UNIVERSITY
OF GÄVLE

FACULTY OF ENGINEERING AND SUSTAINABLE DEVELOPMENT
Department of Electrical Engineering, Mathematics and Science

Prediction of industrial machine failure by analysing anomalies

Md Abdur Rahman Akash

January 2022

Student thesis, Advanced level (Master's degree, two years), 30 HE
Electronics/Automation

Supervisor: Dr Reza Salim (University of Gävle)
Examiner: Dr. José Chilo (University of Gävle)

Acknowlegments

This master thesis was written by me at the department of Electrical Engineering, mathematics, and science at the University of Gävle (HiG), Gävle, Sweden.

Firstly, I would like to thank my supervisor Dr. Reza Salim, assistant professor at the University of Gävle (HiG) for his patient guidance, encouragement, advice and most importantly giving me the space to think critically that he provided for the successful preparation and completion of this thesis work. Whenever I stuck on something he was there for me to give me necessary guidelines and positive thoughts which I appreciate a lot. He always gave me the opportunity to work independently allowing a fresh and highly research-oriented master thesis topic.

Secondly, I would also like to thank my examiner Dr. José Chilo (University of Gävle), for being the examiner of my master thesis paper.

Furthermore, I would also like to thank Professor Edvard Nordlander for his valuable advice during the courses and information on getting a master thesis.

In addition, I would like to thank my classmates with whom I shared a lot of moments during my university years, and this will not be forgotten.

Finally, I would like to thank my mother, my uncle, my sister for supporting me all the time.

January 2022, Gävle.

Md Abdur Rahman Akash

Abstract

The sudden downtime and unplanned maintenance not only drastically increase the maintenance cost but also decreases the production capacity for the manufacturer industries. This is because the machines on these industries fail suddenly and totally stop the production as the machine should be fixed by maintenance before it can run again. To deal with it, several maintenance techniques have been adopted. But as soon as an automated maintenance technique comes in named predictive maintenance, the future machine failure can be predicted. To perform this prediction, a synthetic dataset is used that is taken from 100 industrial machines. From this dataset, the simulated sensor data, error, and failure history have used to calculate the probability of error and failure during the time period of an anomaly. This probability is calculated by the basic probability equation. In addition, the sum of the calculated probability of error and failure, give the intuition about the most relevant sensor data for a machine. This relevant sensor data is then used as response for the prediction with gaussian process regression algorithm. This prediction of response is shown for machine number 85 which is the most important from all 100 machines as this machine is very sensitive to any of the 4 sensor anomalies. Then, the sum of probability can be coherent with the anomaly on the predicted response which is the most relevant sensor data. This defines that the machine is in high risk of experiencing machine failure and thus the machine should be fixed by adopting maintenance. In contrast, the opposite is also true for low probability of error and failure for an anomaly on the predicted response. To evaluate the performance of the algorithm, four statistical metrics are used among which three matric is to estimate the errors and the other one is the correlation coefficient between the actual and predicted data.

List of figures

List of tables

Table of contents

1 Introduction

1.1 Background

In recent years, the use of machines increased rapidly by the industries. These machines depreciate in respect to time and at a time the machine completely stop by experiencing failure. For this reason, the industries expense a lot of money to fix the failure by doing maintenance of the machine [1].

According to a recent survey in United State of America (USA) in 2020, the industries production capacity reduces 20% for the poor maintenance technique. For this poor maintenance technique, $616.1 billion spent by the global maintenance market. This will increase 2.19% every year till 2026 which is $701.3 billion. Specifically, in the automotive industries, the cost for unplanned maintenance is $22,000 per minute. This means that the cost increases 3 to 10 times for the sudden downtime and unplanned maintenance [2].

To fix the machines, several maintenance techniques adopted by the industries. But as soon as the fourth industrial revolution has been experienced by the world, artificial intelligence comes in and make this maintenance technique automated. This maintenance technique predicts the future failure by analyzing the historical data of the machine and do maintenance when requires. This technique is known as predictive maintenance [3].

There are several advantages of using predictive maintenance. They are provided as follows [4]:

- The quality of the service will be increased as the remote diagnosis and fixing failure decrease the performance of the appliances.
- Safety precaution comes in as it will avoid sudden failure.
- There will be a certain threshold of the components efficiency that it will not go beyond this efficiency threshold.
- The lifetime of the component will be increased, and the repairing cost will be decreased.
- The customers loyalty will be increased for the company.
- The electric components can be recycled and reused ensuring the identification of any components warranty violations.
- The sensor has reduced the number of humans dealing with the maintenance and thus the testing expenses has also been reduced.
- As the components will run only the healthy status, it ensures the increased lifetime of the components.

Similarly, a survey reflects that, the maintenance cost reduces & increases production capacity by 25 – 35% and eliminates downtime by 70 – 75% with the adoption of predictive maintenance by the industries. This prediction of failure and maintenance is possible only after the data is available. This data is mainly of two types. One is the real data from real machine from a real industry and the other one is the synthetic data which is usually generated virtually. By using both of this data there are several kinds of applications predictive maintenance deals. These are the diagnosis of fault, prediction of fault, detection of anomalies on the sensor data, prediction of time to failure and remaining useful life predictions. During all this prediction, the major challenge comes about the data quality. The reason behind is that for a new machine, it is nearly impossible to generate failure data which is very important for the machine learning algorithm to predict failure. Thus, synthetic data is a good approach to overcome this challenge [5].

1.2 Thesis scope

To predict failure and adopt maintenance, this thesis work removes noise by filtering and distinguishes anomalies by using synthetic data [6]. As the anomalies play major role for failure prediction and all the anomalies did not lead to machine error and failure, the probability of machine error and failure is calculated for each anomaly on the sensor data. This calculated probability together will then be used to figure out the most relevant sensor data which is the response of a machine. Finally, with the use of machine learning, the response is predicted and coherent with the calculated probability. This probability will then define for an anomaly on the predicted response either the machine is high or low risk of experiencing machine error and failure or not.

In addition, to be responsive of the challenge, the error and failure historical data plays vital role. By checking these two datasets, it is seen that for a machine there might be one or several months that no error and failure occurred rather there are multiple error and failure on another months. For this reason, the monthly data is used to ensure the data quality both with error & failure data and train the machine learning algorithm accordingly.

1.3 Motivation, goal, and scientific tasks

Increased sudden downtime and maintenance cost has become one of the major challenges for sudden machine failure to the industries all over the world. For this reason, it is becoming more and more important to avoid this sudden machine failure. There were several maintenance techniques adopted by the industries but none of them could meet the demand of the industries. Therefore, the prediction of the machine failure would be great in order to reduce suddden downtime and maintenance cost. For this prediction, machine learning provides several suitable techniques. Thus, the prediction of machine failure can be used to schedule the maintenance time which refers as predictive maintenance and that motivates me a lot.

To deal with the similar manner, microsoft azure had provided a syntethic dataset which is not taken from any real industries or machines. The dataset provides information about the machine error, failure, maintenance, age, and sensor data. As soon as any machine experiences machine error and failure, there is an anomaly seen on the sensor data. Hence, anomalies play vital role for the error and failure prediction.

Thus, the main goal of this thesis work is to predict machine failure by checking the probability of error and failure for an anomaly (which distingusihes healthy and faulty behavior of a machine) on the most relevant sensor data and do maintenance of the machine. To meet this main goal, there are three process performed as followed:

- Find the probability of error during the time period of an anomaly.
- Find the probability of failure during the time period of an anomaly.
- Predict anomalies on the most relevant sensor data and relate to the sum of the calculated probability of error and failure of a machine.

Here, anomalies means the anomalies on the voltgae, rotation, pressure and vibration sensor data. Thus, the scientific task of this thesis work is to calculate these three process.

Finally, this prediction can be helpful for the industries that deal with predictive maintenance to reduce sudden downtime and maintenance cost.

1.4 Thesis structure

This thesis work consists of five chapters.

Chapter 1-Introduction: The Background, thesis scope, goal, motivation & scientific tasks.

Chapter 2-Theory: Moving average filter, probability equation, standard scale, gaussian process regression, and statistical metric.

Chapter 3-Process: Briefly explains about the dataset, data preprocessing, probability of errors and failure calculation and prediction of response.

Chapter 4-Results: Presents the result of the probability of errors and failures during the time period of an anomaly and the predicted response data.

Chapter 5-Discussion: Provides discussion about the process and results performed.

Chapter 6-Conclusion: Provides the summary of this thesis work and recommend about the future work.

2 Theory

2.1 Moving average filter

Filtering is an essential tool to eradicate noise. Thus, to eradicate noise, the analysis needs to be done on the entire dataset. The analysis is to take the average of the neighboring elements. The neighboring number of elements along which the averaging is done, depends upon the application on where it is using. Thus, it is called the rolling mean or moving average filter. It is also said that if the number of neighboring elements is odd, then it will be centered around the current position. furthermore, the even element will lead the dataset to be centered around the current and previous element [7], [8].

Additionally, it will take M samples of input data at the same time and takes the average of those M sampled of input data and a unique output is calculated. In this way, the noise is reduced. Moving average filtering is a kind of convolution type and mathematically it can be written as for that M samples as,

$$Y[N] = \frac{1}{M} \sum_{K=0}^{M-1} X[N-K] \tag{1}$$

Thus, the noise reduction is done by filtering as well as the signal is also smoothed [9], [10].

2.2 Standard scale

To analyze the condition of a machine, there are several sensors fitted on the machine. This sensor data provides the sensor values on different units. That means they are not in the same scale. For this reason, it is required to extract all the different scale sensor values into the same scale so that a machine learning algorithm can be trained faster and convergence with better results [11].

There are several scaling techniques of the sensor data. Among them, the standard scaling, scale the sensor data with a mean value of zero and standard deviation of one. This can be done by using the following equation:

$$x_{scaled} = \frac{x - \mu_x}{\sigma_x} \tag{2}$$

Here, μ_x is the mean of the input data and σ_x is the standard deviation of the input data [12].

2.3 Probability equation

In statistics, probability is defined as the possible outcome that might occur based on the past historical information. That means probability is calculated based on the experiment or past historical data. In addition, the event that occurs from the historical information plays vital role. Thus, by this definition, the impossible outcome has probability of 0 and the possible outcome has probability of 1. To be deterministic to a probability of any event, the range of that probability must be between 0 and 1. Thus, the applied probability must satisfy the following condition:

$$\mathbf{0 \leq P(Event) \geq 1} \tag{3}$$

From equation (2), it is seen that if the probability of an event is less than 0 or greater than then the condition is not followed and hence the probability is not accurate. To satisfy this condition, the basic probability equation can be applied and calculate the possibility of occurrence of an event. Thus, the basic probability equation is as follows:

$$\mathbf{P(Event)} = \frac{\mathbf{|S|}}{\mathbf{|N|}} \tag{4}$$

Here,

S is the number of events occurred.

N is the number of total events.

P(Event) is the probability of any events.

From equation (3), it is seen that the modulus is used. This is because to make sure about the positive outcome of an event, as an event of occurrence cannot be negative [13].

2.4 Machine learning

Machine learning is a sub field of artificial intelligence. It is a computer algorithm that gets knowledge and learns from the historical information automatically. It is used in vast of applications that includes self-driving cars, voice & face recognition, industrial machinery failure prediction, medical diagnosis and so on.

Machine learning can be divided into two types. One is the supervised learning and the other one is the unsupervised learning that indicates in the figure below:

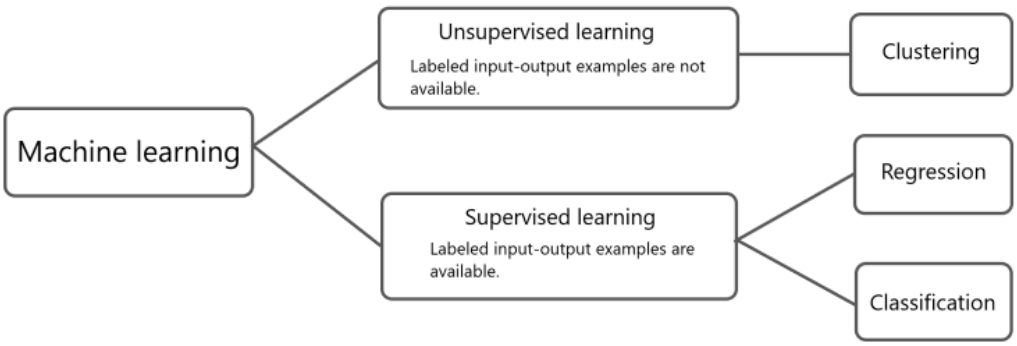

Figure 2.1. Machine learning techniques.

From fig. 2.1. it is seen that unsupervised learning is used when there is not input-output information in the data and thus it will cluster the output in different groups upon the application where it is using. In contrast, supervised learning is used when there is both input-output information on the data. With this, the prediction can be performed. Based on this, supervised learning can be classified further on 2 types. One is the classification techniques which will be used if the output is discrete, and the regression is used if the output is continuous [14].

From the regression supervised machine learning algorithm, gaussian process regression is one of them which is massively used in different application. For the normally distributed data, this algorithm provides a better response [15].

2.4.1 Gaussian process regression

Gaussian process is a stochastic process. This usually can be specified by the mean function m(x) and the covariance function $k(x_i,x_j)$. The equation of a gaussian process is given by:

$$\mathbf{f(x)} \sim \mathbf{GP}\,(\mathbf{m(x)},\mathbf{k}(\mathbf{x_i},\mathbf{x_j})) \tag{5}$$

But the generalized gaussian process regression can be defined as follows:

$$\mathbf{y} = \mathbf{f(x)} + \boldsymbol{\varepsilon} \tag{6}$$

Here, ε is the noise. On the gaussian process regression, this noise must consider with the mean function and covariance function. If this noise follows an independent deviation, then the gaussian process with the noise can be written as follow:

$$\mathbf{y} \sim \mathbf{GP}\,(\mathbf{m(x)},\, \mathbf{k}(\mathbf{x_i},\mathbf{x_j}) + \boldsymbol{\sigma}_\mathbf{n}^\mathbf{2}\partial_{\mathbf{ij}}) \tag{7}$$

Here, σ_n^2 is the variance and ∂_{ij} is the Kronecker delta.

During the modeling, gaussian process uses the Bayesian principle with some training data and target output. Gaussian process usually has the joint gaussian distribution. With that there is output both from the training and test set denoted by y and y_*. Thus, it is given by:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathbf{N}\left(\mathbf{0},\begin{bmatrix} \mathbf{K(X,X)} + \boldsymbol{\sigma}_\mathbf{n}^\mathbf{2}\mathbf{I} & \mathbf{K(X,x_*)} \\ \mathbf{K(x_*,X)} & \mathbf{K(x_*,x_*)} \end{bmatrix}\right) \tag{8}$$

Initially, the prediction can be done with gaussian process regression with the following equation:

$$\mathbf{y_*}|\mathbf{X},\mathbf{y},\mathbf{x_*} \sim \mathbf{N}(\overline{\mathbf{y_*}},\mathbf{cov(y_*)}) \tag{9}$$

Here,

$$\overline{\mathbf{y_*}} = \mathbf{K_*^T}\,\mathbf{K^{-1}y} \tag{10}$$

$$\mathbf{cov(y_*)} = \mathbf{K(x_*,x_*)} - \mathbf{K_*^T K^{-1} K_*} \tag{11}$$

On the above equation, $K = K(X,X) + \sigma_n^2 I$. Thus, the mean function and covariance function can define the target value y_* and the test input X_*.

The covariance function on gaussian process regresion is equivalent to the kernel function. The kernel function of the exponential equation can be given by the following equation:

$$K(x_i, x_j|\theta) = \sigma_n^2 exp\left(-\frac{r}{\sigma_l}\right) \quad (12)$$

Here, σ_l is the length scale of the kernel function.

$$r = \sqrt{(x_i - x_j)^T(x_i - x_j)} \quad (13)$$

This value of r can be calculated with the euclidian distance between the 2 data point [16],[17].

2.5 Statistical metrics

There are several statistical metrics that can evaluate the performance of a regression algorithm. Specifically, for the gaussian process regression, the prediction is done based on the covariance function that means the kernel function and the training sets. This model will then give the uncertainty of the predicted value which is the covariance function. This is the foremost benefit of using a gaussian process regression. The model is thus providing the desired predicted response with a stable model performance and gets the accuracy of the predicted values. For doing this, the possible statistical metric is the RMSE, MAE, RAE, and R^2 [16],[17].

2.5.1 Root mean square error (RMSE)

Root mean square error is one of the popular techniques for evaluating the model performance of a regression algorithm. It is well known if the RMSE value is less than 0.20, then it can be said that the model gives a better response. This model will represent the square's average difference between the predicted response and the actual response. This can be done by using the following equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((Predicted)_i - (actual)_i)} \quad (14)$$

Here, n is the number of observation set[18].

2.5.2 Mean absolute error (MAE)

The mean absolute error usually calculates the average magnitude difference of the errors between the actual data and predicted data. The lower the value of MAE the better the model is, as there is less error between the actual data and predicted data. This can be done by using the following equation:

$$\mathbf{MAE} = \frac{\mathbf{1}}{\mathbf{n}} \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{|(Actual)_i - (Predicted)_i|} \tag{15}$$

Here, n is the number of observation set [19].

2.5.3 Relative absolute error (RAE)

The relative absolute error usually calculates the ratio of the mean error (residual) to the mean absolute error. If the ratio is close to zero, then the model provides better results. In contrast, the ratio greater than results a poor model. This can be done by the following equation:

$$\mathbf{RAE} = \frac{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{((Actual)_i - (Predicted)_i)}}{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{|(Actual)_i - (Predicted)_i|}} \tag{16}$$

Here, n is the number of observation set [20].

2.5.4 Correlation coefficient (R^2)

The correlation coefficient (R^2) will give the intuition about how much the predicted and actual observation set are related. With this, the higher the R^2 value is, the higher the predicted and actual data are related. That means the model is predicting well. This can be done by using the following equation:

$$\mathbf{R^2} = \mathbf{1} - \frac{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{((Actual)_i - (Predicted)_i)^2}}{\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{((Actual)_i - mean(Actual)_i)^2}} \tag{17}$$

Here, n is the number of observation set [20].

3 Methods

This thesis works aims at predictive maintenance. For doing this, the probability of failure for any of the anomalies is calculated, predict the response of the future sensor data by using gaussian process regression algorithm, and relate both to predict the future failure. The algorithm and the calculation are implemented in MATLAB and the code is provided in the appendix section.

3.1 Dataset

Microsoft azure had provided synthetic data for predictive maintenance. This synthetic data contains 5 individual historical information as explained below:

- PdM_maint.csv contains the information of maintenance of all the 100 machines for 1 year.
- PdM_errors.csv contains the information of errors of all the 100 machines for 1 year.
- PdM_failures.csv contains the information of failures of all the 100 machines for 1 year.
- PdM_machines.csv contains the information of age and model number of all the 100 machines.
- PdM_maint.csv contains the information of maintenance of 100 machines for 1 year.
- PdM_telemetry.csv contains the voltage, rotation, pressure, and vibration sensor values for all the 100 machines for 1 year.

For the failure prediction by the algorithm, the PdM_errors.csv, PdM_failures.csv, and PdM_telemetry.csv dataset are used.

3.1.1 PdM_errors.csv

This dataset has the information of the errors for 100 machines and there are 5 errors faced by the machines. The information is provided based on the year, month, day, hour, minutes, and seconds. The information is for the year of 2015 started from 1st January at 06:00:00 and ends on 1st January of 2016 at 06:00:00. This information is provided for each month and day. There is also hour information, but minutes and seconds remains the same. The below figure shows the plot of 100 machine errors based on the year, month, and day.

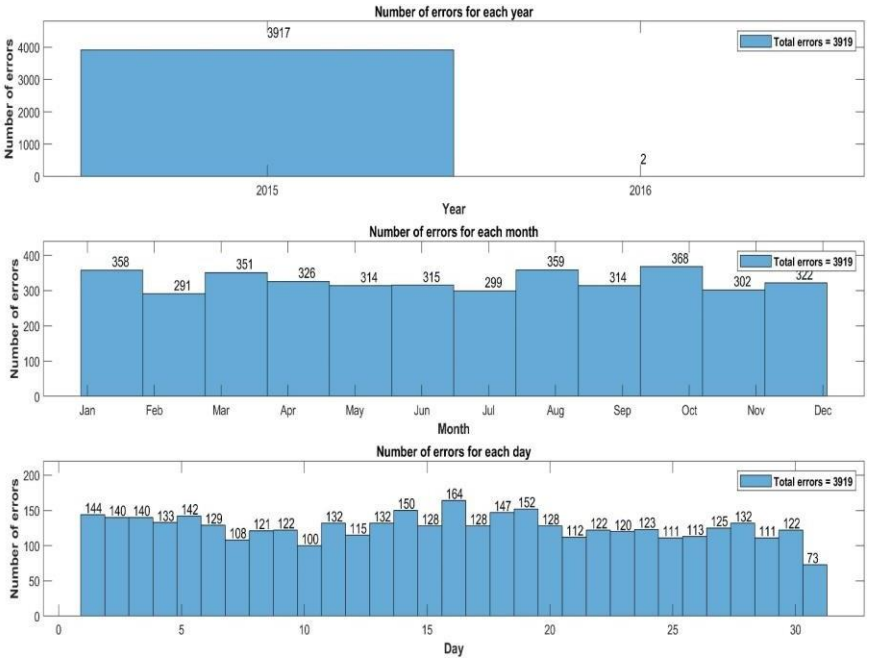

Figure 3.1. No. of errors by year, month, and day.

From fig. 3.1, it is observed that there are 3917 errors encountered in the year 2015 and only 2 errors encountered in the year 2016. These two errors observed in the machine number 8 and 30 and the error 3 and then error 2 occurred in these two machines.

Most of the errors occurred during the month October, then august and January follows this criterion. In contrast, the least number of errors occurred during the month of February followed by July and November.

On the 16th day, most of the errors occurred. On the other hand, a smaller number of errors occurred on the 31st day followed by 10th and 8th day of each month.

Now the visualization for the errors of the 100 machines are observed for each hour of a day, minute and second.

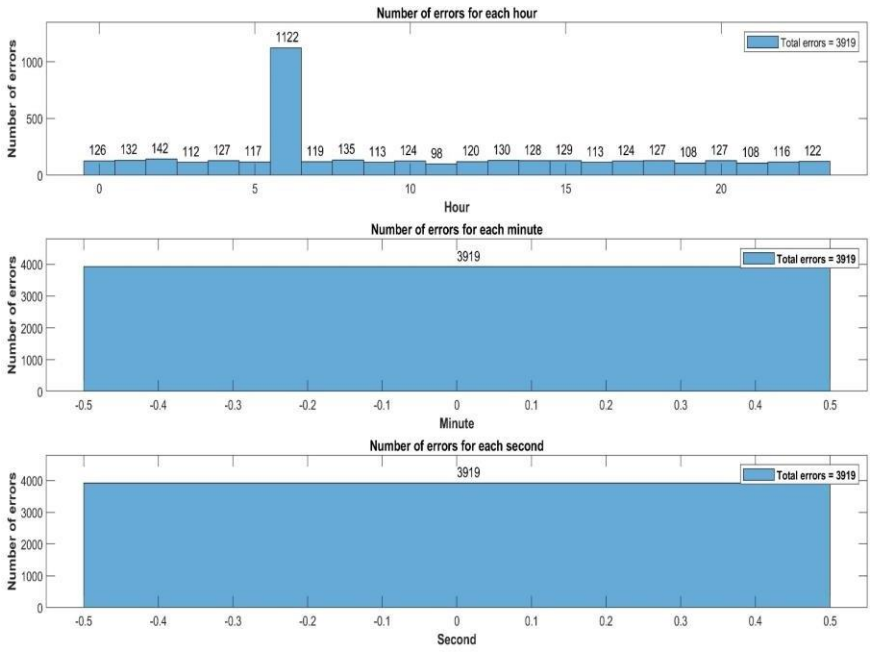

Figure 3.2. No. of errors by hour, minute and second.

From fig. 3.2. the observation is that most of the errors occurred on 6 pm. Another observing thing is that during the whole 24 hours of a day there occurred errors. From the minute and second information it can be said that all the errors occurred on the exact hour time.

Thus, for the prediction based on the machine error history, month, day, and hour is a good predictor. This is because the number of errors changes with the change of month, day, and hour. In contrast, year, minutes, and second do not have any impact on errors.

3.1.2 PdM_failures.csv

This dataset has the information of the failures of 100 machines and there are 4 failures faced by the machines. The information is provided based on the year, month, day, hour, minutes, and seconds. The information is for the year of 2015. This information is provided for each month and day. There is also hour information, but minutes and seconds remains the same. The below figure shows the plot of 100 machine failures based on the year, month, and day.

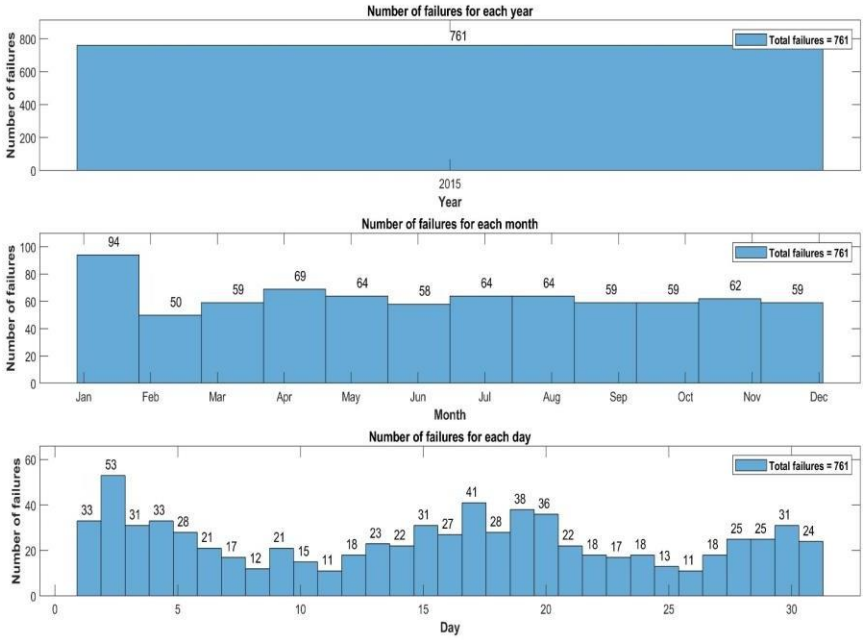

Figure 3.3. No. of failures by year, month, and day.

From fig. 3.3. all the failures occurred during the year 2015. Additionally, most of the failures occurred during the month of January and least number of failures occurred in February. Furthermore, march, September, October, and December showed the greatest number of failures. During each month, the greatest number of failures occur on the 2nd day and least number of failures occurred on 11th and 26th day of each month.

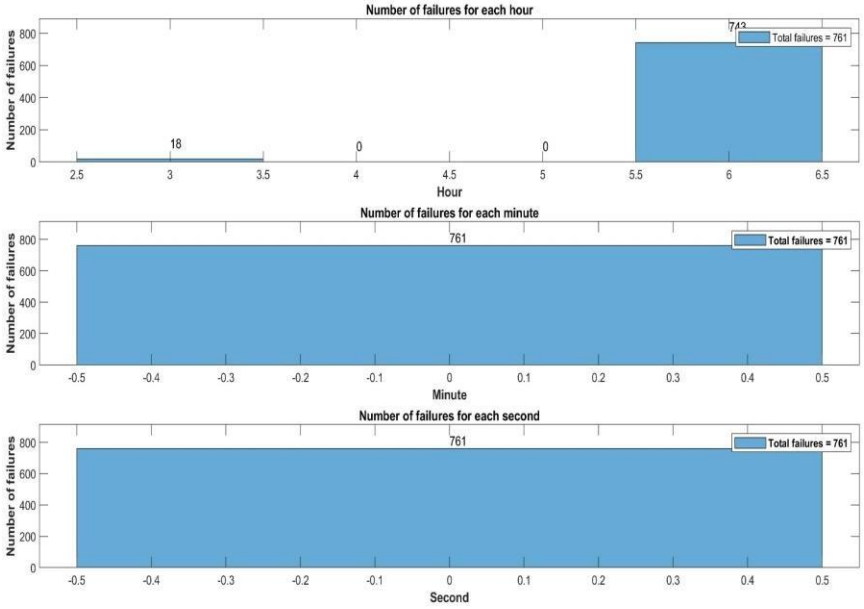

Figure 3.4. No of failures by hour, minute and second.

From the hour information of this failure dataset, most of the failures occurred at 06:00:00 pm. From the minute and second information all those failures occurred at exact 06:00:00. Additionally, there are some other failures that occurred at 03:00:00 pm at the same time, same date, same month, and same year.

Thus, for the prediction based on the machine failure history, month, day, and hour is a good predictor. This is because the number of failures changes with the change of month, day, and hour. In contrast, year, minutes, and second do not have any impact on failures.

3.1.3 PdM_telemetry.csv

This dataset consists of the information for the voltage, rotation, pressure, and vibration data for every hour of the 100 machines. This data is the sensor values taken from the starting time of 06:00:00 of 1st January 2015 and the ending time is also 06:00:00 of 1st January 2016. Now, the data distribution is checked based on density.

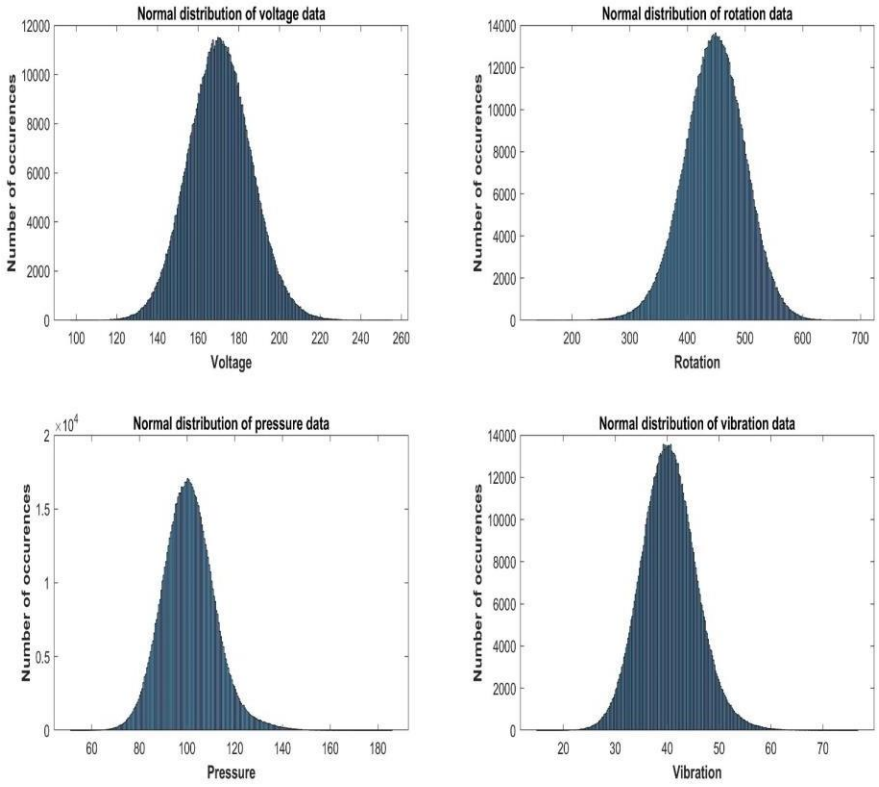

Figure 3.5. Distribution of the sensor data.

Fig. 3.5. shows that the voltage, rotation, pressure, and vibration data are normally distributed. The meaning is that most of the dataset is located near to the average values of the dataset. This is important to figure out either the data is normally distributed or not, as the gaussian process regression algorithm is used which works better on the normally distributed data.

In addition, this dataset contains 8760 rows and 6 columns. The rows are the hourly information, and the columns are the date time, machineID, voltage, rotation, pressure, and vibration data for 1 year. This information of the raw data is shown below for a randomly selected machine number 19.

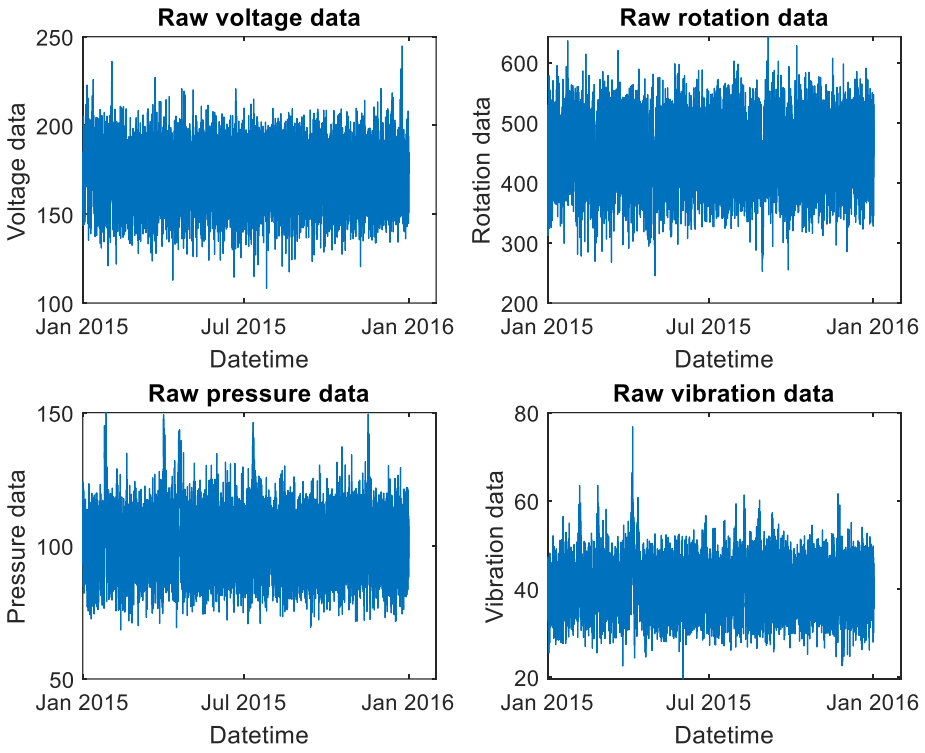

Figure 3.6. Raw dataset for machine no 19.

Fig. 3.6. reflects the raw data of machine number 19. There are data from January 2015 to January 2016. By observing carefully, it is seen that the data deviates very fast and no necessary information can be seen. Another observation is that the 4-sensor data are not on the same scale. Hence, before feeding to the machine learning algorithm, the data requires some preprocessing.

3.2 Data preprocessing

To remove the noise, the data is filtered with the moving average filter taking 12 nearest neighbors by using equation (1). Then, smoothing technique have been used to make the edges of the signal smoother. After this, the dataset looks like below:

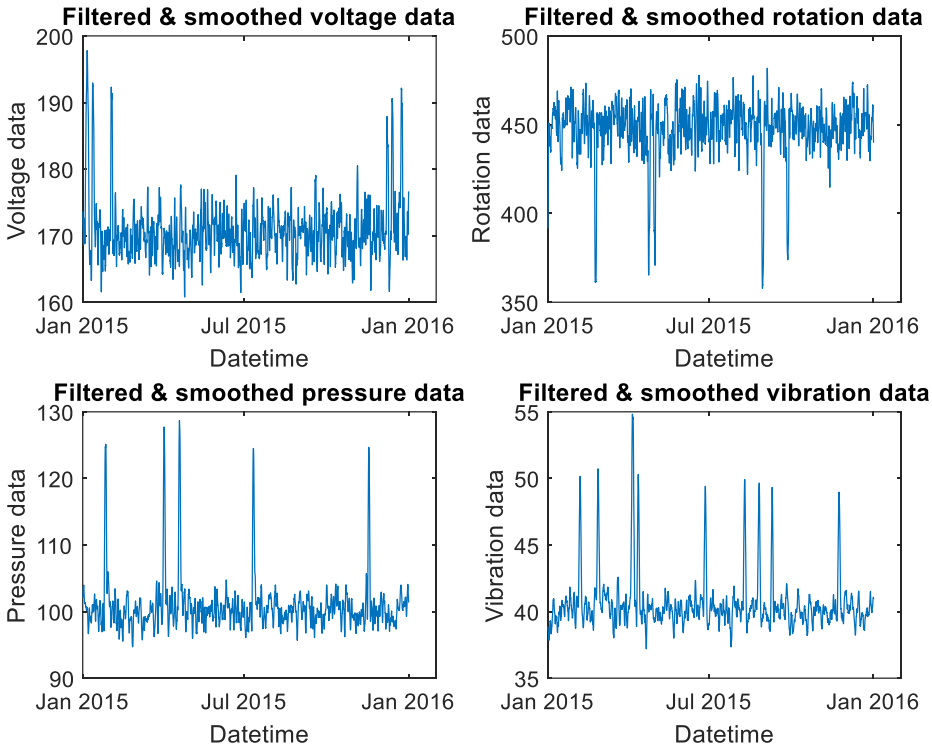

Figure 3.7: Filtered and smoothed sensor data for machine no19.

In fig. 3.7, it is seen that the filtering and smoothing technique on the sensor data, distinguished the anomalies successfully. In addition, the anomalies are the abnormal behavior of the machines which indicates the degradation of the machine condition. On the other hand, the healthy behavior of the machines lies near the mean of the sensor data.

The mean of the sensor data is given below:

Voltage Mean.	Rotation Mean.	Pressure Mean.	Vibration Mean.
170.8320	446.3353	100.6719	40.5858

Table 3.1: Mean of the sensor values.

Thus, from fig. 3.7. and table 3.1, it is seen that if the machines are in healthy operation the sensor readings lie near to the mean value. On the other hand, if the sensor readings go beyond the mean values (anomalies), then the machines are in faulty operation.

To detect the anomalies, peak accurately, the rotation data is inverted. Then, the finding peak technique is used to detect those anomalies peak. This is done by sorting all the sensor data from higher to lower values. Then the peaks of these higher values are extracted which is marked red star as shown in the figure below:

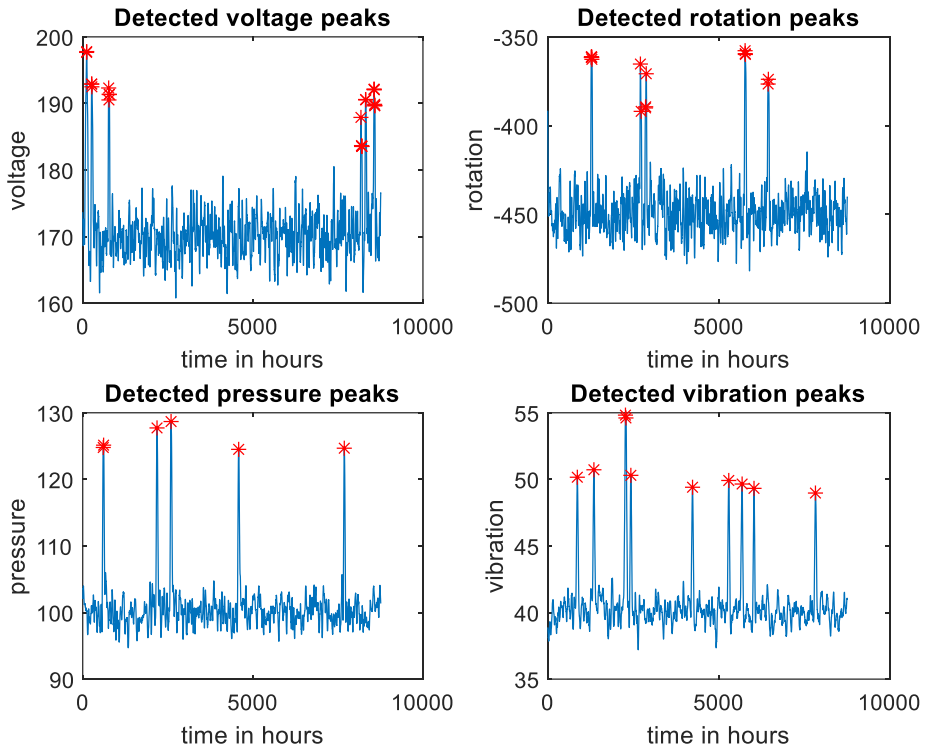

Figure 3.8: Detected voltage, rotation, pressure, and vibration anomaly peaks for machine no 19.

The red star shows the detected peaks for all the 4-sensor data. This calculation is further done for all the 100 machines and saved their time on 4 different csv file named voltage.csv, rotate.csv, pressure.csv and vibration.csv.

Finally, to predict the future data, machine learning is being used. Before inserting the data into a machine learning algorithm, the data must be transformed into a form that the algorithm can understand. So, the data is scaled, as all the sensor data are not on the same scale which can be seen from fig. 3.8. To make them on the same scale, equation (2) is being used. The mean and standard deviation is calculated. Then the mean value is subtracted from all the current sensor value and divided by the standard deviation. This scenario can be seen from the figure below:

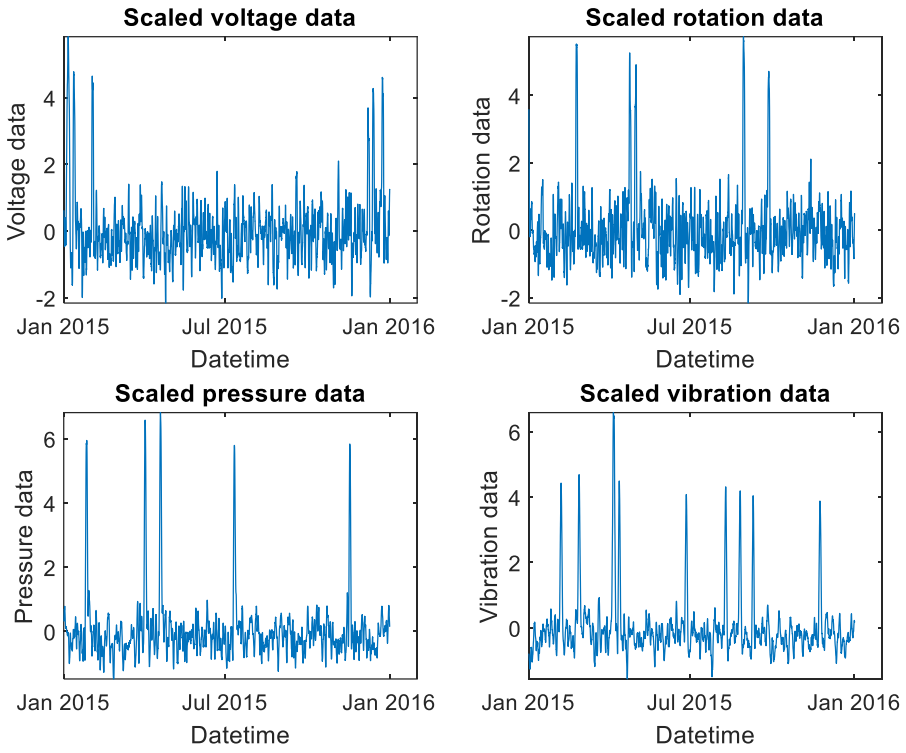

Figure 3.9: Scaled sensor data for machine number 19.

From fig. 3.9, it is seen that the data is scaled now as the data is centered around 0 mean and the standard deviation is 1. This is done to make all the sensor values in a same scale.

3.3 Probability calculation

During the investigation, it is also seen that all the anomalies on each sensor data do not lead to a machine error and failure. In addition, the number of error and failure changes for an anomaly on different sensor and different machines. Thus, the probability must be calculated to figure out which sensor anomalies have high relevancy to indicate machine failure.

Thus, it is very important to figure out the time period of an anomaly. This can be done from the detected anomaly from the sensor data. The detected peak of anomaly is then related with the error and failure history to find out the time period of an anomaly. Specifically, this can be derived from one such failure occurred for machine number 19 in 2015.02.27 06:00:00.

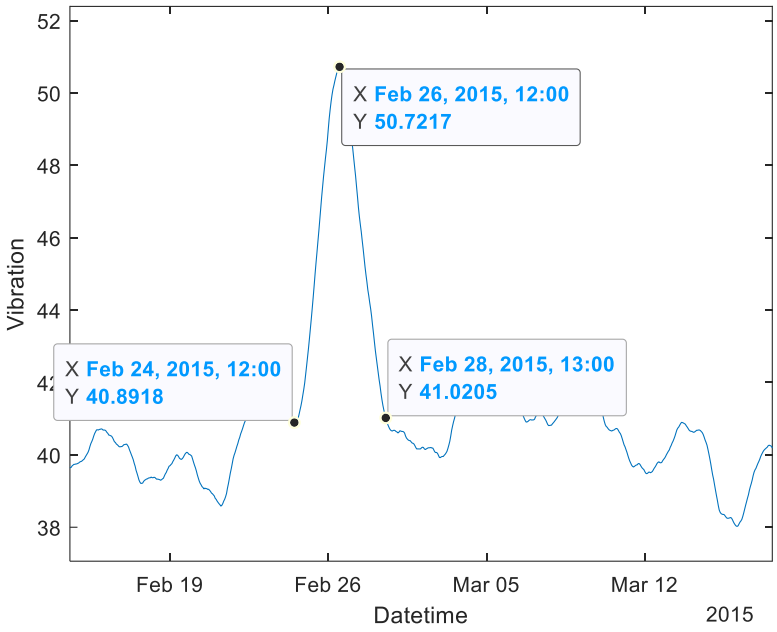

Figure 3.10: Time period of anomaly of a failure for machine number 19.

It is seen from fig. 3.10. that, the time period of an anomaly is approximately 96 hours from the start to end time of that anomaly. Machine number 19 experience all the errors and failures during this anomalies time period which is also true for all 100 machines.

To calculate the probability of failure during the time period of an anomaly for machine number 19, the datetime is sorted to remove closest and duplicate datetime. By doing this, the number of voltage, rotation, pressure, and vibration anomalies are 6, 5, 5 and 9 respectively. The number of failures occurred during the time period of these anomalies are 2,0,0 and 5 respectively. By using equation (4), the calculated probability of failure during the time period of an anomaly for voltage, rotation, pressure, and vibration anomalies are 0.3333, 0.0, 0.0, and 0.5556 respectively. In this similar way, the probability of error and failure for all the 100 machines are calculated.

After the probability is calculated it is seen that machine number 85 is the most important among the 100 machines. Thus, the probability of error and failure together can be seen from the figure below:

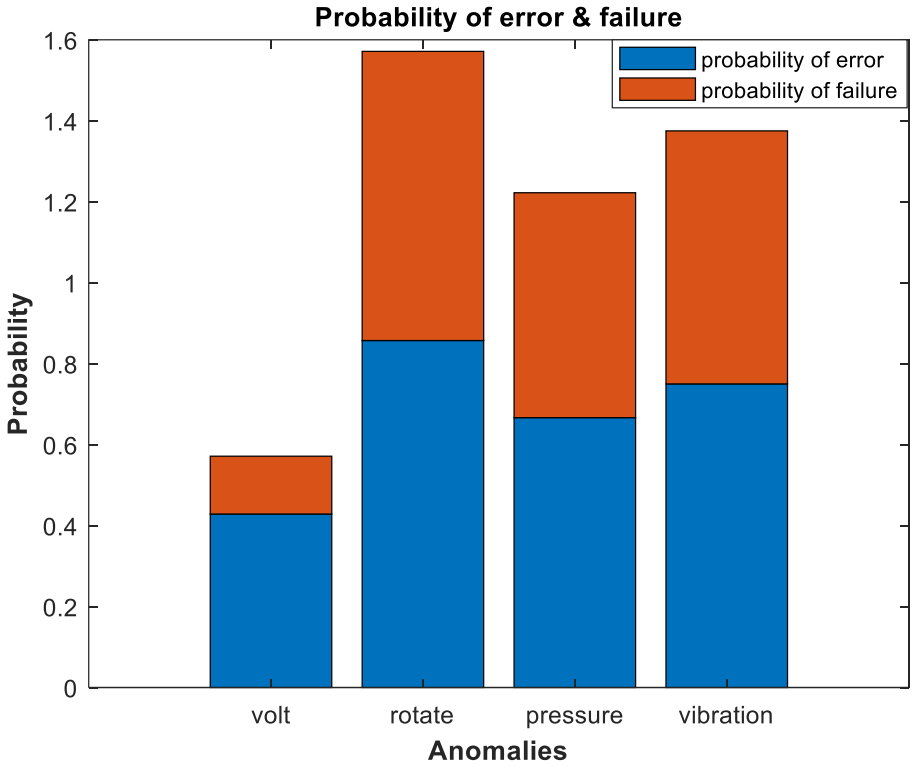

Figure 3.11: Probability of error & failure together for machine number 85.

From fig. 3.11. it is seen that the probability of error and failure together is high if there is an anomaly on the rotation data. The probability of error and failure together is approximately close to 1.60. Thus, rotation sensor data is the most relevant sensor data for machine number 85. As rotation sensor data shows most relevancy among the other sensor data for machine number 85, hence it can be said that for this machine the response is the rotation sensor data for failure prediction.

Though rotation anomalies have shown high relevancy, it is also observable that for vibration anomalies the probability of error and failure together is close to 1.40 and for pressure anomalies the probability of error and failure together is more than 1.20. For voltage anomalies both error and failure probability are also seen. But for other machines, either the probability of error and failure is less or even close to zero. For this reason, machine number 85 is the most important and also chosen for failure prediction.

3.4 Prediction of the most relevant sensor data as response

To predict the response, a machine learning model must be trained with the training data and then the trained model will be validated with the testing data. In this way, the process of building a machine learning model is shown on the figure below:

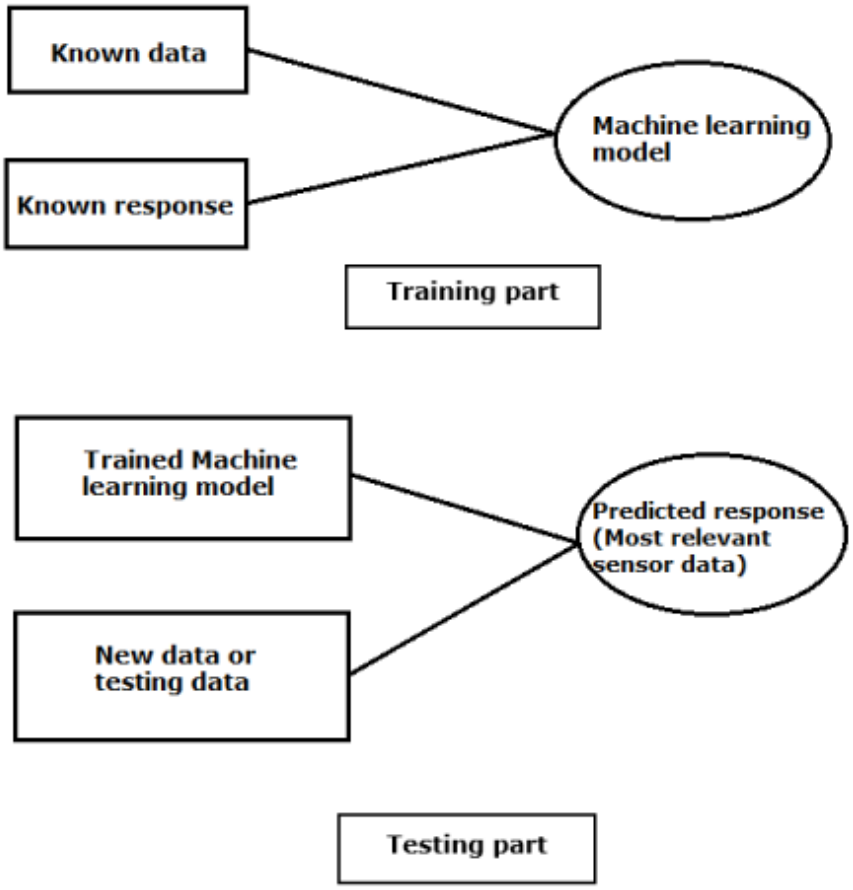

Figure 3.12: Prediction of response with machine learning model.

By using fig. 3.12. the prediction of response for machine number 85 is performed. But before that the most relevant sensor data must be extracted which is the response. The calculated error and failure probability together which is the most among 4 sensors, reflects the most relevant sensor data for a machine. As explained in section 3.3 in fig. 3.11. rotation sensor data is the most relevant sensor data for machine number 85. Thus, the prediction of this sensor data is highly related to failure prediction.

To do this, the historical error, failure, and sensor data is used. For machine number 85, one error occurred at 2015-10-11 06:00:00 and one failure occurred at 2015-10-12 06:00:00. Both error and failure occurred within an interval of 24 hours, and it happened on the month October. Thus, to assure the quality of data only the data of October is used. This will also lead to the concept that the machine learning algorithm will know about the error, failure, & healthy data and the algorithm can also distinguish both the healthy and faulty operation of a machine.

To predict the rotation data as response, the data from the month October is extracted. This data from the month October is scaled with standard scale. Then, the data is divided into 80% training and 20% testing by using cross validation randomly. The table below shows the error datetime, failure datetime, predictors, and response for the machine number 85 to train the algorithm.

Error datetime	Failure datetime	Predictors	Response
2015-10-11 06:00:00	2015-10-12 06:00:00	Month, day, hour, voltage, pressure & vibration data	Rotation

Table 3.2: Error & failure datetime, predictors and response for machine number 19

The training data of the predictors can be seen from the figure below:

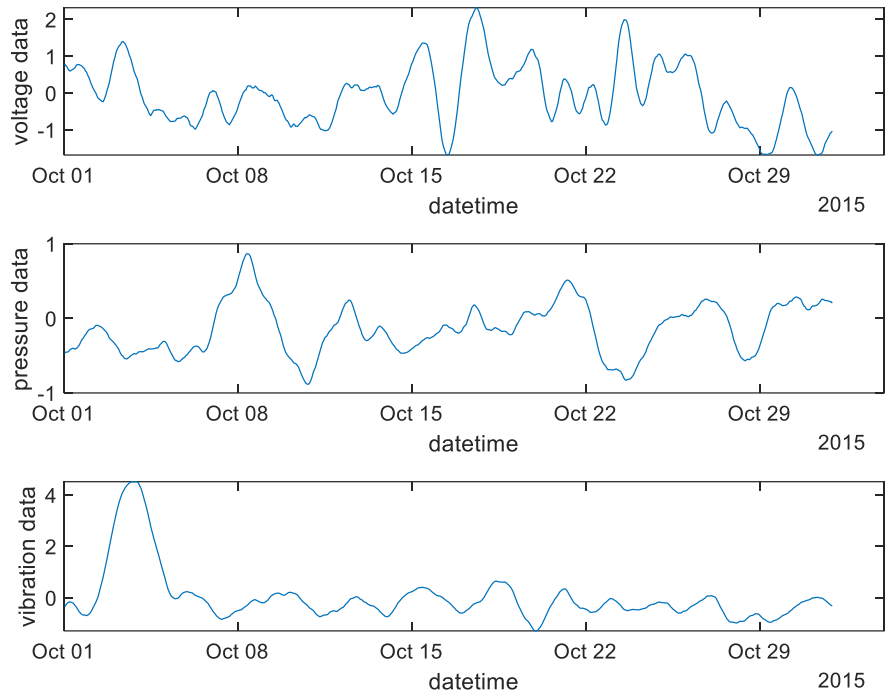

Figure 3.13: Training data of predictors for machine number 85.

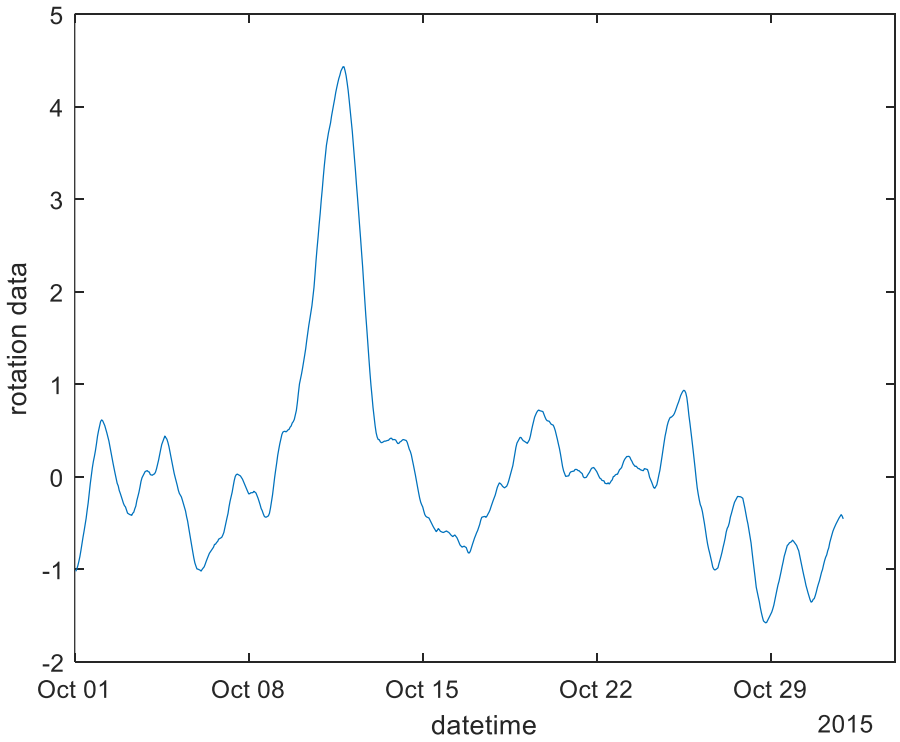

Figure 3.14: Training data of response for machine number 85.

As explained in table 3.2, the predictors data to train the algorithm, can be seen from the fig. 3.13. In addition, the response data to train the algorithm, can be seen from fig. 3.14. These predictors and response must be defined properly so that the algorithm properly understand about the difference between the healthy and faulty data.

With this known training data as in fig. 3.13, 3.14 and known response, the gaussian process regression algorithm is trained. The major parameters are the covariance function and the sigma value which is the length scale in equation (12) which is shown in detail from equation (5). The covariance function is the exponential function, and the length scale of sigma is calculated by using equation (13). This length scale is calculated by using the Euclidian scale within 2 data points. Other functions are kept fully independent during the calculation. To avoid overfitting, regularization is one of the parameters on gaussian process regression and it is set as 0.05 for the prediction.

By doing all this calculation, the most relevant sensor data is predicted as response. In addition, this predicted response is related with the calculated probability to define the high or low risk of experiencing machine failure.

Finally, by using the equation (14), (15), and (16) RMSE, MAE, and RAE is calculated between the actual and predicted data. This three calculates the error estimation. In addition, by using equation (17) the correlation coefficient is also calculated between the actual and predicted data. This usually calculates how much the actual and predicted data are related to each other. With all these 4 statistical metrics the performance of the algorithm is evaluated.

4 Results

4.1 Anomalies to error probability

During the time period of an anomaly, the probability of error is calculated as shown in the figure below:

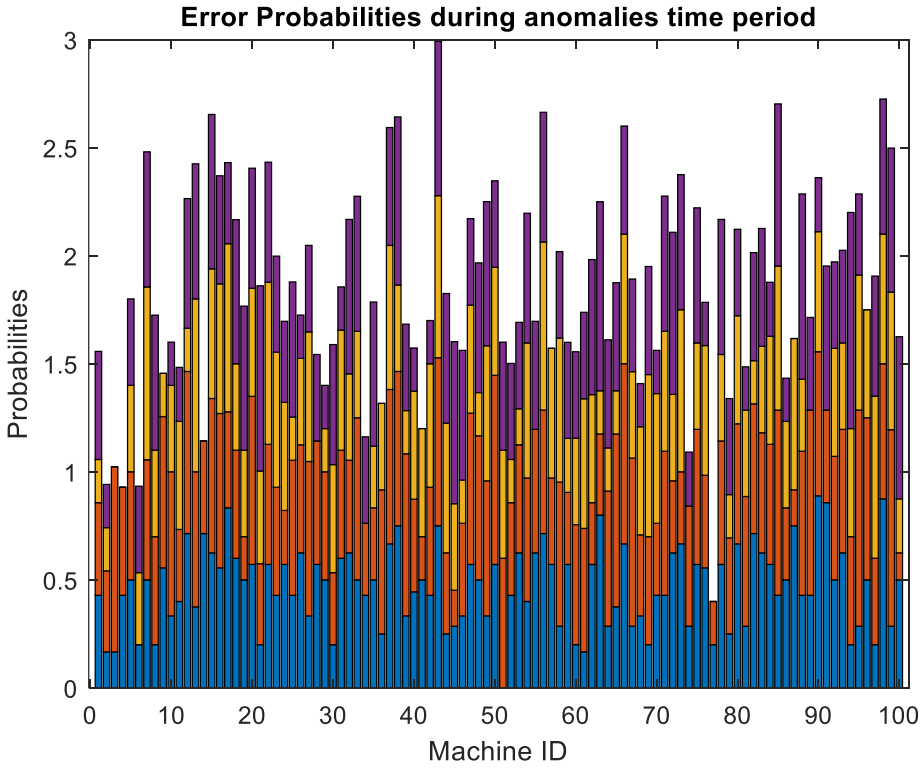

Figure 4.1: Error probabilities during the anomalies time period.

From fig. 4.1, it is seen that machine number 2 have low probability of errors, in contrast machine number 43 have high probability of errors during the time period of anomalies. There are some other machines for which both of this statement is positive.

In the similar procedure, the machines with high probability of errors are distinguished and this can be found on table 4.1.

Anomaly	Machine number	Total machines
Voltage	12,14,15,17*,18,26,31,32,37,38,43,53,55,56, 63,66,72,73,80,82,83,87,90*,91*,93,98*.	26
Rotation	3*,9,10,12,13,15,16,20,25,27,33,36,37,38,39, 43,47,48,49,50*,51,58,64,65,66*,67,71,75,81, 82,85*,88,89*,90,95*,96,98,99*.	38
Pressure	7*,13*,15,16,17,22,23,27,37,43,44,49,54,56, 57,58,61,66,69,70,73,76,85,87,95,97,98,99.	28
Vibration	7,8,12,13,15,18,19,21*,25,32,33,35,38,43,44, 45,46,48,49,54,56,62,63*,71,72,73,75,78,85, 88*,91,94*,98,99,100.	35

Table 4.1: Machines with high probability of errors during anomalies time period.

N.B: The blue star is for the machines with very high error probability and red star are for the machines that have the error probability 1.

From table 4.1, it is seen that for rotation anomalies machine number 95 and for vibration anomalies machine number 94 have the error probability of 1 which are marked in red star. For rotation anomalies, there are 38 machines, for vibration anomalies, there are 35 machines, for pressure anomalies, there are 28 anomalies, and for voltage anomalies, there are 26 machines having high to very high probability of errors. Thus, rotation anomalies have high relevancy for having more machines with high probability of errors during the time period of an anomaly.

4.2 Anomalies to failure probability

The errors do not let the machine shut rather they lead to the machine failures. For this reason, the probability of machine failure is calculated during the time period of anomaly. The probability of machine failure is shown in the figure below:

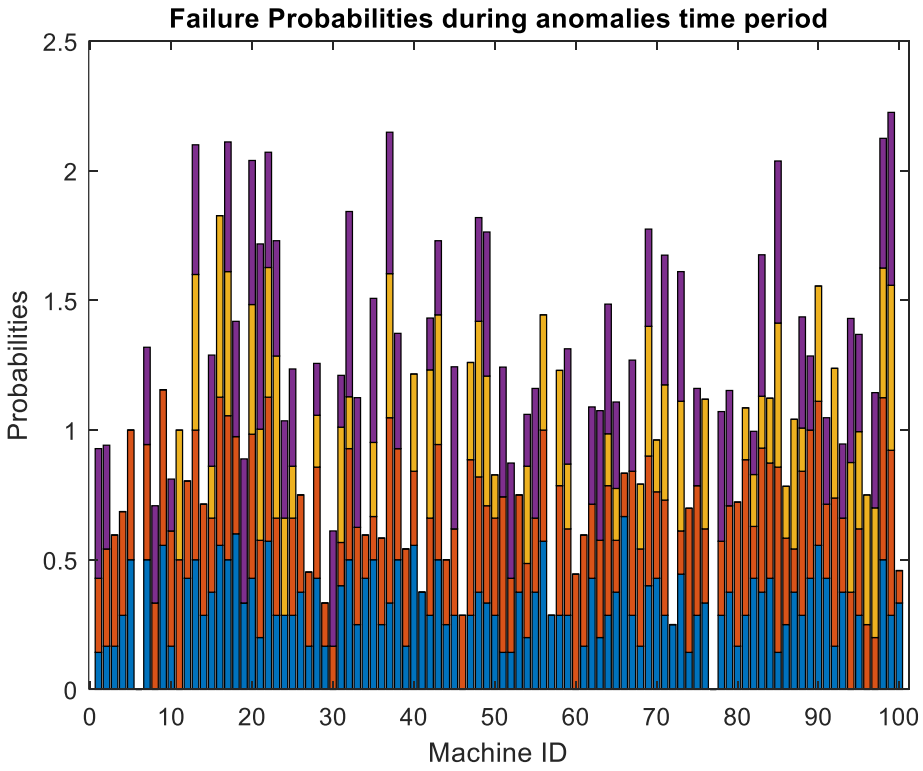

Figure 4.2: Failure probabilities during the anomalies time period.

From fig. 4.2, machine number 6 and 77 still have zero probability of failures. The probability of machine failure is high for machine number 99 while the reverse is true for machine number 100 for any of the 4 anomalies. There are some other machines that satisfies this statement.

In this way, the machines with high probability of failures are distinguished and these machines can be found on the table 4.2.

Anomaly	**Machine number**	**Total machines**
Voltage	18,66.	2
Rotation	9,37,47,51,81,85,98,99.	8
Pressure	13,16,23,48,99.	5
Vibration	21,32,45,85,99.	5

Table 4.2: Machines with high probability of failure during the anomalies time period.

From table 4.2. there are 8 machines for rotation anomalies, 2 machines for voltage anomalies and 5 machines each for pressure and vibration anomalies having high probability of failure during the time period of an anomaly. Thus, rotation anomalies have high relevancy of machine failures during this time period.

In addition, machine number 99 have experienced high probability of failures for multiple anomalies. Thus, there is high probability that this machine will experience multiple failures. For machine number 85 this is also positive.

4.3 Prediction of rotation data as response

For machine number 85, rotation data is predicted as response. The predicted rotation response for machine number 85 can be seen on fig. 4.3.

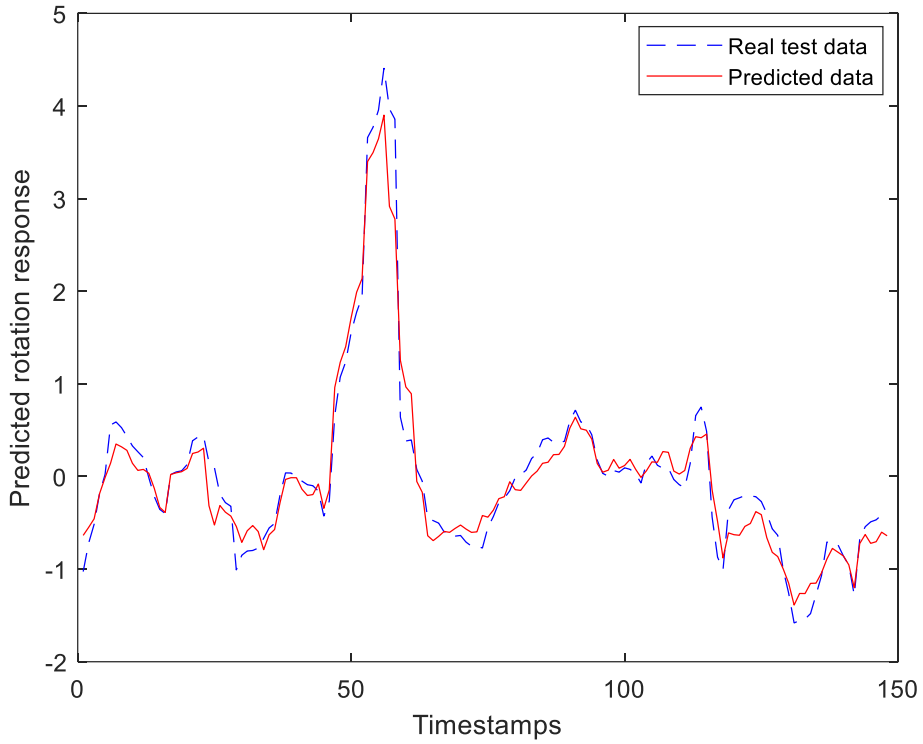

Figure 4.3: Predicted rotation data for machine number 85.

From fig. 4.3, it is seen that, the timestamps are in hour. Then, on the test data there is an anomaly, and the sum of the calculated probability of error and failure is approximately close to 1.60. Thus, an anomaly on the rotation data for this machine means that the machine is in high risk of experiencing error and failure. For this reason, the machine requires maintenance.

To validate the performance of this prediction, 4 statistical metrics values are given below:

RMSE	MAE	RAE	R^2
0.0571	0.1727	0.0550	0.9435

Table 4.3: RMSE, MAE, RAE, and R^2 values between the actual and predicted data.

In table 4.3. RMSE, MAE, RAE values are the error estimation between the actual and predicted data. In contrast, the R^2 values indicate how much the actual and predicted data are related to each other.

5 Discussion

The maintenance of the machines can be performed by predicting the machine failure. The prediction of this failure is performed mainly by three parts.

On the 1st part, the anomalies peak must be detected properly from the sensor data. To distinguish the anomalies from the noisy sensor data, filtering was performed by using the 12 nearest moving average filter. Several filters were tested but this filter was chosen as it is easy to implement than the other tested filter. In addition, several techniques had been tested for the detection of anomalies peak. But the sorting of the sensor values from higher to lower provided the better result to detect the anomalies peak. For further preprocessing of the sensor data, the standard scale was performed to carry out all the sensor data into a same scale before training the machine learning algorithm.

On the 2nd part, the probability of machine error and failure was calculated by using the basic probability equation. The sum of this calculated probability of error and failure was used to distinguish the most relevant sensor data of a machine.

On the 3rd part, gaussian process regression algorithm was used to predict the most relevant sensor data as response. For this prediction, several algorithms had been tested. But this algorithm was chosen as the data is normally distributed. Another reason was the minimum RMSE, MAE and RAE and maximum R^2 values between the actual test data and predicted data. During this, the RMSE, MAE, RAE gave the error value of less than 0.20 and the R^2 value was 0.9435 individually, which is acceptable.

In addition, the monthly observation of a machine was performed for assuring the data quality by training the algorithm with both one error and one failure data. In addition, for a month there are several machines that have no failure rather for another month there are multiple failures.

The united nation had planned a 15-year plan. This plan aims at eradicating poverty, save the world, and develop the life of human being. This goal is not much achieved within the year 2020, but it is expected that within 2030, the most of these goals will be achieved [21].

There are 17 goals among which ranked 9 have the goal about industry, infrastructure, and innovation. Among the goal 9 targets, 9.4 says about the efficiency of resources should be increased. Thus, the prediction of machine errors and failures will reduce the sudden downtime and unusual maintenance cost. Furthermore, any small problem can be fixed before it becomes big. This will increase any machine health by replacing only the defected components of any machine and it will be done only when any of the component experiences error and failure. Thus, the efficiency of a machine can be increased which can help sustainable development [21].

6 Conclusion

In this thesis work, the anomaly peak detection was performed to calculate the probability of error and failure. The sum of this calculated probability defined the most relevant sensor data. The use of gaussian process regression algorithm successfully predicted this most relevant sensor data and coherent with the calculated probability to define the condition of a machine. If the sum of the calculated probability of error and failure is high and there is an anomaly on the predicted most relevant sensor data, then the machine requires maintenance. But if the sum of the calculated probability of error and failure is low and there is an anomaly on the most relevant sensor data, then the machine is in a good condition. From the calculation, it can be said that rotation data have the most relevancy for most of the machine's failure prediction.

The monthly observation of the machines can reduce the time consumed and ensures the data quality. During this observation, the prediction and detection of anomaly peak are very important for pdm. Finally, this prediction of failure can be done in real industries upon the policy and preference of a company.

In future work, the remaining useful life (RUL) can be estimated by taking the error and failure history into account of all the 100 machines. In addition, more data can be used by using multiple errors and failures, for the similar prediction instead of using monthly observation.

References

[1] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone and A. Beghi, "Machine Learning for Predictive Maintenance: A Multiple Classifier Approach," in IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 812-820, June 2015, doi: 10.1109/TII.2014.2349359.

[2] Gocodes.com. [Online]. Available: https://gocodes.com/equipment-maintenance-statistics/.

[3] *Divaportal.org*.[Online].Available:http://www.divaportal.org/smash/get/diva2:1377581/FULLTEXT01.pdf.

[4] S. Çoban, M. O. Gökalp, E. Gökalp, P. E. Eren and A. Koçyiğit, "[WiP] Predictive Maintenance in Healthcare Services with Big Data Technologies," 2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA), 2018, pp. 93-98, doi: 10.1109/SOCA.2018.00021.

[5] S. Y. Selçuk, P. Ünal, Ö. Albayrak, and M. Jomâa, "A workflow for synthetic data generation and predictive maintenance for vibration data," *Information (Basel)*, vol. 12, no. 10, p. 386, 2021.

[6] M. L. Araiza, "A formal framework for predictive maintenance," Proceedings AUTOTESTCON 2004., 2004, pp. 489-495, doi: 10.1109/AUTEST.2004.1436938.

[7] V. Pandey and V. K. Giri, "High frequency noise removal from ECG using moving average filters," 2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES), 2016, pp. 191-195, doi: 10.1109/ICETEESES.2016.7581383.

[8] Mathworks.com.[Online].Available:https://www.mathworks.com/content/dam/mathworks/ebook/gated/predictive-maintenance-ebook-all-chapters.pdf.

[9] H. Magsi, A. H. Sodhro, F. A. Chachar and S. A. K. Abro, "Analysis of signal noise reduction by using filters," 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018, pp. 1-6, doi: 10.1109/ICOMET.2018.8346412.

[10] D. T. Thinh, N. B. H. Quan and N. Maneetien, "Implementation of Moving Average Filter on STM32F4 for Vibration Sensor Application," 2018 4th International Conference on Green Technology and Sustainable Development (GTSD), 2018, pp. 627-631, doi: 10.1109/GTSD.2018.8595630.

[11] D. Chanal, N. Y. Steiner, D. Chamagne and M. -C. Pera, "Impact of standardization applied to the diagnosis of LT-PEMFC by Fuzzy C-Means clustering," 2021 IEEE Vehicle Power and Propulsion Conference (VPPC), 2021, pp. 1-6, doi: 10.1109/VPPC53923.2021.9699234.

[12] C. Zhang, J. Yella, Y. Huang and S. Bom, "Learning from Failures in Large Scale Soft Sensing," 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 6067-6070, doi: 10.1109/BigData52589.2021.9671366.

[13] *Personal.utdallas.edu*, 2022. [Online]. Available: https://personal.utdallas.edu/~scniu/OPRE-6301/documents/Probability.pdf.

[14] J. Hua, "Study on the Application of Rough Sets Theory in Machine Learning," 2008 Second International Symposium on Intelligent Information Technology Application, 2008, pp. 192-196, doi: 10.1109/IITA.2008.154.

[15] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu and M. Stanley, "A brief survey of machine learning methods and their sensor and IoT applications," 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA), 2017, pp. 1-8, doi: 10.1109/IISA.2017.8316459.

[16] X. Wang, J. Zhou and Peng Guo, "Wind turbine gearbox forecast using Gaussian process model," The 26th Chinese Control and Decision Conference (2014 CCDC), 2014, pp. 2621-2625, doi: 10.1109/CCDC.2014.6852616.

[17] "Kernel (covariance) function options - MATLAB & Simulink - MathWorks Nordic," Mathworks.com. [Online]. Available: https://se.mathworks.com/help/stats/kernel-covariance-function-options.html.

[18] M. Assim, Q. Obeidat and M. Hammad, "Software Defects Prediction using Machine Learning Algorithms," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-6, doi: 10.1109/ICDABI51230.2020.9325677.

[19] D. Purwanto, C. Eswaran and R. Logeswaran, "A Comparison of ARIMA, Neural Network and Linear Regression Models for the Prediction of Infant Mortality Rate," 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, 2010, pp. 34-39, doi: 10.1109/AMS.2010.20.

[20] S. Hiregoudar, "Ways to evaluate regression models," *Towards Data Science*, 04-Aug-2020. [Online]. Available: https://towardsdatascience.com/ways-to-evaluate-regression-models-77a3ff45ba70.

[21] "Infrastructure and Industrialization - United Nations Sustainable Development," United Nations.[Online].Available:https://www.un.org/sustainabledevelopment/infrastructureindustrialization/.

Appendix A

MATLAB code for the background of the dataset.

```
%#################################################################%
%#################################################################%
%%%%%%%%%% Master Thesis in Electronics/Automation %%%%%%%%%%%%%%
%-----------------------------------------------------------------%
%-------------- Background of the maintenance, error ------------%
%--------------     failure and machines dataset     ------------%
%-----------------------------------------------------------------&
%--------------         University of Gavle          ------------%
%-----------------------------------------------------------------%
%------- Topic: Analysis of Machine Condition to predict Failure %
%-----------------------------------------------------------------%
%--------------    Masters in Electronics/Automation ------------%
%-----------------------------------------------------------------%
%--------------         Md Abdur Rahman Akash        ------------%
%-----------------------------------------------------------------%
%---------------            January 2022             ------------%
%-----------------------------------------------------------------%
%#################################################################%
%#################################################################%

%%
% Read the error data
data = readtable('PdM_errors.csv');
% Slice the table to extract only the datetime array
tt = data.datetime;
% Convert the datetime to year, month and day
[y,mo,d] = ymd(tt);
% Convert the datetime to hour, minute and second
[h,mi,s] = hms(tt);
% Insert the year, month, day, hour, minute, second value to the dataset
data.year = y;
data.month = mo;
data.day = d;
data.hour = h;
data.minute = mi;
data.second = s;
% Convert the categorical data to numeric
data.errorID = categorical(data.errorID);
et = double(data.errorID);
data.errorID = et;
figure(1)
subplot(3,1,1)
% Plotting the histogram for years
h_y = histogram(y);
% Set the xlabel
set(gca,'xtick', 2014:2016,...
'xticklabel',{'2014','2015','2016'})
% Calculate the number of bin and edges for histogram plot
```

```
Edg_y = h_y.BinEdges;
count_y = h_y.BinCounts;
center_y = Edg_y(1:end-1)+diff(Edg_y)/2;
% Inserting the number for each year to the histogram plot
text(center_y, count_y+550, string(count_y))
% Labelling the axis
xlabel('Year','fontweight','bold')
ylabel('Number of errors','fontweight','bold')
title('Number of errors for each year','fontweight','bold')
legend('Total errors = 3919','fontweight','bold')
% Putting the limit for yaxis
ylim(ylim.*[1 1.2])

subplot(3,1,2)

% Plotting the histogram for months
h_h = histogram(mo,'NumBins',12);
% Set the xlabel
set(gca,'xtick',1:12,...
'xticklabel',{'Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'})
% Calculate the number of bin and edges for histogram plot
Edg_y = h_h.BinEdges;
count_y = h_h.BinCounts;
center_y = Edg_y(1:end-1)+diff(Edg_y)/2;
% Inserting the number for each month to the histogram plot
text(center_y, count_y+25, string(count_y))
% Labelling the axis
xlabel('Month','fontweight','bold')
ylabel('Number of errors','fontweight','bold')
title('Number of errors for each month','fontweight','bold')
legend('Total errors = 3919','fontweight','bold')
ylim(ylim.*[1 1.1])
subplot(3,1,3)
h_d = histogram(d,'NumBins',31);
% Calculate the number of bin and edges for histogram plot
Edg_d = h_d.BinEdges;
count_d = h_d.BinCounts;
center_d = Edg_d(1:end-1)+diff(Edg_d)/6;
% Inserting the number for each year to the histogram plot
text(center_d, count_d+10, string(count_d))
xlabel('Day','fontweight','bold')
ylabel('Number of errors','fontweight','bold')
title('Number of errors for each day','fontweight','bold')
legend('Total errors = 3919','fontweight','bold')
ylim(ylim.*[1 1.1])
figure(2)
subplot(3,1,1)
```

```
h_h = histogram(h);
Edg_h = h_h.BinEdges;
count_h = h_h.BinCounts;
center_h = Edg_h(1:end-1)+diff(Edg_h)/6;
text(center_h, count_h+100, string(count_h))
xlabel('Hour','fontweight','bold')
ylabel('Number of errors','fontweight','bold')
title('Number of errors for each hour','fontweight','bold')
legend('Total errors = 3919')
ylim(ylim.*[1 1.2])
subplot(3,1,2)
h_mi = histogram(mi);
Edg_mi = h_mi.BinEdges;
count_mi = h_mi.BinCounts;
center_mi = Edg_mi(1:end-1)+diff(Edg_mi)/2;
text(center_mi, count_mi+350, string(count_mi))
xlabel('Minute','fontweight','bold')
ylabel('Number of errors','fontweight','bold')
title('Number of errors for each minute','fontweight','bold')
legend('Total errors = 3919','fontweight','bold')
ylim(ylim.*[1 1.2])
subplot(3,1,3)
h_s = histogram(s);
Edg_s = h_s.BinEdges;
count_s = h_s.BinCounts;
center_s = Edg_s(1:end-1)+diff(Edg_s)/2;
text(center_s, count_s+350, string(count_s))
xlabel('Second','fontweight','bold')
ylabel('Number of errors','fontweight','bold')

title('Number of errors for each second','fontweight','bold')
legend('Total errors = 3919','fontweight','bold')
ylim(ylim.*[1 1.2])
figure(3)
h_comp = histogram(data.errorID);
set(gca,'xtick',1:5,...
'xticklabel',{'error1','error2','error3','error4','error5'})
Edg_comp = h_comp.BinEdges;
count_comp = h_comp.BinCounts;
center_comp = Edg_comp(1:end-1)+diff(Edg_comp)/2;
text(center_comp, count_comp+20, string(count_comp))
xlabel('Errors','fontweight','bold')
ylabel('Number of errors','fontweight','bold')
title('Number of errors of the components','fontweight','bold')
legend('Total errors = 3919')
ylim(ylim.*[1 1.1])
%%
```

```
% Read the failure data
data = readtable('PdM_failures.csv');
% Slice the table to extract only the datetime array
tt = data.datetime;
% Convert the datetime to year, month and day
[y,mo,d] = ymd(tt);
% Convert the datetime to hour, minute and second
[h,mi,s] = hms(tt);
% Insert the year, month, day, hour, minute, second value to the dataset
data.year = y;
data.month = mo;
data.day = d;
data.hour = h;
data.minute = mi;
data.second = s;
% Convert the categorical data to numeric
data.failure = categorical(data.failure);
ft = double(data.failure);
data.failure = ft;
figure(1)
subplot(3,1,1)
% Plotting the histogram for years
h_y = histogram(y);
% Set the xlabel
set(gca,'xtick', 2014:2016,…
'xticklabel',{'2014','2015','2016'})
% Calculate the number of bin and edges for histogram plot
Edg_y = h_y.BinEdges;
count_y = h_y.BinCounts;
center_y = Edg_y(1:end-1)+diff(Edg_y)/2;
% Inserting the number for each year to the histogram plot
text(center_y, count_y+60, string(count_y))
% Labelling the axis
xlabel('Year','fontweight','bold')
ylabel('Number of failures','fontweight','bold')
title('Number of failures for each year','fontweight','bold')
legend('Total failures = 761','fontweight','bold')
% Putting the limit for yaxis
ylim(ylim.*[1 1.2])
subplot(3,1,2)
% Plotting the histogram for months
h_h = histogram(mo,'NumBins',12);
% Set the xlabel
set(gca,'xtick',1:12,…

'xticklabel',{'Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec'})
% Calculate the number of bin and edges for histogram plot
```

```
Edg_y = h_h.BinEdges;
count_y = h_h.BinCounts;
center_y = Edg_y(1:end-1)+diff(Edg_y)/2;
% Inserting the number for each month to the histogram plot
text(center_y, count_y+10, string(count_y))
% Labelling the axis
xlabel('Month','fontweight','bold')
ylabel('Number of failures','fontweight','bold')
title('Number of failures for each month','fontweight','bold')
legend('Total failures = 761','fontweight','bold')
ylim(ylim.*[1 1.1])
subplot(3,1,3)
h_d = histogram(d,'NumBins',31);
% Calculate the number of bin and edges for histogram plot
Edg_d = h_d.BinEdges;
count_d = h_d.BinCounts;
center_d = Edg_d(1:end-1)+diff(Edg_d)/6;
% Inserting the number for each year to the histogram plot
text(center_d, count_d+5, string(count_d))
xlabel('Day','fontweight','bold')
ylabel('Number of failures','fontweight','bold')
title('Number of failures for each day','fontweight','bold')
legend('Total failures = 761','fontweight','bold')
ylim(ylim.*[1 1.1])
figure(2)
subplot(3,1,1)
h_h = histogram(h);
Edg_h = h_h.BinEdges;
count_h = h_h.BinCounts;
center_h = Edg_h(1:end-1)+diff(Edg_h)/2;
text(center_h, count_h+100, string(count_h))
xlabel('Hour','fontweight','bold')
ylabel('Number of failures','fontweight','bold')
title('Number of failures for each hour','fontweight','bold')
legend('Total failures = 761')
ylim(ylim.*[1 1.2])
subplot(3,1,2)
h_mi = histogram(mi);
Edg_mi = h_mi.BinEdges;
count_mi = h_mi.BinCounts;
center_mi = Edg_mi(1:end-1)+diff(Edg_mi)/2;
text(center_mi, count_mi+50, string(count_mi))
xlabel('Minute','fontweight','bold')
ylabel('Number of failures','fontweight','bold')
title('Number of failures for each minute','fontweight','bold')
legend('Total failures = 761','fontweight','bold')
```

```
ylim(ylim.*[1 1.2])
subplot(3,1,3)
h_s = histogram(s);
Edg_s = h_s.BinEdges;
count_s = h_s.BinCounts;
center_s = Edg_s(1:end-1)+diff(Edg_s)/2;
text(center_s, count_s+50, string(count_s))
xlabel('Second','fontweight','bold')
ylabel('Number of failures','fontweight','bold')
title('Number of failures for each second','fontweight','bold')
legend('Total failures = 761','fontweight','bold')
ylim(ylim.*[1 1.2])
figure(3)

h_fail = histogram(data.failure);
set(gca,'xtick',1:5,...
'xticklabel',{'comp1','comp2','comp3','comp4'})
Edg_fail = h_fail.BinEdges;
count_fail = h_fail.BinCounts;
center_fail = Edg_fail(1:end-1)+diff(Edg_fail)/2;
text(center_fail, count_fail+6, string(count_fail))
xlabel('Failures','fontweight','bold')
ylabel('Number of failures','fontweight','bold')
title('Number of failures of the components','fontweight','bold')
legend('Total failures = 761')
ylim(ylim.*[1 1.1])

%%
% Load the sensor data
data = readtable('PdM_telemetry.csv');
% Plot the histogram of voltage data
subplot(2,2,1) A8
```

```
histogram(data.volt)
xlabel('Voltage','fontweight','bold')
ylabel('Number of occurences','fontweight','bold')
title('Normal distribution of voltage data','fontweight','bold')
% Plot the histogram of rotation data
subplot(2,2,2)
histogram(data.rotate)
xlabel('Rotation','fontweight','bold')
ylabel('Number of occurences','fontweight','bold')
title('Normal distribution of rotation data','fontweight','bold')
% Plot the histogram of pressure data
subplot(2,2,3)
histogram(data.pressure)
xlabel('Pressure','fontweight','bold')
ylabel('Number of occurences','fontweight','bold')
title('Normal distribution of pressure data','fontweight','bold')
% Plot the histogram of vibration data
subplot(2,2,4)
histogram(data.vibration)
xlabel('Vibration','fontweight','bold')
ylabel('Number of occurences','fontweight','bold')
title('Normal distribution of vibration data','fontweight','bold') [1]
```

References

[1] File exchange - MATLAB central.
[Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/.

Appendix B

MATLAB code for probability of error calculation during the anomalies time period.

```
%#####################################################################%
%#####################################################################%
%%%%%%    Master Thesis in Electronics/Automation   %%%%%%%%%%%%%%%%%%
%---------------------------------------------------------------------%
%--------------    Calculate the anomalies to error   ----------%
%--------------    probability for all 100 machines   ----------%
%---------------------------------------------------------------------&
%--------------      University of Gavle             ------------%
%---------------------------------------------------------------------%
%------- Topic: Analysis of Machine Condition to predict Failure %
%---------------------------------------------------------------------%
%-------------- Masters in Electronics/Automation    ------------%
%---------------------------------------------------------------------%
%--------------      Md Abdur Rahman Akash           ------------%
%---------------------------------------------------------------------%
%---------------            January 2022             ------------%
%---------------------------------------------------------------------%
%#####################################################################%
%#####################################################################%
%%
% Load the failure, error and anomalies data
fail = readtable('PdM_failures.csv'); % Load failure data
error = readtable('PdM_errors.csv'); % Load error data
volt = readtable('voltage.csv'); % Load voltage anomalies data
rot = readtable('rotate.csv'); % Load rotation anomalies data
pres = readtable('pressure.csv'); % Load pressure anomalies data
vib = readtable('vibration.csv'); % Load vibration anomalies data
machine = readtable('PdM_machines.csv');
% Calculate the probality of error after
% 12hours,24 hours and 48 hours for each machine
% Calculate the probability for all the machines through iteration
for iii = 1:100 % Iterate 100 times as machines are 100
% Load the machine information with iteration
mc1 = error(error.machineID ==iii,:); % Iterate error data 100 times
mc2 = fail(fail.machineID ==iii,:); % Iterate failure data 100 times
mc3 = volt(volt.machineid ==iii,:); % Iterate voltage
% anomalies data 100 times
mc4 = rot(rot.machineid ==iii,:); % Iterate rotate anomalies
% data 100 times
mc5 = pres(pres.machineid ==iii,:); % Iterate pressure anomalies
% data 100 times
mc6 = vib(vib.machineid ==iii,:); % Iterate vibration anomalies
mc7 = machine(machine.machineID == iii,:);
% data 100 times
% Extract the error, failure and anomalies datetime
e1 = mc1.datetime; % Extract error datetime
```

```
e2 = mc2.datetime; % Extract failure datetime
e3 = unique(mc3.datetime); % Extract vibration anomalies datetime
e4 = unique(mc4.datetime); % Extract rotation anomalies datetime
e5 = unique(mc5.datetime); % Extract pressure anomalies datetime
e6 = unique(mc6.datetime); % Extract vibration anomalies datetime
n3 = e3([true; diff(e3) >= hours(48)]); % Extract vibration anomalies datetime
n4 = e4([true; diff(e4) >= hours(48)]); % Extract rotation anomalies datetime
n5 = e5([true; diff(e5) >= hours(48)]); % Extract pressure anomalies datetime
n6 = e6([true; diff(e6) >= hours(48)]);% Extract vibration anomalies datetime
% Caulculate the number of voltage anomalies becomes
% any error before and after 48 hours
volt_one_48_48 = 0; % Define to count number of logical 1
volt_zero_48_48 = 0; % Define to count number of logical 0
for pppt = 1:numel(n3) % Iterate through all the rows
   % of voltage anomalies
st1_48_48 = n3(pppt,1) - hours(48); % Define the start time
et1_48_48 = n3(pppt,1) + hours(48); % Define the end time
dh1_48_48 = isbetween(e1,st1_48_48,et1_48_48); % Find the match between
% start and end time
if dh1_48_48 == 0
   volt_zero_48_48 = volt_zero_48_48+1; % Count number of 0
else dh1_48_48 == 1
   volt_one_48_48 = volt_one_48_48+1; % Count number of 1
end
end
% Caulculate the number of rotation anomalies becomes any error
% for before and after 48 hours
rot_one_48_48 = 0;  % Define to count number of logical 1
rot_zero_48_48 = 0; % Define to count number of logical 0
for qqqt = 1:numel(n4) % Iterate through all the rows
   % of rotation anomalies
st2_48_48 = n4(qqqt,1) - hours(48); % Define the start time
et2_48_48 = n4(qqqt,1) + hours(48); % Define the end time
dh2_48_48 = isbetween(e1,st2_48_48,et2_48_48); % Find the match between
% start and end time
if dh2_48_48 == 0
   rot_zero_48_48 = rot_zero_48_48+1; % Count number of 0
else dh2_48_48 == 1
   rot_one_48_48 = rot_one_48_48+1; % Count number of 1
end
end
% Caulculate the number of pressure anomalies becomes any error
% before and after 48 hours
pres_one_48_48 = 0; % Define to count number of logical 1
pres_zero_48_48 = 0; % Define to count number of logical 0
for rrrt = 1:numel(n5) % Iterate through all the rows
```

```
    % of pressure anomalies
st3_48_48 = n5(rrrt,1) - hours(48); % Define the start time
et3_48_48 = n5(rrrt,1) + hours(48); % Define the end time
dh3_48_48 = isbetween(e1,st3_48_48,et3_48_48); % Find the match between
% start and end time
if dh3_48_48 == 0
    pres_zero_48_48 = pres_zero_48_48+1; % Count number of 0
else dh3_48_48 == 1
    pres_one_48_48 = pres_one_48_48+1; % Count number of 1
end
end
% Caulculate the number of vibration anomalies becomes
% any error before and after 48 hours
vib_one_48_48 = 0; % Define to count number of logical 1
vib_zero_48_48 = 0; % Define to count number of logical 0
for ssst = 1:numel(n6) % Iterate through all the rows
    % of vibration anomalies
st4_48_48 = n6(ssst,1) - hours(48); % Define the start time
et4_48_48 = n6(ssst,1) + hours(48); % Define the end time
dh4_48_48 = isbetween(e1,st4_48_48,et4_48_48); % Find the match between
% start and end time
if dh4_48_48 == 0
    vib_zero_48_48 = vib_zero_48_48+1; % Count number of 0
else dh4_48_48 == 1
    vib_one_48_48 = vib_one_48_48+1; % Count number of 1
end
end
% probability of any error for voltage anomalies before and after 48 hours
p1_48 = volt_one_48_48/length(n3);
% probability of any error for rotation anomalies before and after 48 hours
p2_48 = rot_one_48_48/length(n4);
% probability of any error for pressure anomalies before and after 48 hours
p3_48 = pres_one_48_48/length(n5);
% probability of any error for vibration anomalies before and after 48 hours
p4_48 = vib_one_48_48/length(n6);
% Create empty matrix with 100 rows and 5 columns
em1 = zeros(100,5);
    em1(iii,1) = iii; % Store machine ID
    em1(iii,2) = p1_48; % Store voltage anomalies to error
    % anomalies before and after 48 hours
    em1(iii,3) = p2_48; % Store rotation anomalies to error
    % anomalies before and after 48 hours
    em1(iii,4) = p3_48; % Store pressure anomalies to error
    % anomalies before and after 48 hours
    em1(iii,5) = p4_48; % Store vibration anomalies to error
    % anomalies before and after 48 hours
```

```
    prob1_48(iii,1) = em1(iii,1); % Machine ID
    prob1_48(iii,2) = em1(iii,2); % Voltage vs error
    % probability before and after 48 hours
    prob1_48(iii,3) = em1(iii,3); % Rotation vs error
    % probability before and after 48 hours
    prob1_48(iii,4) = em1(iii,4); % Pressure vs error
    % probability before and after 48 hours
    prob1_48(iii,5) = em1(iii,5); % Vibration vs error
    % probability before and after 48 hours
end
figure(1)
x1 = prob1_48(:,1);
y1 = prob1_48(:,2:end);
bar(x1,y1,'stacked')
xlabel('Machine ID')
ylabel('Probabilities')
title('Error Probabilities during anomalies time period',...
    'fontweight','bold')
probability1_48 = array2table(prob1_48,'VariableNames',{'MachineID',...
    'volt','rotate','pressure','vibration'});
```

References

[1] File exchange - MATLAB central.
[Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/.

Appendix C

MATLAB code for probability of failure calculation during the anomalies time period.

```
%#####################################################################%
%#####################################################################%
%%%%%%    Master Thesis in Electronics/Automation   %%%%%%%%%%%%%%%%%%
%---------------------------------------------------------------------%
%--------------              Calculate the             -----------%
%--------------   anomalies to failure probability    -----------%
%---------------------------------------------------------------------&
%--------------       University of Gavle             ------------%
%---------------------------------------------------------------------%
%------- Topic: Analysis of Machine Condition to predict Failure %
%---------------------------------------------------------------------%
%-------------- Masters in Electronics/Automation    ------------%
%---------------------------------------------------------------------%
%--------------       Md Abdur Rahman Akash          ------------%
%---------------------------------------------------------------------%
%---------------              January 2022           ------------%
%---------------------------------------------------------------------%
%#####################################################################%
%#####################################################################%
% Load the failure, error and anomalies data
fail = readtable('PdM_failures.csv'); % Load failure data
error = readtable('PdM_errors.csv'); % Load error data
volt = readtable('voltage.csv'); % Load voltage anomalies data
rot = readtable('rotate.csv'); % Load rotation anomalies data
pres = readtable('pressure.csv'); % Load pressure anomalies data
vib = readtable('vibration.csv'); % Load vibration anomalies data
% Calculate the probality of failure after
% 12 hours,24 hours and 48 hours for each machine
% Calculate the probability for all the machines through iteration
for iii = 1:100 % Iterate 100 times as machines are 100
% Load the machine information with iteration
mc1 = error(error.machineID ==iii,:); % Iterate error data 100 times
mc2 = fail(fail.machineID ==iii,:); % Iterate failure data 100 times
mc3 = volt(volt.machineid ==iii,:); % Iterate voltage
% anomalies data 100 times
mc4 = rot(rot.machineid ==iii,:); % Iterate rotate anomalies
% data 100 times
mc5 = pres(pres.machineid ==iii,:); % Iterate pressure anomalies
% data 100 times
mc6 = vib(vib.machineid ==iii,:); % Iterate vibration anomalies
% data 100 times
% Extract the error, failure and anomalies datetime
e1 = mc1.datetime; % Extract error datetime
e2 = mc2.datetime; % Extract failure datetime
e3 = unique(mc3.datetime); % Extract vibration anomalies datetime
e4 = unique(mc4.datetime); % Extract rotation anomalies datetime
```

```
e5 = unique(mc5.datetime); % Extract pressure anomalies datetime
e6 = unique(mc6.datetime); % Extract vibration anomalies datetime
n3 = e3([true; diff(e3) >= hours(48)]); % Extract vibration anomalies datetime
n4 = e4([true; diff(e4) >= hours(48)]); % Extract rotation anomalies datetime
n5 = e5([true; diff(e5) >= hours(48)]); % Extract pressure anomalies datetime
n6 = e6([true; diff(e6) >= hours(48)]); % Extract vibration anomalies datetime
% Caulculate the number of voltage anomalies becomes
% any failure before and after 48 hours
volt_one_48_48 = 0; % Define to count number of logical 1
volt_zero_48_48 = 0; % Define to count number of logical 0
for pppt = 1:numel(n3) % Iterate through all the rows
   % of voltage anomalies
st1_48_48 = n3(pppt,1) - hours(48); % Define the start time
et1_48_48 = n3(pppt,1) + hours(48); % Define the end time
dh1_48_48 = isbetween(e2,st1_48_48,et1_48_48); % Find the match between
% start and end time
if dh1_48_48 == 0
   volt_zero_48_48 = volt_zero_48_48+1; % Count number of 0
else dh1_48_48 == 1
   volt_one_48_48 = volt_one_48_48+1; % Count number of 1
end
end
% Caulculate the number of rotation anomalies becomes any failure
% for before and after 48 hours
rot_one_48_48 = 0;  % Define to count number of logical 1
rot_zero_48_48 = 0; % Define to count number of logical 0
for qqqt = 1:numel(n4) % Iterate through all the rows
   % of rotation anomalies
st2_48_48 = n4(qqqt,1) - hours(48); % Define the start time
et2_48_48 = n4(qqqt,1) + hours(48); % Define the end time
dh2_48_48 = isbetween(e2,st2_48_48,et2_48_48); % Find the match between
% start and end time
if dh2_48_48 == 0
   rot_zero_48_48 = rot_zero_48_48+1; % Count number of 0
else dh2_48_48 == 1
   rot_one_48_48 = rot_one_48_48+1; % Count number of 1
end
end
% Caulculate the number of pressure anomalies becomes any failure
% before and after 48 hours
pres_one_48_48 = 0; % Define to count number of logical 1
pres_zero_48_48 = 0; % Define to count number of logical 0
for rrrt = 1:numel(n5) % Iterate through all the rows
   % of pressure anomalies
st3_48_48 = n5(rrrt,1) - hours(48); % Define the start time
et3_48_48 = n5(rrrt,1) + hours(48); % Define the end time
```

```
dh3_48_48 = isbetween(e2,st3_48_48,et3_48_48); % Find the match between
% start and end time
if dh3_48_48 == 0
   pres_zero_48_48 = pres_zero_48_48+1; % Count number of 0
else dh3_48_48 == 1
   pres_one_48_48 = pres_one_48_48+1; % Count number of 1
end
end
% Caulculate the number of vibration anomalies becomes
% any failure before and after 48 hours
vib_one_48_48 = 0; % Define to count number of logical 1
vib_zero_48_48 = 0; % Define to count number of logical 0
for ssst = 1:numel(n6) % Iterate through all the rows
   % of vibration anomalies
st4_48_48 = n6(ssst,1) - hours(48); % Define the start time
et4_48_48 = n6(ssst,1) + hours(48); % Define the end time
dh4_48_48 = isbetween(e2,st4_48_48,et4_48_48); % Find the match between
% start and end time
if dh4_48_48 == 0
   vib_zero_48_48 = vib_zero_48_48+1; % Count number of 0
else dh4_48_48 == 1
   vib_one_48_48 = vib_one_48_48+1; % Count number of 1
end
end
% probability of any failure for voltage anomalies before and after 48 hours
p1_48 = volt_one_48_48/length(n3);
% probability of any failure for rotation anomalies before and after 48 hours
p2_48 = rot_one_48_48/length(n4);
% probability of any failure for pressure anomalies before and after 48 hours
p3_48 = pres_one_48_48/length(n5);
% probability of any failure for vibration anomalies before and after 48 hours
p4_48 = vib_one_48_48/length(n6);
% Create empty matrix with 100 rows and 5 columns
em1 = zeros(100,5);
   em1(iii,1) = iii; % Store machine ID
   em1(iii,2) = p1_48; % Store voltage anomalies to failure
   % probability before and after 48 hours
   em1(iii,3) = p2_48; % Store rotation anomalies to failure
   % probability before and after 48 hours
   em1(iii,4) = p3_48; % Store pressure anomalies to failure
   % probability before and after 48 hours
   em1(iii,5) = p4_48; % Store vibration anomalies to failure
   % probability before and after 48 hours
   prob1_48(iii,1) = em1(iii,1); % Machine ID
   prob1_48(iii,2) = em1(iii,2); % Voltage vs failure
   % probability before and after 48 hours
```

```
    prob1_48(iii,3) = em1(iii,3); % Rotation vs failure
    % probability before and after 48 hours
    prob1_48(iii,4) = em1(iii,4); % Pressure vs failure
    % probability before and after 48 hours
    prob1_48(iii,5) = em1(iii,5); % Vibration vs failure
    % probability before and after 48 hours
end
prob1_48(6,:) = [6 0 0 0 0]; % Replace the infinity values
prob1_48(77,:) = [77 0 0 0 0]; % Replace the infinity values
prob1_48(6,:) = [6 0 0 0 0]; % Replace the infinity values
prob1_48(77,:) = [77 0 0 0 0]; % Replace the infinity values
figure(1)
x1 = prob1_48(:,1);
y1 = prob1_48(:,2:end);
bar(x1,y1,'stacked')
xlabel('Machine ID')
ylabel('Probabilities')
title('Failure Probabilities during anomalies time period',...
'fontweight','bold')
probability1_48 = array2table(prob1_48,'VariableNames',...
{'MachineID','volt','rotate','pressure','vibration'});
%%
% Save probability of failure and error
save('fail.mat','probability1_48')
% Load error probability
p_e = load('error.mat');
% Load failure probability
p_f = load('fail.mat');
% Define machine number
mac_error_19 = p_e.probability1_48(19,:);
mac_fail_19 = p_f.probability1_48(19,:);
% Extract on one array
all = [mac_error_19;mac_fail_19];
arr = table2array(all);
% Slice the array
with_mac_error = arr(1,:);
wi_out_mac_error = with_mac_error(:,2:end);
with_mac_fail = arr(2,:);
wi_out_mac_fail = with_mac_fail(:,2:end);
x = [1 2 3 4];
all_prob = [wi_out_mac_error;wi_out_mac_fail];
bar(x,all_prob,'stacked')
set(gca,'xtick',1:4,...
 'xticklabel',{'volt','rotate','pressure','vibration'})
legend('probability of error','probability of failure');
```

References

[1] File exchange - MATLAB central.
[Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/.

Appendix D

MATLAB code for data preprocessing, finding peaks and gaussian process regression.

```
%#####################################################################%
%#####################################################################%
%%%%%%    Master Thesis in Electronics/Automation   %%%%%%%%%%%%%%%%%%
%---------------------------------------------------------------------%
%--------------   Data preprocessing, finding peaks   ----------------%
%--------------   and Gaussian process regression (ML) ---------------%
%---------------------------------------------------------------------&
%--------------      University of Gavle              ----------------%
%---------------------------------------------------------------------%
%------- Topic: Analysis of Machine Condition to predict Failure %
%---------------------------------------------------------------------%
%-------------- Masters in Electronics/Automation    ----------------%
%---------------------------------------------------------------------%
%--------------      Md Abdur Rahman Akash            ----------------%
%---------------------------------------------------------------------%
%---------------            January 2022              ----------------%
%---------------------------------------------------------------------%
%#####################################################################%
%#####################################################################%
clc
clear all
close all
%% Load the sensor data
data = readtable('PdM_telemetry.csv'); % Sensor data
error  = readtable('PdM_errors.csv');
fail = readtable('PdM_failures.csv');
%%
% Define the machine number
r = data(data.machineID == input('machine ID =:'),:); % Machine ID
f = fail(fail.machineID == input('machine ID =:'),:);
e = error(error.machineID == input('machine ID =:'),:);
% Split the datetime
[y,mo,d] = ymd(r.datetime);
[h,mi,s] = hms(r.datetime);
% Year
r.year = y;
% Month
r.month = mo;
% Day
r.day = d;
% Hour
r.hour = h;
% Minute
r.minute = mi;
% Second
r.second = s;
```

```
%%
% Convert the categorical failure data to numeric
f.failure = categorical(f.failure);
ft = double(f.failure);
f.failure = ft;
% Convert the categorical error data to numeric
e.errorID = categorical(e.errorID);
et = double(e.errorID);
e.errorID = et;
%%
% Split failure datetime into year, month and day
[yf,mf,df] = ymd(f.datetime);
% Split error datetime into year, month and day
[ye,me,de] = ymd(e.datetime);
f.month = mf;
e.month = me;
% Extract the monthly data
f_1 = f(f.month == 1,:);
e_1 = e(e.month == 1,:);
plot(f_1.datetime,f_1.failure,'r*')
hold on
plot(e_1.datetime,e_1.errorID,'ko')
xlabel('Datetime')
ylabel('Errors and failures')
legend('Failure','Error')
hold off
%%
subplot(2,2,1)
plot(r.datetime,r.volt)
xlabel('Datetime')
ylabel('Voltage data')
title('Raw voltage data')
subplot(2,2,2)
plot(r.datetime,r.rotate)
xlabel('Datetime')
ylabel('Rotation data')
title('Raw rotation data')
subplot(2,2,3)
plot(r.datetime,r.pressure)
xlabel('Datetime')
ylabel('Pressure data')
title('Raw pressure data')
subplot(2,2,4)
plot(r.datetime,r.vibration)
xlabel('Datetime')
ylabel('Vibration data')
```

```
title('Raw vibration data')
%%
a1 = r.volt; % Split the voltage data
a2 = r.rotate; % Split the rotation data
a3 = r.pressure; % Split the pressure data
a4 = r.vibration; % Split the vibration data
s1 = movmean(a1,[12,0]); % Filter with 12 neighbouring points
% of the voltage data
s2 = movmean(a2,[12,0]); % Filter with 12 neighbouring points
% of the rotation data
s3 = movmean(a3,[12,0]); % Filter with 12 neighbouring points
% of the pressure data
s4 = movmean(a4,[12,0]); % Filter with 12 neighbouring points
% of the vibration data
f1 = smoothdata(s1); % Smooth the voltage filted signal edge
f2 = smoothdata(s2); % Smooth the rotation filted signal edge
f3 = smoothdata(s3); % Smooth the pressure filted signal edge
f4 = smoothdata(s4); % Smooth the vibration filted signal edge
r.volt = f1;
r.rotate = f2;
r.pressure = f3;
r.vibration = f4;
%%
subplot(2,2,1)
plot(r.datetime,r.volt)
xlabel('Datetime')
ylabel('Voltage data')
title('Filtered & smoothed voltage data')
subplot(2,2,2)
plot(r.datetime,r.rotate)
xlabel('Datetime')
ylabel('Rotation data')
title('Filtered & smoothed rotation data')
subplot(2,2,3)
plot(r.datetime,r.pressure)
xlabel('Datetime')
ylabel('Pressure data')
title('Filtered & smoothed pressure data')
subplot(2,2,4)
plot(r.datetime,r.vibration)
xlabel('Datetime')
ylabel('Vibration data')
title('Filtered & smoothed vibration data')
%%
g1 = f1; % Define the voltage filtered and smoothed data
g2 = -f2; % Define the inverted rotation filtered and smoothed data
```

```
g3 = f3; % Define the pressure filtered and smoothed data
g4 = f4; % Define the vibration filtered and smoothed data
[pks1,locs1]=findpeaks(g1,'SortStr','descend','Npeaks',22); % Find the
% voltage peaks and plot them
subplot(2,2,1)
plot(g1)
hold on;
plot(locs1, g1(locs1),'r*'); % These red start are the detected peaks
xlabel('time in hours')
ylabel('voltage')
title('Detected voltage peaks')
hold off
[pks2,locs2]=findpeaks(g2,'SortStr','descend','Npeaks',15); % Find the
% rotation peaks and plot them
subplot(2,2,2)
plot(g2)
hold on;
plot(locs2, g2(locs2),'r*'); % These red start are the detected peaks
xlabel('time in hours')
ylabel('rotation')
title('Detected rotation peaks')
hold off
[pks3,locs3]=findpeaks(g3,'SortStr','descend','Npeaks',6);% Find the
% pressure peaks and plot them
subplot(2,2,3)
plot(g3)
hold on;
plot(locs3, g3(locs3),'r*'); % These red start are the detected peaks
xlabel('time in hours')
ylabel('pressure')
title('Detected pressure peaks')
hold off
[pks4,locs4]=findpeaks(g4,'SortStr','descend','Npeaks',10); % Find the
% vibration peaks and plot them
subplot(2,2,4)
plot(g4)
hold on;
plot(locs4, g4(locs4),'r*'); % These red start are the detected peaks
xlabel('time in hours')
ylabel('vibration')
title('Detected vibration peaks')
hold off
%%
m1 = mean(r.volt); % Voltage mean calculation
m2 = mean(r.rotate); % Rotation mean calculation
m3 = mean(r.pressure); % Pressure mean calculation
```

```
m4 = mean(r.vibration); % Vibration mean calculation
std1 = std(r.volt); % Voltage standard deviation calculation
std2 = std(r.rotate); % Rotation standard deviation calculation
std3 = std(r.pressure); % Pressure standard deviation calculation
std4 = std(r.vibration); % Vibration standard deviation calculation
sensor1 = (r.volt - m1) / std1; % Standard scaler on voltage data
sensor2 = (r.rotate - m2) / std2; % Standard scaler on rotation data
sensor3 = (r.pressure - m3) / std3; % Standard scaler on pressure data
sensor4 = (r.vibration - m4) / std4; % Standard scaler on vibration data
r.volt = sensor1;
r.rotate = -sensor2; % Invert rotation data
r.pressure = sensor3;
r.vibration = sensor4;
%%
subplot(2,2,1)
plot(r.datetime,r.volt)
xlabel('Datetime')
ylabel('Voltage data')
title('Scaled voltage data')
subplot(2,2,2)
plot(r.datetime,r.rotate)
xlabel('Datetime')
ylabel('Rotation data')
title('Scaled rotation data')
subplot(2,2,3)
plot(r.datetime,r.pressure)
xlabel('Datetime')
ylabel('Pressure data')
title('Scaled pressure data')
subplot(2,2,4)
plot(r.datetime,r.vibration)
xlabel('Datetime')
ylabel('Vibration data')
title('Scaled vibration data')
%%
% Extract the monthly data
r_10 = r(r.month == 2,:);
rng('default') % For reproducibility
PD = 0.2;
% Perform Nonstratified validation
cv = cvpartition(size(r_10,1),'HoldOut',PD);
% Split into 80% traing
train = r_10(cv.training,:);
% % Split into 20% testing
test = r_10(cv.test,:);
%%
```

```
% Train the model by changing the response variable name
gprMdl = fitrgp(train,'volt~rotate+pressure+vibration+month+day+hour',...
'KernelFunction','exponential',...
    'FitMethod','fic','PredictMethod','fic','Regularization',0.05);
%%
% Define the predictors on the testing data
x1 = [test(:,4),test(:,5),test(:,6),test(:,8),test(:,9),test(:,10)];
% Actual test data
y1 = test.volt;
% Predicted respons data with the model
ypred = predict(gprMdl,x1)
figure;
plot(y1,'b--');
hold on;
plot(ypred,'r');
xlabel('Timestamps')
ylabel('Predicted vibration response')
legend('Real test data','Predicted data','Location','Best');
hold off
% R-square calculation
Rsq = 1 - sum((y1 - ypred).^2)/sum((y1 - mean(y1)).^2);
% RMSE calculation
RMSE = immse(y1,ypred);
% MAE calculation
MAE = mean(abs(y1-ypred));
% RAE calculation
RAE = sum((y1 - ypred))/sum(abs(y1 - mean(y1))); [1]
```

References

[1] File exchange - MATLAB central.

[Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/.