomaly Detection: A Review. ACM Comput. Surv. 54, 2, Article 38 (March 2022), . https://doi.org/10.1145/3439950