ANALYSIS AND MONITORING OF PUMP AND MOTOR ASSEMBLY VIBRATION TO IMPROVE PERFORMANCE THROUGH SUPERVISED LEARNING

A

Dissertation

submitted in partial fulfillment of the

requirements for the award of the degree of

MASTER OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWORK

by

Umang Agarwal (R102213012)

under the guidance of

Mr. Anil Kumar

Assistant Professor (UPES)

Dr. Rashmi Sharma

Assistant Professor (UPES)

Mr. Amit Purohit

External Guide (ABB GISL, Bangalore)

Department of Computer Science & Engineering

Centre for Information Technology

University of Petroleum & Energy Studies

Bidholi, Via Prem Nagar, Dehradun, UK

May – 2015

CANDIDATE'S DECLARATION

I hereby certify that the project work entitled **"ANALYSIS AND MONITORING OF PUMP AND MOTOR ASSEMBLY VIBRATION TO IMPROVE PERFORMANCE THROUGH SUPERVISED LEARNING"** in partial fulfilment of the requirements for the award of the Degree of MASTER OF TECHNOLOGY in ARTIFICIAL INTELLIGENCE AND ARTIFICIAL NEURAL NETWORK and submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **January, 2015** to **May, 2015** under the supervision of Mr. Anil Kumar, Assistant Professor, UPES, Dr. Rashmi Sharma, Assistant Professor, UPES and Mr. Amit Purohit, External Guide, ABB GISL, Bangalore.

The matter presented in this project has not been submitted by me for the award of any other degree of this or any other University.

Umang Agarwal
R102213012

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Mr. Anil Kumar
Project Guide

Dr. Rashmi Sharma
Project Guide

Dr. Amit Agarwal
Program Head – M.Tech Artificial Intelligence and Artificial Neural Network
Center for Information Technology
University of Petroleum & Energy Studies
Dehradun – 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

I wish to express my deep gratitude to my guides **Mr. Anil Kumar, Dr. Rashmi Sharma** and **Mr. Amit Purohit,** for all advice, encouragement and constant support they have given me throughout my project work. This work would not have been possible without their support and valuable suggestions.

I sincerely thank to my respected Program Head of the Department, **Dr. Amit Agarwal**, for his great support in doing my project in **ANALYSIS AND MONITORING OF PUMP AND MOTOR ASSEMBLY VIBRATION TO IMPROVE PERFORMANCE THROUGH SUPERVISED LEARNING** at **CIT** and my course coordinator **Mr. Vishal Kaushik** for his continuous support and guidance.

I am also grateful to **Dr. Manish Prateek, Associate Dean** and **Dr. Kamal Bansal, Dean CoES, UPES** for giving me the necessary facilities to carry out my project work successfully.

I would like to thank all my **friends** for their help and constructive criticism during my project work. Finally I have no words to express my sincere gratitude to my **parents** who have shown me this world and for every support they have given me.

Umang Agarwal
R102213012

ABSTRACT

Computerized control systems that monitor, operate, and diagnose process variables such as pressure, flow, and temperature have been implemented for various processes. When these systems are for large-scale processes, they generate many process variable values, and operators often find it difficult to effectively monitor the process data, analyze current states, detect and diagnose process anomalies, and/or take appropriate actions to control the processes. To assist plant operators, decision support systems that include artificial intelligence (AI) and non-AI technologies are needed for the tasks of monitoring, operating, and diagnosis. The support systems can be implemented based on the data-driven, analytical, and knowledge-based approach so that, it is presented in a manner that reflects the important underlying trends or events in the process. In this work we have considered a pump and motor system and analyzed it various operating conditions and implemented an automated learning system to recognize the different environmental conditions and generate an alert level for the plant operator.

TABLE OF CONTENT

Candidate Declaration	i
Acknowledgement	ii
Abstract	iii
Table of Contents	iv
List of Figures	vii
List of Tables	ix
Chapter1. Introduction	1
1.1 Motivation	1
1.2 Problem Description	2
1.3 Our Contribution	3
1.4 Thesis Outline	4
Chapter2. Literature Survey	5
2.1 Related Work	5
2.1.1 Dynamic Time Warping	5
2.1.2 Longest Common Subsequence	7
2.1.3 Edit Distance with Real Penalty	7
2.1.4 Edit Distance with Real Sequence	8
2.1.5 Statistical Process Monitoring Techniques	9
2.1.6 Expert Systems	11
2.1.7 Neural Network Approach	12
2.2 Data Mining Techniques	12
2.2.1 Classification Tasks	13
2.2.2 Process Data Mining	13
Chapter3. Preliminaries	14
3.1 Definition and Notations	14
3.1.1 Time Series	14
3.1.2 Instruments	14
3.1.3 Sensors	14

3.1.4 Univariate Analysis 14
3.1.5 Multivariate Analysis 14
3.1.6 Process States 14
3.1.7 Moving Window 15
3.1.8 Key Performance Indicator 15
Chapter4. Process Data Analysis 16
4.1 Data Pre-Processing 17
4.1.1 Separation of Datasets 17
4.1.1.1 Calculation of Variances 19
4.2 Feature Generation 19
4.2.1 Feature Selection 19
4.2.2 Identifying Feature for Time Series Data 20
4.2.3 Feature Extraction Parameters Definitions and Notations 20
4.2.4 Calculating Feature Values 22
4.2.5 Formulating Feature Parameters 22
4.3 Creating Process Data Feature Profiles 24
4.4 Feature Selection for Classification 25
Chapter5. Algorithms and Visualization Techniques 27
5.1 Classification Algorithms 27
5.1.1 Naïve Bayes Classification 27
5.1.2 Nearest-Neighbor Classification 30
5.1.3 Classification and Regression Trees (CART) 32
5.1.4 Discriminant Analysis 34
5.2 Comparison of Classification Algorithms 38
5.3 Visualization Techniques 41
Chapter6. Experiment and Results 44
6.1 Data and Setup 44
6.2 Data Pre-Processing 44
6.2.1 Separation of raw data for Feature Classification 44
6.2.2 Time Series Feature Calculation 46

6.3 Classification of Process States 47
Chapter7. Conclusion and Future work 53
Chapter8. References 54
Chapter9. Appendix A 56

LIST OF FIGURES

1. Chapter 2

Figure2.1 Dynamic Time Warping 5

Figure 2.2: Representation of Control Chart 10

Figure 2.3: Representation of Histogram 10

Figure 2.4: Knowledge-based feedback planner used as controller 11

Figure 2.5: Plant control using Neural Nets 12

2. Chapter 4

Figure 4.1: X-axis process data variation 17

Figure 4.2: Y-axis process data variation 17

Figure 4.3: Z-axis process data variation 18

Figure 4.4: Variance plot for Z-axis with moving window 200 18

Figure 4.5: Root Mean Square vs Standard Deviation plot for Moving Window size 21 23

Figure 4.6: Root Mean Square vs Standard Deviation plot for Moving Window size 900 23

Figure 4.7: Line plot representation 25

Figure 4.8: Scatter plot representation 26

3. Chapter 5

Figure 5.1: Classification done by Naïve Bayes 29

Figure 5.2: Clusters generated by k-means for classification 32

Figure5.3: Classification Tree generated by CART for classification 34

Figure 5.4: Execution Time (Training and Testing) for Classification Algorithms 39

Figure 5.5: Total number of miss-classifications done by classification algorithms 39

Figure 5.6: Scatter Plot for Standard deviation and Shape Factor 40

Figure5.7: Three-tone coloring for different vacuum motors showing their current state 42

Figure5.8: Scatter plot showing deviation of crest factor and kurtosis value from normal operating condition 42

Figure5.9 Cluster representation of the dataset features 43

4. Chapter 6

Figure 6.1: Separation of ON and OFF state data by variance difference 45

Figure 6.2 Showing feature values variation for moving window size 1350 46

Figure 6.3 Showing feature values variation for moving window size 1350 47

5. Appendix A

FigureA.1: Vacuum pump conveying system 57

FigureA.2: Internal architecture of piston pump 58

FigureA.3: Showing placement of sensors in pump and motor system 59

FigureA.4: Sensor reading in the 3-co-ordinate axis 59

LIST OF TABLES

1. Chapter 4

Table 4.1: Sample time series process data 16

Table 4.2: Sample feature values for moving window size 1350 24

2. Chapter 5

Table 5.1: Parameters comparison of classification algorithms 38

3. Chapter 6

Table 6.1: Miss-classification rate using Discriminant analysis for X-axis features 48

Table 6.2: Miss-classification rate using Discriminant analysis for Y-axis features 48

Table 6.3: Miss-classification rate using Discriminant analysis for Z-axis features 48

Table 6.4: Miss-classification rate using Naïve Bayes for X-axis features 49

Table 6.5: Miss-classification rate using Naïve Bayes for Y-axis features 49

Table 6.6: Miss-classification rate using Naïve Bayes for Z-axis features 49

Table 6.7: Miss-classification rate using CART for X-axis features 50

Table 6.8: Miss-classification rate using CART for Y-axis features 50

Table 6.9: Miss-classification rate using CART for Z-axis features 50

Table 6.10: Miss-classification rate using Nearest-Neighbor for X-axis features 51

Table 6.11: Miss-classification rate using Nearest-Neighbor for Y-axis features 51

Table 6.12: Miss-classification rate using Nearest-Neighbor for Z-axis features 51

CHAPTER 1

Introduction

1.1 Motivation

When we consider an industry where monitoring is done for large area, operating of each section involved in the industry is a big task. It involves huge human resource and time. These two factors can be overcome by developing some technology which makes use of single operative for monitoring and operating the entire area.

The need to maintain better product quality while improving production rates has driven the industry to automate the machining operations. Increases in labor and raw material costs have also provided the incentive to increase production throughput as well as reduce the number and technical skill levels of the required workforce. A disadvantage of this trend is that a change in the condition of the machining process or the resultant quality of the work piece can go undetected for a considerable period of time causing significant impact on the economics of the machining operation. Substantial research has been conducted in developing remote sensing strategies to aid machine operators in detecting unwanted machining conditions or assisting in maintaining proper machining parameters regardless of the condition of the tools.

Research in the area of process monitoring and operating has included determining the "best" sensor for monitoring the operation as well as various abrasive machining operations. This research included monitoring different variables and determining the current working state of the system for efficient process operation.

Process monitoring and operating is a combination of architectures, mechanisms, and algorithms used in the industrial factory for monitoring and control the activities of a specific process to achieve the goal.

Complex processes involve many process variables, and operators faced with the tasks of monitoring, operating, and diagnosis of these processes often find it difficult to efficiently monitor the process data, analyze current states, detect and diagnose process anomalies, or take appropriate actions to control the processes.

Computerized control systems that monitor, operate, and diagnose process variables such as pressure, flow, and temperature have been implemented for various processes. When these systems are for large-scale processes, they generate many process variable values, and operators often find it difficult to effectively monitor the process data, analyze current states, detect and diagnose process anomalies, and/or take appropriate actions to control the processes.

To assist plant operators, decision support systems that include artificial intelligence (AI) and non-AI technologies have been adopted for the tasks of monitoring, operating, and diagnosis. The support systems can be implemented based on the data-driven, analytical, and knowledge-based approach so that, it is presented in a manner that reflects the important underlying trends or events in the process

The main objective of this thesis is to introduce the data-driven, analytical, and knowledge-based approaches for developing solutions in intelligent support systems for process manufacturing plants.

1.2 Problem Description

In this work, we address the problem of monitoring and guiding a plant operator using real time sensor data obtained for different instruments. The sensor data for various instruments vary for different times because of the varying operating conditions. We address four different problems in designing efficient data monitoring and guidance system for the process plant operator.

Sensor Data Analysis: Given the instruments data collected by sensors attached to them, we analyze this data using various statistical measures for time series data. We present data analysis procedures for deriving various features for time series sensor data and use these features for determining the various operating conditions.

Data Classification: We use the sensor data analysis procedure to compute feature profiles for the different time series data obtained for different instruments. We address the problem of efficiently finding the normal and abnormal operating conditions along with different operating conditions.

Classification of real time sensor data: After we obtained profile features for different system operating conditions we determine the current working state of the system for the purpose of proper flow of the process by alerting the operator if any anomaly occurs.

Process Visualization: When so many variables are involved in the process flow then it becomes extremely difficult to analyze this large amount of data. This data can better be perceived with the help of graph plots and using different coloring schemes for different variables. This is another area of focus in this thesis.

1.3 Our Contribution

Monitoring any process by analyzing the incoming sensor data by which we can define the operational state of the process is the primary task. Operators have defined tasks of continuously monitoring the ongoing process and analyze each and every detail about the ongoing process state, if any abnormal state occurred and system is running normally. If any deviation from normal condition is seen then taking appropriate necessary actions to eliminate it.

We start with the time series sensor data corresponding to different instruments. We present processing procedures over these datasets which address the problem of calculating the different time series features for these datasets. We generate these time series features for finding or separating out different operating conditions in a process plant for different instruments.

After generating these feature datasets and classifying the instrument operational state we try to assist the plant operator for the disturbance occurred and what necessary actions should be taken to prevent it.

We also tried to define visualization techniques like two-tone coloring and representing time series data with the help of different plots so they can be effectively visualized and analyzed by the plant operators.

1.4 Thesis Outline

Chapter 2 presents the summary of previous work, literature survey related to techniques of data mining in context of time series data and some standard features for time series identification.

Chapter 3 outlines the basic definitions, notations and fundamentals which have been used throughout the thesis.

Chapter 4 presents sensor time series data analysis done for generating the time series features for different instruments. Details about different time series properties and methods to calculate them are described.

Chapter 5 describes different classification algorithms for determining operational states of the instruments. Also, it discusses some methods to effectively visualize this time series data.

Chapter 6 describes the experimental process and results of proposed work.

Chapter 7 concludes the thesis and also provides some pointers about possible future work.

Chapter 2

Literature Survey

2.1 Related Work

In this chapter, we discussed about the existing work and techniques for matching similarity over the time series data. We mention different existing work along with their applicability and the type of time series data for which they are better suited. This chapter presents the mathematical formulations of the existing models to calculate time series data and hoe each method is better than the other and what are the limitations of each method.

This chapter also discusses some existing process monitoring and control strategies being deployed in industries to carry out smooth flow of the machine process.

2.1.1 Dynamic Time Warping

Dynamic time warping is a similarity method for comparing two different time series which allows phase shift between two or more time series [1]. Euclidean distance is used for measuring the difference between two or more entities in many places, but Euclidean distance is not suitable for calculating the similarity where the time series data is out of phase, see Figure 2.1. The time series plot shown in Figure 2.1 is identical but calculating Euclidean distance between them will show them as completely different time series.

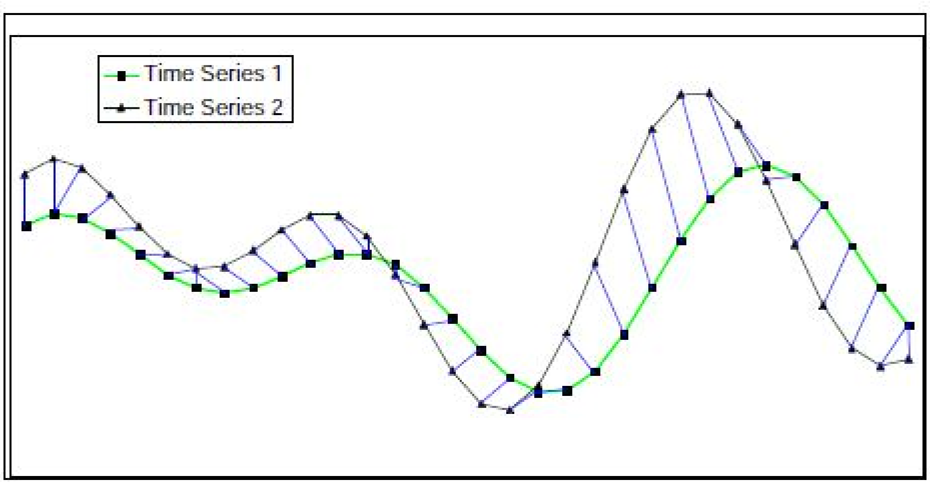


Figure 2.1: Dynamic Time Warping [1].

Dynamic time warping is a more better and appropriate method or technique for calculating similarity between two or more different time series data and it also allows phase shifting. The distance between two time series by Dynamic time warping is calculated by the formula-:

$$X = \min \left(D^2_{DTW}(a, Rest(b)),\ D^2_{DTW}(a, Rest(b)),\ D^2_{DTW}(Rest(a), Rest(b))\right) \tag{2.1}$$

$$D^2_{DTW}(a, b) = D^2(First(a), First(b)) + X \tag{2.2}$$

First(a) represents the first value of the time series a, and Rest(a) represents remaining values of the time series a apart from the value contained in First(a).

One major drawback of Dynamic time warping is that it violates triangle inequality which is given by-:

$$D(a,c) \le D(a,b) + D(b,c) \text{ ,for any data x, y, z} \tag{2.3}$$

Triangle inequality helps in providing indexing structures by which track of the similar points over time series data can be maintained. So, DTW does not have any indexing structure. Let, us consider three different time series represented by a, b and c-:

a=1, 1, 1, 2, 2, 2

b=1, 1, 2, 2, 2, 2

c=1, 1, 1, 1, 2, 2

The distance calculated by dynamic time warping between these three time series a, b and c is D(a, b) = 0, D(a, c) = 0 and D(b, c) = √2 respectively. Here, we can see that dynamic time warping does not follow triangle inequality as-: $D(a,c) > D(a,b) + D(b,c)$.

Since, dynamic time warping can handle phase shifting so, it is more better than the calculating euclidean distance technique. Another advantage of dynamic time warping is that it can match time series of different lengths where Euclidean distance can only measure similarity between the time series having similar length.

The drawback of DTW with respect to Euclidean distance is that the distance computation cost between two points is higher than that in Euclidean distance.

2.1.2 Longest Common Subsequence

Longest common subsequence method or technique is a time series similarity matching method which can also handle time series data containing noise. The noise present in the time series can be because of the hardware malfunctioning, distorted signals, and errors during data transmission. Longest common subsequence has the ability to remove the effect produced by noise in the time series data by counting only the similar data elements from the time series [2].

The distance given by LCSS comparing two different time series A and B having lengths P and Q respectively is given as-:

$$\begin{aligned} LCSS(A, B) &= 0, \text{ if } P = 0 \text{ or } Q = 0 \\ &= LCSS(Rest(A), Rest(B)) + 1, \text{ if } distancw(a_1, b_1) \leq \varepsilon \\ &= \max\{LCSS(Rest(A), B), LCSS(A, Rest(B))\}, \text{ otherwise} \end{aligned} \quad (2.4)$$

where distance(a_1, b_1) = |(a_1, b_1)| .

Distance measure in LCSS is defined as "score", so more the score calculated by the equation 2.4 between different time series more closer they are. The LCSS give the score which can be converted into the distance measure between two time series by the formula-:

$$LCSS_{distance}(A,B) = 1 - \frac{LCSS(A,B)}{\min(|A|,|B|)} \quad (2.5)$$

In this method, a threshold value ε is used which determines the similarity between two different elements of the time series data. LCSS eliminates the effect of noise during matching the different time series as this threshold ε quantizes the distance between the two elements of the time series data by, 0 and 1, which is helpful in eliminating the large distance between two points in the time series generated by the unwanted noise.

2.1.3 Edit Distance with Real Penalty

Phase shifting or, local time shifting is phenomenon which occurs mostly in every time series data, to eliminate this effect we can take an idea from the sting matching methods. A string is represented by sequence of alphabets or symbols. When two different strings of different lengths are compared together they are aligned in such a manner that they both become identical by adding, deleting or changing the smallest number of alphabets or symbols [2]. These different operations of adding, deleting or changing the symbols or alphabets in a string can all be considered as a single operation of adding the symbols or alphabets. The added symbol or alphabet in the string is known as 'gap' element. The distance defined by this method between two or more different time series is called 'string edit distance' which is defined in next paragraph.

Suppose we have to different string A and B whose respective lengths are given by P and Q, the Edit Distance (ED) between the two time series A and B is the total number of altering operations performed which are required to convert A into B. ED(A, B) is given by the formula-:

$$
\begin{aligned}
ED(A, B) &= P, \text{ if } Q=0 \\
&= Q, \text{ if } P=0 \\
&= ED(Rest(A), Rest(B)), \text{ if } first(A) = first(B) \\
&= \min\{ED(Rest(A), Rest(B)) + 1, \\
&\qquad ED(Rest(A), B + 1, \qquad \text{otherwise} \\
&\qquad ED(A, Rest(B)) + 1\},
\end{aligned} \tag{2.6}
$$

where the gap element is introduced in the string with a cost/distance of 1, which is given by the formula-:

$$
\begin{aligned}
distance(a_i, b_i) &= 0, \text{ if } a_i = b_i \\
&= 1, \text{ if } a_i \text{ or } b_i \text{ is a gap element} \\
&= 1, \text{ otherwise}
\end{aligned} \tag{2.7}
$$

2.1.4 Edit Distance on Real Sequence

Edit distance on real sequence (EDR) is another method or technique for eliminating phase or local time shifting and removing noise from the time series data. EDR is more accurate and strong method to calculate similarity between two time series representation than any other existing method or technique.

Suppose, we have two time series element vectors a_i and b_j from two different time series A and B respectively, and they are said to be similar (match(a_i, b_j) = true) if and only if $| a_{i,x} - b_{j,x} | \leq \varepsilon$ and $| a_{i,y} - b_{j,y} | \leq \varepsilon$, where ε is the similarity threshold defined for comparing the time series.

Given two time series A and B with different length P and Q respectively, than the Edit distance on real sequence between the two time series A and B is defined by the total number of insertion, replacing or delete operations performed which are needed to convert time series A into B. So, the distance function for EDR is given by-:

$$
\begin{aligned}
EDR(A, B) &= P, \text{ if } Q = 0 \\
&= Q, \text{ if } P = 0 \\
&= \min\{EDR\{Rest(A), Rest(B) + subcost, \\
&\qquad EDR(Rest(A), B) + 1, \qquad \text{otherwise} \\
&\qquad EDR(A, Rest(B)) + 1\}
\end{aligned}
\tag{2.8}
$$

where subcost = 0 if match(a_1, b_1) = true and subcost = 1 otherwise.

In EDR the effect of disturbance in time series data is eliminated by quantizing the distance between the two elements between which distance is calculated with the help of the threshold ε.

Similar to ERP, determining the total number of minimum altering operations which are needed to change one time series data into another is the advantage for EDR to eliminate phase or local time shifting.

Opposite to that of LCSS, EDR uses penalties for the gap between two similar sub sets of time series data which is calculated based on the lengths of the gap between the two vectors, which gives EDR more advantage than LCSS.

2.1.5 Statistical Process Monitoring Techniques

Statistical monitoring of the process is a method or technique based on logical decision making. It is a graphical representation of the ongoing current process, by analyzing this statistical analysis plot operatives can decide if a process is working correctly or not. These representations can be in the form of the connected-point charts. The current activity of the ongoing process is represented with the help of points in 2-dimensional i.e., x/y axis plot where x-axis mostly represents the time and y-axis represents the changing behavior of the process. The points represented in the plots can be either individual measurements of any tool or, they can be averages of various variables or subparts with in the whole process [3].

Information in statistical process monitoring is represented with the help of numbers so, all the decision making and actions to be taken in particular situation all depends on this number information. To record these numbers process monitoring system must have a data recording system installed. For interpretation of this information or numbers there should be some set of tools present so that maximum information or knowledge can be obtained from the data. Some methods of extracting the useful information or knowledge used in statistical process monitoring are mentioned below-:

Process flowcharting – describing the process flow.

Check sheets/tally charts – frequency of occurrence of any event.

Histograms – summarization of the total process flow.

Graphs – representing the flow of process in time series.

Pareto analysis – describes the problems in the process.

Cause and effect analysis and brainstorming – reasons for the occurrence of problems.

Scatter diagrams – similarity or relationship between process factors.

Control Charts – variations to be controlled and ways to control it.

The information or numbers in these plots may vary significantly over time, recognizing when the variation is normal and when it is not is the main challenge and step in statistical process control.

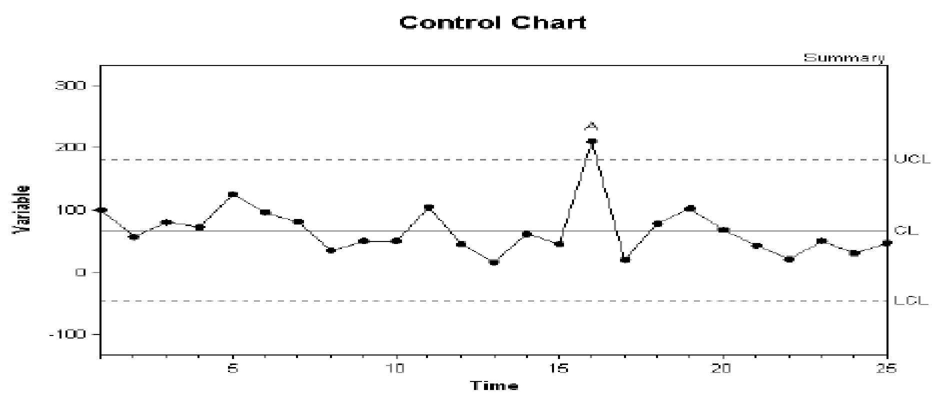


Figure 2.2: Representation of Control Chart [3].

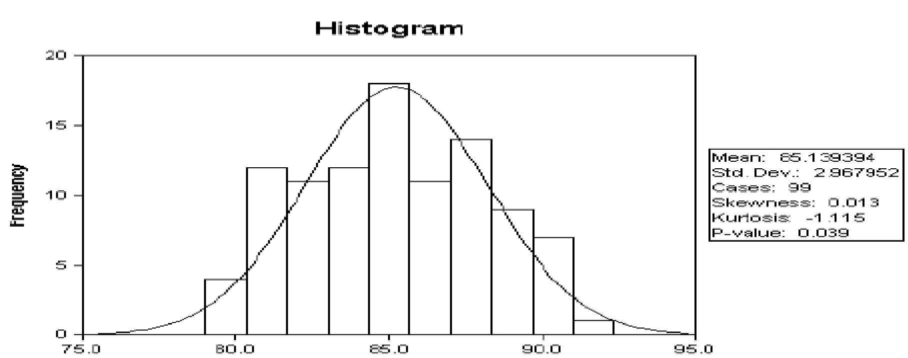


Figure 2.3: Representation of Histogram [3].

2.1.6 Expert Systems

Expert systems are computer programs, which provide a man-machine interface for providing solution to some specific problem. These computer programs are coded with real world knowledge and symbolic reasoning for the purpose of performing complicated task, matching the level that of a human. Supervising the process and controlling requires knowledge and experience, which may sometimes be not achieved by skilled human operatives and engineers. The system is modelled with estimations and algorithms for controlling the process which are embedded with a knowledge-bank which captures human knowledge of design and operational practices.

The main objective of knowledge-based systems is that the information required by the system to control the total work flow should work explicitly and not implicitly. In a computer program the algorithm is designed in the form of a code, but in expert system objective is to mention the rules in an intuitive and easy way that can be understood by everyone, and even be analyzed and updated by domain experts.

Many different outputs are generated during the process flow and these outputs depend on many factors and variables which can be measured by some data-recording device. Abnormal variations change or alter these normal outputs by generating some random events in the ongoing process. The specifications are laid for the normal desired performance of the process and this information is known to the system designer. The goals and outputs of the machine should be analyzed so that proper control signals can be generated to cancel out the effect of the abnormal random events generated in the system and design specifications are full-filled by the machine.

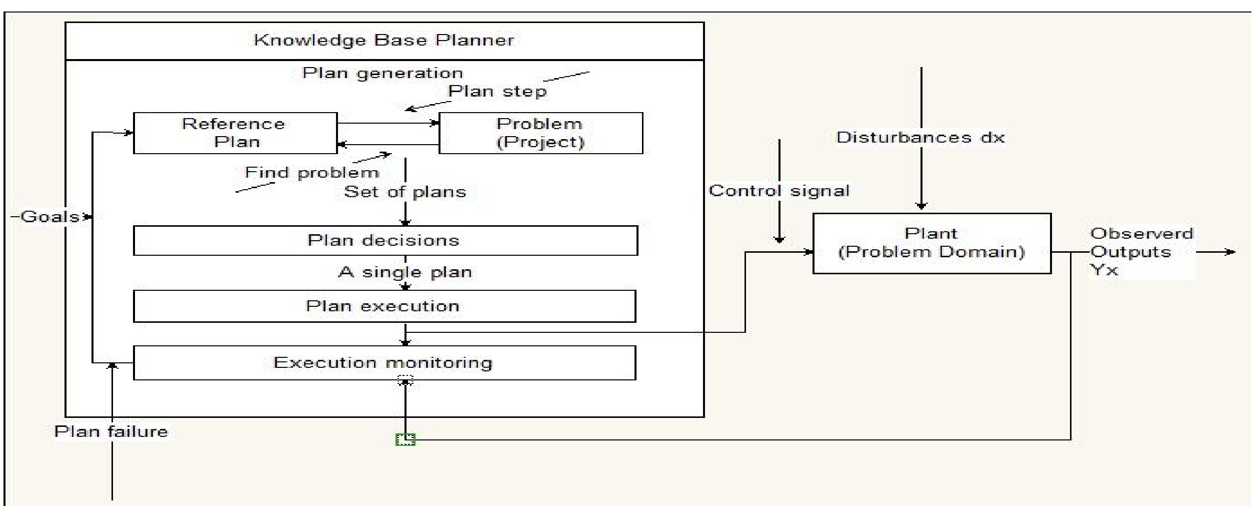


Figure 2.4: Knowledge-based feedback planner used as controller [4].

2.1.7 Neural Network Approach

Neural Networks comes into picture in process control when the behavior of the system is very complex and non-linear and then these process models are represented with the help of sigmoid functions. The use of neural networks increased in process control because of- need of complicated system to deal with complex situations, rise in the demand of design requirements and to fulfill all these demands without any deep and advanced knowledge of the process plant.

The neural networks required for these kinds of operations can be designed with the help of determining the relationship with the input and output of the process plant. Different experiments are carried to train the neural networks on this input and output data. Also, in some industries controllers (P/PI/PID-controllers)are used to provide stability to the plant and some basic control structure.

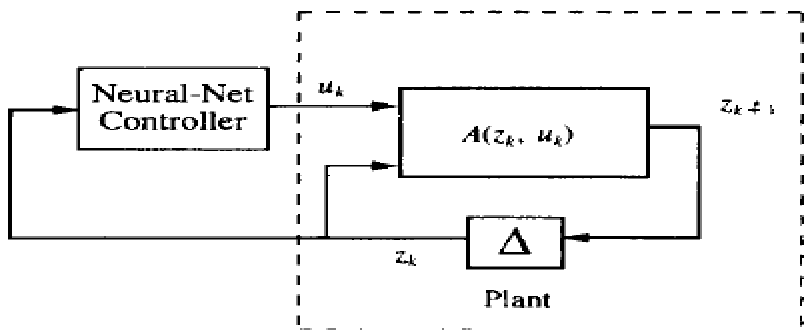


Figure 2.5: Plant control using Neural Nets [5].

With the help of the input vectors μ_k, the state of the machine represented by vector z_k at time k we can map these three inputs to define the machine function $A(z_k, \mu_k)$. The main challenge of using neural networks in control plants is to obtain correct input to determine the non-linear model of the plant. The common solution to this problem is achieved by linearizing the operating points of the plants over a specific set of points and then modelling a linear model of the plant for these operating points.

2.2 Data Mining techniques

Extracting the hidden knowledge from data is an interesting problem. While the traditional databases are capable of showing the stored data, there might exist non trivial relationships between individual records which cannot be directly revealed by the databases. Data mining refers to the process of revealing these hidden relationships in existing data by means of different techniques and algorithms. There are several tasks which are required to be addressed, which may include data classification/aggregation/prediction/association, and different algorithms are employed to accomplish these tasks. We discuss some of the data mining techniques which are relevant for process data analysis.

2.2.1 Classification Tasks

In classification tasks, all the records in the existing data sets are mined in order to find characterization. Each record is assumed to belong to a particular class or category which is appropriately defined based on the data characteristics and classification algorithms are used to automatically determine the category or class to which an individual record belongs. It involves supervised learning since the categories are required to be defined before the classification process begins. Classes must be defined based on the value of the attribute. An input entity is classified into the class based on the closeness or similarity to predefined class or category.

With respect to process data analysis, movement of the process can be described using classification decision trees which are useful in visual surveillance and tracking process movements. For classification using Decision Trees, current sensor data is compared with the defined limits for a process to be grouped in certain class.

The choice of function or algorithm used for classification depends on the *Key Performance Indicator* (KPI). For example, in order to classify certain pattern of data from the data generated by the sensors, the one with least number of miss-classification algorithms are used. This analysis also facilitates extraction of relations between different classes and the overall dataset.

2.2.2 Process Data Mining

Considerable research is available which extend the existing data mining techniques in the context of time series data for process flow analysis. In (useful 4), the authors discuss the techniques to discover similarity in the time series data which correspond to the current working state for the ongoing process. The authors propose different algorithm which matches the time series based on different distance functions based on the datasets represented by the time series. It also proposed a time series matching algorithm which is also defined for noisy data. The authors also discuss the need of indexing techniques to keep track of the total similarity points within the two or more different time series.

CHAPTER 3

Preliminaries

In this chapter, we formally define the terminology used to describe the formulations presented in the thesis.

3.1 Definitions and Notations

3.1.1 Time Series

Time series is a representation of the events w.r.t, the time of their occurrence or, it defines a particular set of events or data points in the order in which they occurred.

3.1.2 Instruments

Instrument refers to some equipment with some measuring sensors which is used in process industry for carrying out process flow.

3.1.3 Sensors

Sensor can be defined as a data-reading device which measures the current value or measure of the system or unit it is attached to.

3.1.4 Univariate Analysis

Univariate analysis is a quantitative (statistical) analysis in which observation and analysis of only one variable is done at a single instance of time.

3.1.5 Mutivariate Analysis

Mutivariate analysis is a quantitative (statistical) analysis in which observation and analysis of more than one variable is done at a single instance of time.

3.1.6 Process States

Process states refers to the different operating or working conditions of an instrument under some circumstances.

3.1.7 Moving Window

Moving window signifies dividing our datasets into subsets with each subset carrying the data points equal to the size of window and then performing the specified operation of the data subsets. In other words Moving Window is the specified length of the subsets of data.

3.1.8 Key Performance Indicator

A Key Performance Indicator (KPI) is a measurable value that demonstrate how effectively a process is achieving the desired results. In our thesis it is the total number of miss-classifications done by our classification algorithms.

CHAPTER 4

Process Data Analysis

Raw data in the form of time series is collected from sensor enabled pump system which provide information about the following four attributes: Time Stamp of the reading being taken, x-axis reading of the sensor, y-axis reading of the sensor, z-axis reading of the sensor.

Time Stamp (HH:MM:SS)	X-axis data (in mm/s)	Y-axis data (in mm/s)	Z-axis data (in mm/s)
13:47:40	-0.5	0.1	-0.1
13:47:40	-0.7	0.3	-0.4
13:47:41	-1.4	0.4	-0.6
13:47:41	-1.2	0.5	-0.8
13:47:41	-1.0	0.4	-0.6
13:47:42	-0.5	0.4	-0.7
13:47:42	-0.7	0.1	-0.7
13:47:42	-0.6	0.1	-0.7
13:47:42	-0.9	0.2	-0.7
13:47:43	-0.9	0.3	-0.8
13:47:43	-0.8	0.1	-0.8

Table 4.1: Sample time series process data.

Indication about the process state flow can be derived from the data coming in from sufficiently large number of sensors and instruments. We have acquired the data corresponding to time stamp for a particular time span. We have processed this time series data to determine the current state of our process. The details of processing have been provided in this chapter.

4.1 Data Preprocessing

4.1.1 Separation of Datasets

In this preprocessing stage, the sensor data is firstly separated on the basis of ON and OFF states. Sensors take about 3-4 readings in a span of 1 second. Approximately, sensors provides us with 200 readings in 1 minute. The instrument has two different states namely, ON and OFF states. For a span of 45 seconds it is in ON state and for the next 45 seconds OFF state likewise, alternatively changing their states. So, our first task is to separate out this ON and OFF state data. For separating the data we have three different axis x/ y/ z-axis. We separated the two states based on the z-axis readings because they provide much more visible separation rather than x/ y-axis readings. It can be seen in the figures below.

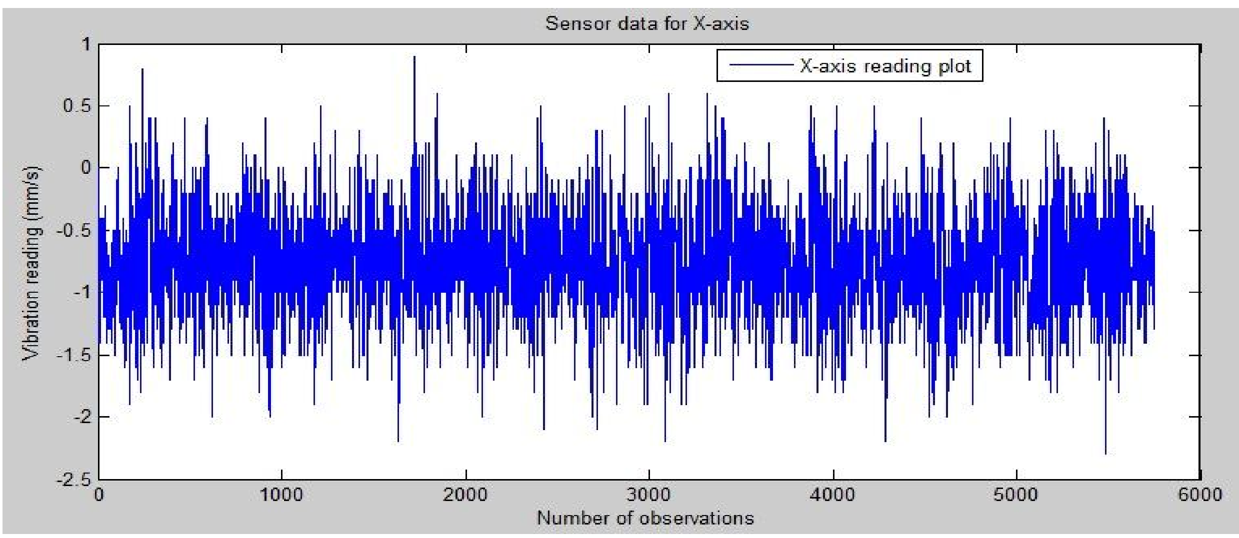


Figure 4.1: X-axis process data variation

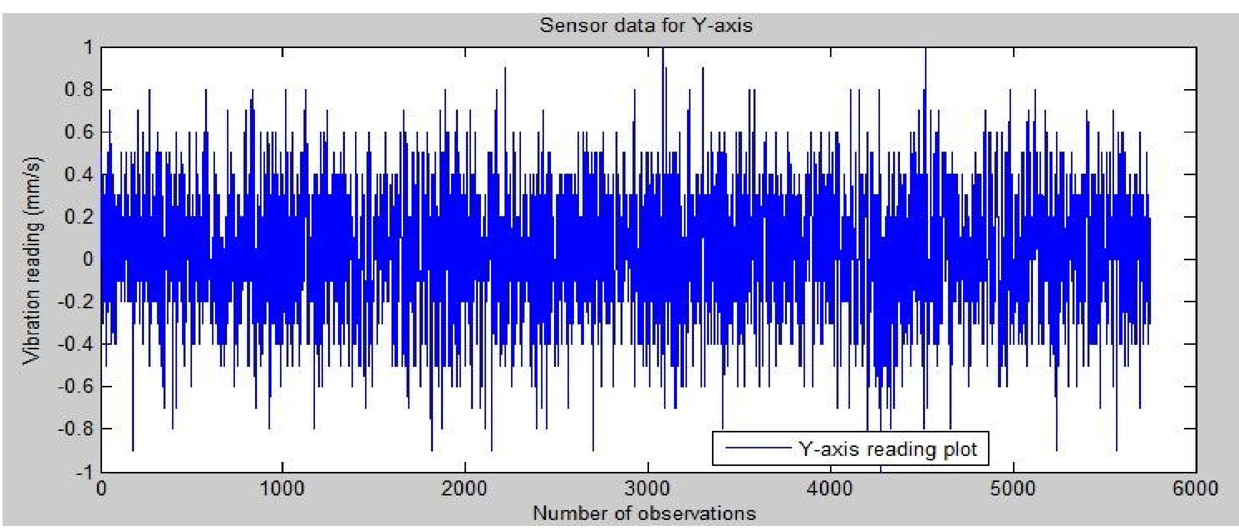


Figure 4.2: Y-axis process data variation.

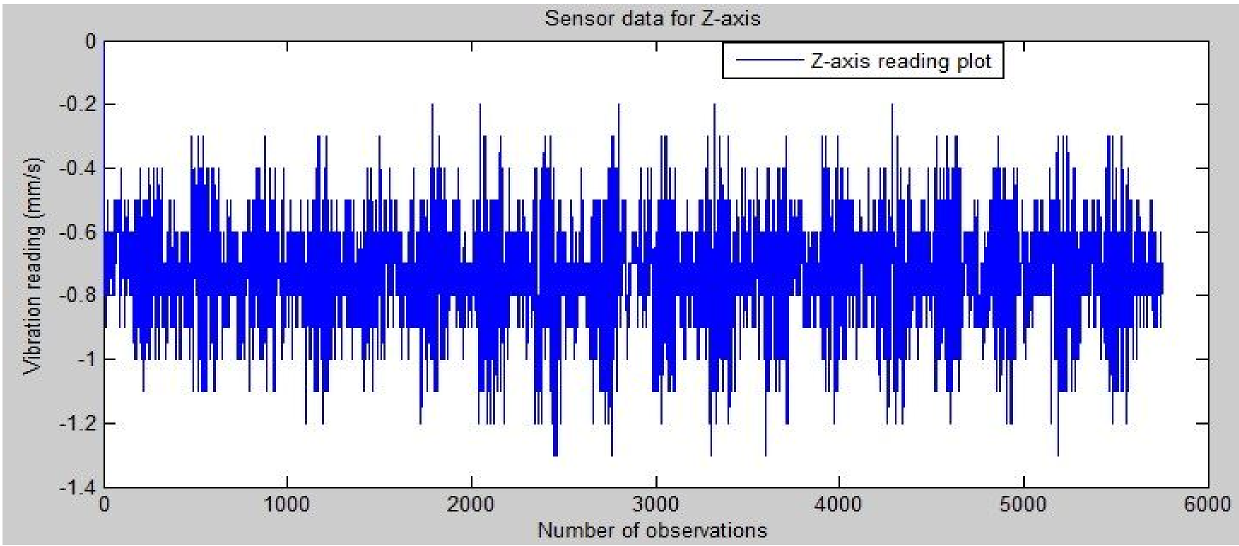


Figure 4.3: Z-axis process data variation.

We know that first 45 seconds corresponds to ON state data and then next 45 seconds to OFF state and so on alternatively. Our task is to device such method so both the states data are separated out automatically and no manual data separation is required.

Idea behind this separation of data is the changing variance in the datasets. Variance for ON state data sets and OFF state data sets vary significantly so this statistical difference can be used to carry out the separation of two states from one another.

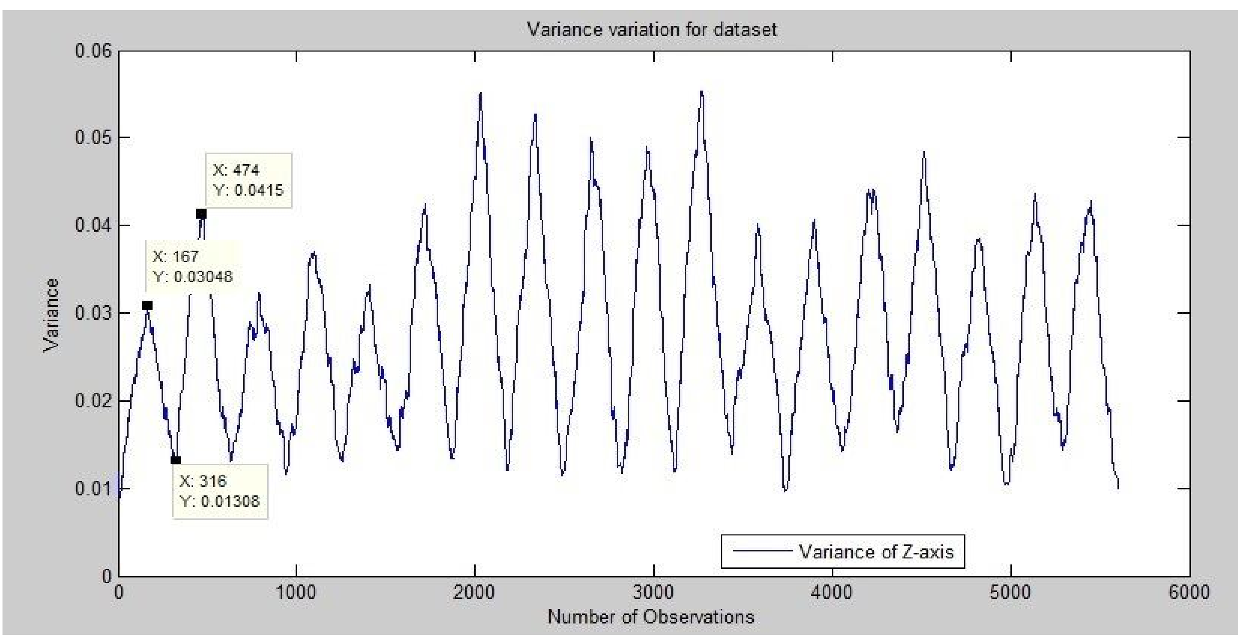


Figure 4.4: Variance plot for Z-axis with moving window 200.

As it can be seen from figure 4.4 that variance is de-trending after about every 150 elements. Since, sensors are reading 3-4 values every 1 second so in a span of 45 seconds they will provide about 150 different readings.

4.1.1.1 Calculation of Variance

For calculating the variance we considered a moving window of 150 size as our sensor will provide approximately 150 different readings in 45 seconds span and calculated the variance for the whole dataset.

After calculating the variance second, objective is to consider the slip point for separating the ON and OFF state datasets. For this purpose we consider the calculated variance in a moving window of size 200 as our sensor will provide approximately 200 different readings in 1 minute span and performed the following steps:

Consider the next 200 variances and find the maximum out of them and note the data point corresponding to that variance value.

This data point is out split points. Data elements from the last reading grouped under OFF state upto this point is our ON state data points

Keep next 150 points in the OFF state dataset.

Continue these steps until all the data points are grouped under ON or OFF state.

Now, we have separated out data sets into two different states ie, ON and OFF state. Next objective is to calculate the feature space to identify the nature of this time series data separated in this step.

4.2 Feature Generation

4.2.1 Feature Selection

The hidden information in the time series signals obtained from the pump and motor system can be analyzed by many different methods like-: time-domain feature calculation, frequency domain analysis and time-frequency methods. Averaging the time series values and then comparing then using them as a baseline data for matching the real time sensor data is another method of comparing the two time series.

In this thesis we have defined some time domain features for which the values over pump and motor system can be calculated and then these feature value can be used as a baseline for evaluating or analyzing the ongoing process state of the vacuum pump and motor system.

.

4.2.2 Identifying Features for Time Series Data

In this certain properties/ features were identified in the time domain which provides information about the varying nature of our time series data. The listed features considered are-:

- Peak Value
- Root Mean Square
- Standard Deviation
- Crest Factor
- Kurtosis Value
- Clearance Factor
- Impulse Factor
- Shape Factor

4.2.3 Feature Extraction Parameter Definitions and Notations

1. Peak Value

The maximum instantaneous value of a signal during the time interval under consideration.

$$\text{Peak value, } P_v = (1/2)\,[\max(x) - \min(x)] \qquad (4.1)$$

2. Root Mean Square

The root mean square, is the quadratic mean of a statistical measure defined as the square root of the means of the samples squared.

$$\text{RMS value, } RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x)^2} \qquad (4.2)$$

3. Standard Deviation

The Standard Deviation is a unit which is used to quantify the amount of variations or scattering of a set of data values.

$$\text{Standard deviation, } SD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x-\overline{x})^2} \quad (4.3)$$

4. Kurtosis Value

Kurtosis value is the measure of "peakedness" of the probability distribution for random real value variables. Kurtosis value defines the shape of a probability distribution.

$$\text{Kurtosis value, } K_v = \frac{\frac{1}{N}\sum_{i=1}^{N}(x-\overline{x})^4}{(RMS)^4} \quad (4.4)$$

5. Crest Factor

Crest factor is defined for a time series, which shows the peak value to the effective value. It defines how high peaks are present in the time series data.

$$\text{Crest factor, } Crf = \frac{Peakvalue}{RMS} \quad (4.5)$$

6. Clearance Factor

Clearance factor represents the relative pelleting efficiency of a given instrument at maximum speed.

$$\text{Clearance factor, } Clf = \frac{Peakvalue}{(\frac{1}{N}\sum_{i=1}^{N}\sqrt{\text{mod}(x)})^2} \quad (4.6)$$

7. Impulse Factor

Impulse factor indicates the fault in rotating machinery and it is calculated by taking ratio of peak value with the mean value of the time signal.

$$\text{Impulse factor, Imf} = \frac{Peakvalue}{\frac{1}{N}\sum_{i=1}^{N}\text{mod}(x)} \quad (4.7)$$

8. Shape Factor

Shape factor is a dimensionless quantity which defines the shape of an object or signal.

$$\text{Shape factor, Shf} = \frac{RMS}{\frac{1}{N}\sum_{i=1}^{N}\text{mod}(x)} \quad (4.8)$$

4.2.4 Calculating Feature Values

The goal of feature extraction step is to process the time series data into a sequence of ordered pair values for the multivariate classification technique. The raw data available is grouped accordingly as defined in section 4.1. In this phase, we first order the data temporally and then consolidate the feature extraction information. The details of this have been presented below.

The main idea of representing our time series data with the help of features is to identify its behavior over a certain period of time. After we have identified certain pattern in these feature values we can easily differentiate between different patterns present in the time series data and what nature or state process is residing corresponding to the correct pattern.

So, there are two important things to be considered during time series feature extraction which are listed as-:

- Determine the time length for which data is to be considered for feature extraction or, in other words moving window size for feature extraction process. As if we consider too large window size then our system might not predict the current state of the system before the failure occurs.
- After choosing the moving window size next important task is to identify if this window will be able to correctly distinguish between different feature values for different process states or, will it provide different patterns for different states or similar. See Figure 4.5 and Figure 4.6.

4.2.5 Formulating Feature Parameters

Once the feature values are computed, we create the feature table. We scan the table which is grouped by Process State and ordered by the feature parameters. Each table constitutes single state flow for a particular span of time.

After the feature extraction is completed than these parameters are evaluated via single Plots, Scatter Plots and Key Performance Indicator by univariate and multivariate analysis.

In this work we proposed, time domain features which are extracted from the raw data and then statistical analysis are used for the purpose of finding best separation features.

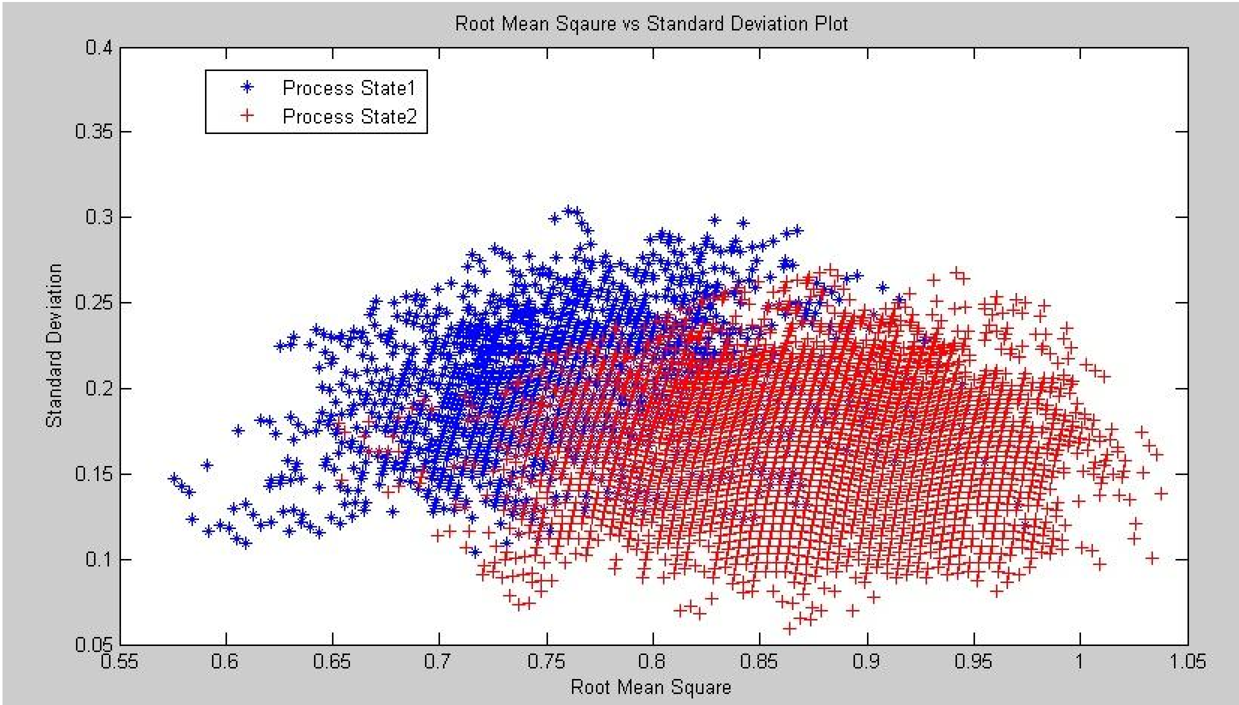


Figure 4.5: Root Mean Square vs Standard Deviation plot for Moving Window size 21.

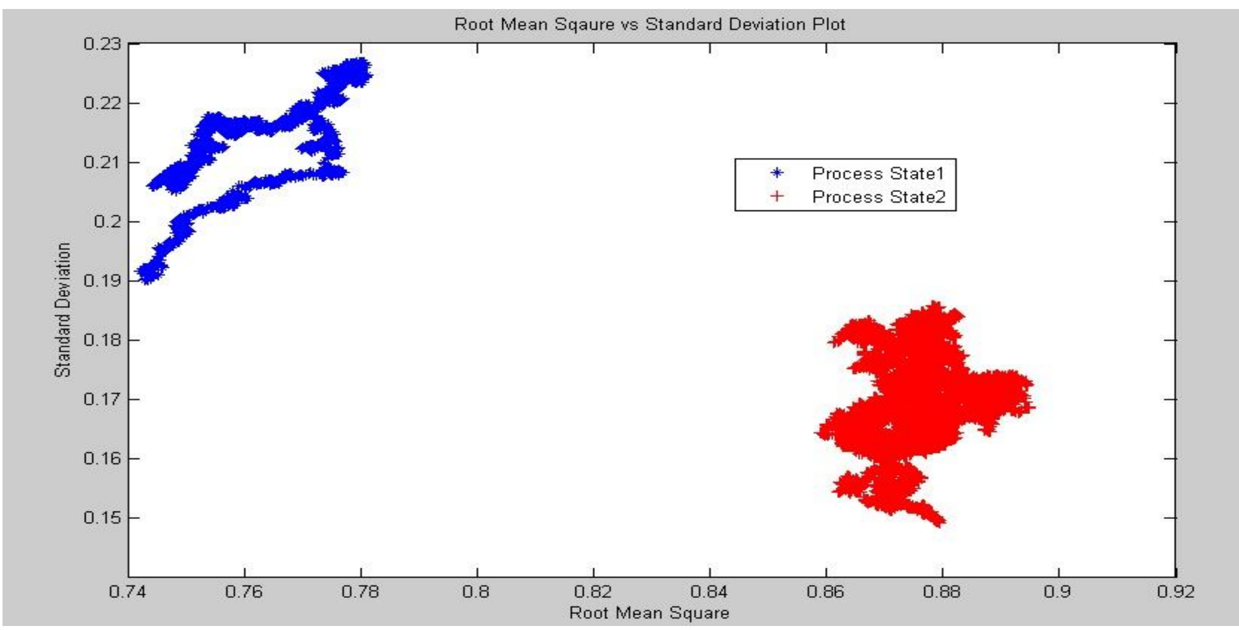


Figure 4.6: Root Mean Square vs Standard Deviation plot for Moving Window size 900.

4.3 Creating Process Data Feature Profiles

Table shows information about the feature value available after the data preprocessing and feature calculation stage. Further feature values are calculated for other process states and stored in feature table. We use this data to determine which features can be used to separate out different process states.

Data points	Root Mean Square	Standard Deviation	Peak value	Crest factor	Kurtosis value	Clearance factor	Impulse factor	Shape factor
1-1350	**0.8718**	**0.1888**	**0.550**	**0.6309**	**0.0062**	**0.6545**	**0.6462**	**1.0243**
2-1351	**0.8722**	**0.1887**	**0.550**	**0.6306**	**0.0062**	**0.6542**	**0.6458**	**1.0242**
3-1352	**0.8725**	**0.1887**	**0.550**	**0.6304**	**0.0062**	**0.6540**	**0.6540**	**1.0242**
4-1353	**0.8725**	**0.1884**	**0.550**	**0.6304**	**0.0062**	**0.6543**	**0.6540**	**1.0242**
5-1354	**0.8720**	**0.1844**	**0.550**	**0.6308**	**0.0062**	**0.6549**	**0.6543**	**1.0242**
6-1355	**0.8712**	**0.1891**	**0.550**	**0.6313**	**0.0062**	**0.6553**	**0.6549**	**1.0242**

Table 4.2: Sample feature values for moving window size 1350.

4.3.1 Calculating Feature Signatures

We introduce the notion of feature signature corresponding to each process state data in the process network. Feature signature has been defined as the sequence of feature values based on the data provided by the sensors. The value in the feature signature represents the profile of the process corresponding to the data obtained by sensors. In general, we can use these signatures to solve the questions related to estimate the current state of the ongoing process. We get a upper and lower bound for the feature values which are calculated for the ongoing process.

We perform the following set of operations for each dataset to derive process state wise feature profiles for the system:

- Separate out the ON and OFF state for the raw data obtained from the sensors.
- Define a moving window size for feature value calculation process.
- Calculate all the eight feature values for all the process data available.
- Store the obtained feature values in the form of a table.

- Repeat step 2 to step 4 for other defined moving window sizes.

At the end of this computation we have the process state wise feature profiles for all the datasets obtained by the sensor data. Next, we use statistical analysis like Scatter Plots to determine the best features combination for classifying different states obtained during the process flow.

4.4 Feature Selection for Classification

After calculating the feature parameter values another important task is of finding the suitable feature parameters which can be used to classify are different process states with minimum rate of misclassification. For this purpose we use a manual approach of statistical analysis with the help of Line Plots for univariate analysis and Scatter Plots for multivariate analysis.

- **Line Plots:** Line plot describes the occurrence of a data along the instance of occurrence.
- **Scatter Plots:** Scatter plots defines two feature plot against each other on the co-ordinate axis i.e., variation of one time feature with respect to other time feature

The minimum the overlap between the feature data of different state the better will be the classification. Line Plot and Scatter Plot are shown in figure 4.7 and figure 4.8 respectively.

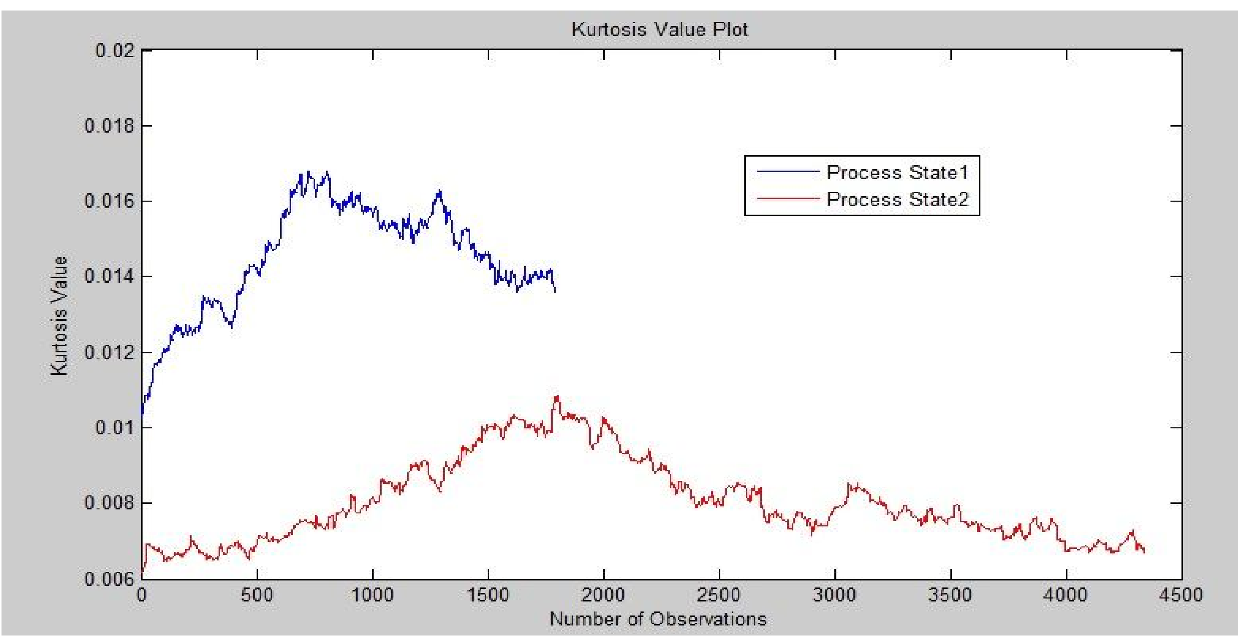


Figure 4.7: Line plot representation

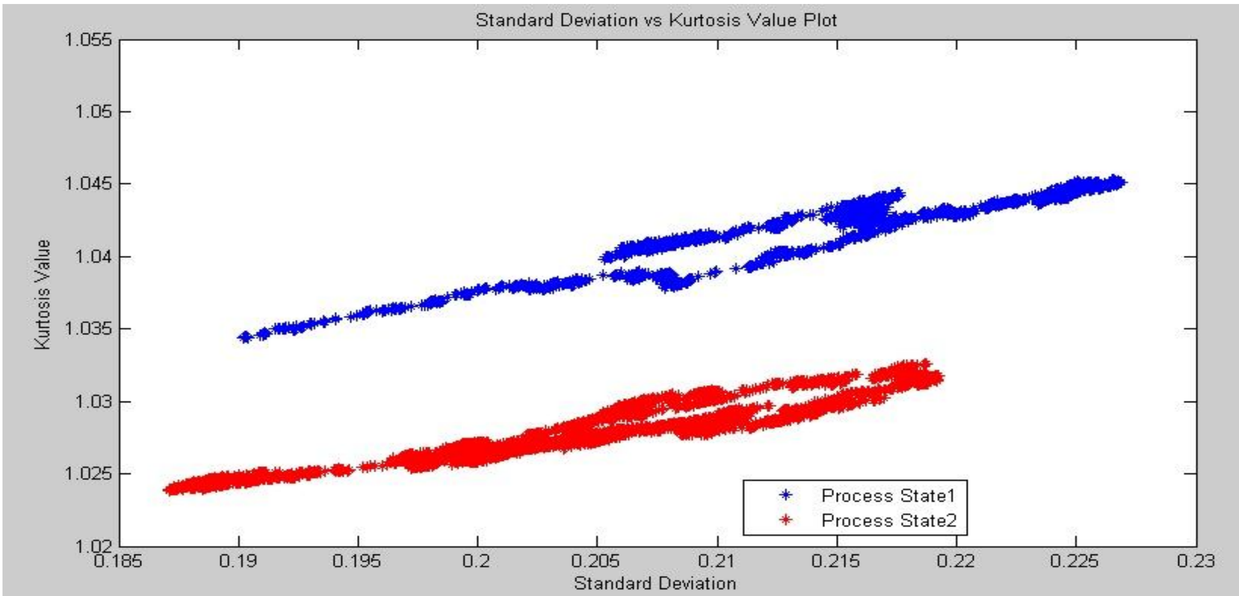


Figure 4.8: Scatter plot representation.

The better the separation the of feature values for different process states the better the classification. But this process of statistical analysis is a manual process of identifying potential feature parameters for classification. In next chapter we define another concept of *Key Performance Indicator* which requires no human effort for identifying the potential feature parameters for classifying different process states.

Chapter 5

Algorithms and Visualization Methods

Our work in thesis focus upon determining some key features from the time series data and then using these feature classify different operating conditions present in pump and motor system using these features. Classification or mapping is a method of tagging the entities with some class label to which they are closely related with. Since, classification falls under supervised learning category we need training data with contains some variables assigned with values and corresponding to each variable vector a class tag to which they belong. This training data is then used to model the decision parameters based on the learning of classification algorithms from this training data. Here, we describe some already existing classification algorithms which can be used to design learning and decision model from the training data and then we can test these learning by classifying the variable vectors of testing data sets and determine the accuracy of the learning done by each classification algorithm. The accuracy of each algorithm is defined based on the key performance indicator which is the ration of the miss-classified testing samples out of the total testing data samples.

5.1 Classification Algorithms

5.1.1 Naïve Bayes Classification

In naïve bayes method of classification learning is based on the probabilities defined for the membership of each variable vector associated to a particular class. It is assumed that each attribute independently classify each tuple during the learning process.

Let us define a data tuple by D, it contains the measurements for a set of attributes associated to some class and is known as "evidence". Let T defines a hypothesis that a particular tuple belongs to certain class. For learning process we have to determine P(T/D), the chance that a hypothesis T is true for data tuple D. It defines the membership for a tuple being associated to a particular class label. P(T/D) is also known as posteriori probability, of T over D. P(T) is prior probability which defines that a data tuple D belongs to a particular class label irrespective of any other information. P(T/D), posterior probability is more defined by the data tuple D then P(T) which does not depend on D. In the same way P(D/T) defines as the posterior probability of data tuple D over T. P(D) is prior probability for D.

The formula for calculating these memberships required in naïve bayes is given as-:

$$P(T/D) = \frac{P(D/T)P(T)}{P(D)} \tag{5.1}$$

Some basic assumptions in naïve bayes classification are-:

1. $P(l_j)$ can be obtained by the class label frequency in training data set.
2. Probability of combination of attributes belonging to particular class is equal to the combined probability of each individual attribute $P(d_j/l_j)$ belonging to a particular class.

Algorithm 5.1 Naïve Bayes (F, I, C, N)

Input: F : frequency tables

I : total number of samples

C : total number of class labels

N : number of samples for each class

Output: Most likely hypothesis for a test case

```
function update(class, train) {
 I++
 if    (++N[class]==1)
then  C++
end if
for <attributes, measure> in train
do  if   (measure != "?")
      then F[class, attribute, range]++
    end if
done   }
function classify(test) {
m=2
l=1;
like = -100000
for (T in N)
do  priorprob = (N[T]+l)/(I+(l*C))
      temp  = log(priorpob)
```

```
for <attributes, measures> in attributes
do  if  (value != "?")
        then increase = F[T, attributes, measures]+(m*priorprob))/(N[T]+m)
              temp += log(increase)
    end if
 done
 if  (temp >= like)
  then like = temp
        class = T
end if
done
return class    }
```

Let us assume our training examples are defined for k attributes, with n values contained in them. Then we have total number of kn+1 parameters which take 2kn+2 counts to complete the learning process with one count for every attribute value present in the training data set. So, if there are m different examples total training time will be given by O(mk) which is independent of n total number of values. This time complexity obtained is optimized for all the learning algorithms which learns considering all the attribute values present in training example.

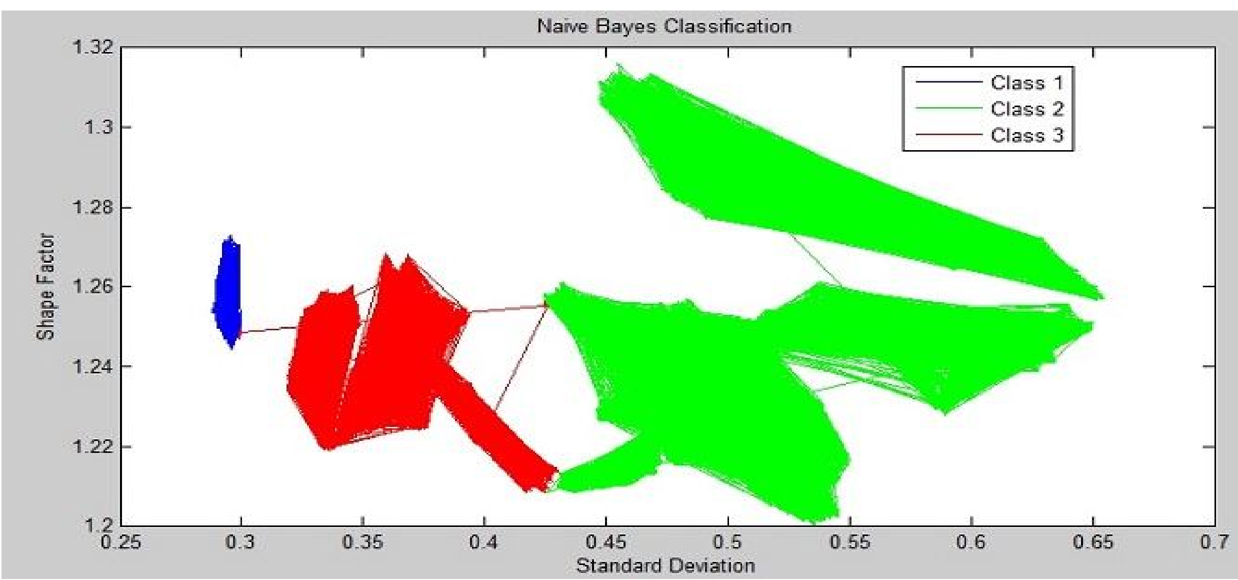


Figure 5.1: Classification done by Naïve Bayes.

5.1.2 Nearest-Neighbor Classification

Nearest-Neighbor classifications groups similar objects together on the basis of the distance between the objects to be grouped and the center or representative object of that group. The total number of groups being equal to the different number of class labels present in the data set. The number of different groups created by this method is always equal to less than the total number of data samples present in the training data. Each group or partition created is known as cluster.

Nearest-Neighbor learning algorithm takes two input parameters, total number of different groups to be created and data sample to be grouped. The grouping or forming clusters is done by reducing the within group distance and increasing the between group difference. The between group difference is calculated by taking distance difference between the group mean or their representative element.

Nearest-Neighbor classification algorithm works as-: First task is to select c different objects randomly here c belongs to different number of class labels present in the data set with each object representing their group or mean value for the group. Than each of the remaining n-c objects in the training sample are grouped in these c different groups based on distance between them and group representative. The object which is closest to a particular group representative is assigned to that particular group. After adding the object to the group new mean or representative for the group is defined. This process repeats until the stopping criterion is matched. The mean for the group is defined by-:

$$\text{Mean } m_i = \frac{c_{n1} + c_{n2} + \ldots\ldots + c_{nk}}{k} \tag{5.2}$$

where m_i is the mean of the cluster and c_{nk} defines the group element where n is the group number and k is the element in that group.

Algorithm 5.2 Nearest-Neighbor (D, k, m)

Input: D : dataset to be clustered

k : number of clusters to be formed

m : maximum number of iterations

Output: C={c_1, c_2,….,c_k} (cluster means or representative)

L={l(e) | e = 1,2,….,n} (class labels for clusters)

1. **for each** ci $\in$C do

2. $c_i \leftarrow e_j \in E$ (random selection)
3. **end**
4. **for each** $e_i \in E$ do
5. $l(e_i) \leftarrow$ calmindistance(e_i,c_j) $j \in \{1,\ldots,k\}$
6. **end**
7. alteration $\leftarrow$ false;
8. iter = 0;
9. **repeat**
10. **for each** $c_i \in C$ do
11. refreshCluster(c_i);
12. **end**
13. **for each** $e_i \in E$ do
14. minDist $\leftarrow$ calmindistance(e_i, c_j) $j \in \{1,\ldots.,k\}$
15. **if** minDist $\neq l(e_i)$ then
16. $l(e_i) \leftarrow$ minDist
17. alteration $\leftarrow$ true;
18. **end**
19. **end**
20. iter ++
21. **until** alteration =true and iter $\leq$m;

The computational complexity for the learning method is given as-:

1. In Euclidean space d it is NP-hard.
2. If total groups and dimensions of the training data are fixed than the groups can be formed in $O(k^{dn+1} \log k)$ time.
3. If number of different groups to be created are not specified than it again NP-hard.

After all the objects in the training data set are grouped with similar objects together we can classify our test data sets based on the mean value or representative of each group, the minimum the distance with the specified group the data set belongs to that particular class label or group.

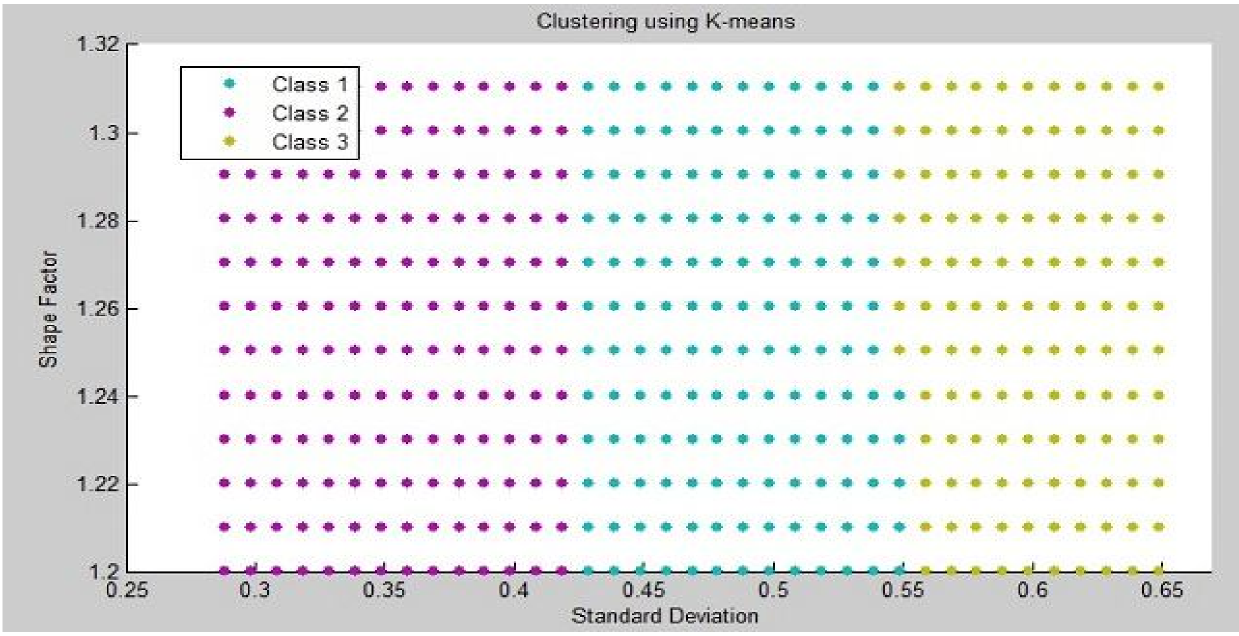


Figure 5.2: Clusters generated by k-means for classification.

5.1.3 Classification and Regression Tree (CART)

The learning model obtained by classification and regression trees contains nodes and edges which define a tree structure with nodes representing the split points and the attribute which is considered for splitting and edges defines the measurement limits for those attributes. This approach is known as decision tree learning. In this tree learning method external nodes defines the class labels and internal nodes define the split points i.e., attribute for decision making.

Classification tress contains discrete class labels at the external nodes but regression trees define continuous class labels in the leaf nodes. This method or technique also falls under supervised learning category.

The learning tree obtained by this method contains rules defined on data variables from the training data set-:

1. Different variable values are selected for deciding the split point in the tree and these splits forms the basic rules for classification.
2. This splitting of nodes is a repetitive procedure and is applied all over the tree to obtain rules for classification.
3. This process finding the split point stops when no further information can be obtained from splitting the nodes or, any predefined stopping condition is met.

Every branch of the decision tree contains leaf node and each entry of the training data is classified into certain group and only one terminal node and every leaf node is reached by different rules.

The measure to decide the split point in the learning tree in this classification method is given by Gini index which defines the measure of miss-classifications if that particular attribute is selected for splitting for which Gini index is being calculated. So, lower the Gini index value for a attribute better it is for split point.

$$G(D) = 1 - \sum_{i=1}^{m} \left(\frac{x_i}{n} \right)^2 \qquad (5.3)$$

where G(D) is the Gini index value and $(x_i/n)^2$ is the total probability for a data tuple D that it belongs to a particular class label C_i. In CART each attribute in divided by binary split. For the attributes having continuous values every split point should be taken into consideration. In continuous values split point is taken by calculating mid-point of every sorted adjacent values of the attributes. The reduction in Gini index measure at each split point is given by-:

$$\Delta G(A) = G(D) - G_A(D) \qquad (5.4)$$

The attribute which provides greater reduction in index value is chosen as the split point. Thus if a split point is made at A than the rules will be governed as-: D_1 if in D satisfying $A \leq split\_point$, and D_2 if in D satisfying A > split_points, where D_1 and D_2 are the set of tuples.

Algorithm 5.3 CART (D)

Input: an attribute-valued dataset D

Output: Decision Tree

1. Tree={}
2. **if** D is "pure" OR any stopping criteria is met **then**
3. STOP
4. **end if**
5. **for all** attribute $a \in D$ **do**
6. Measure Gini value for splitting at point a
7. **end for**
8. a_{best} = attribute with least Gini measure

9. Tree = Create a decision node that tests a_{best} in the root.
10. D_v = Induced sub-datasets from D based on a_{best}
11. **for all** D_v do
12. $Tree_v$ = CART(D_v)
13.   Attach $Tree_v$ to the matching branch of the Tree
14. **end for**
15. **return** Tree

Time complexity for CART is defined as-: If we have k different training samples in d dimensional data, and we want to create learning tree by CART based on the Gini index, then the calculation of Gini measure will take O(k) + (k-1) O(d) for k-1 split points. The training data must be sorted in order to create the tree and it will take O(klog k) time for each variable in d-dimension at the root node. Further splitting the node at level 1 has the time complexity given by O(d k/2 log(k/2)) because there are two nodes being created at every level complexity is O(dklog(k/2)). For creating the full decision tree total average time complexity is $O(dk(\log k)^2)$.

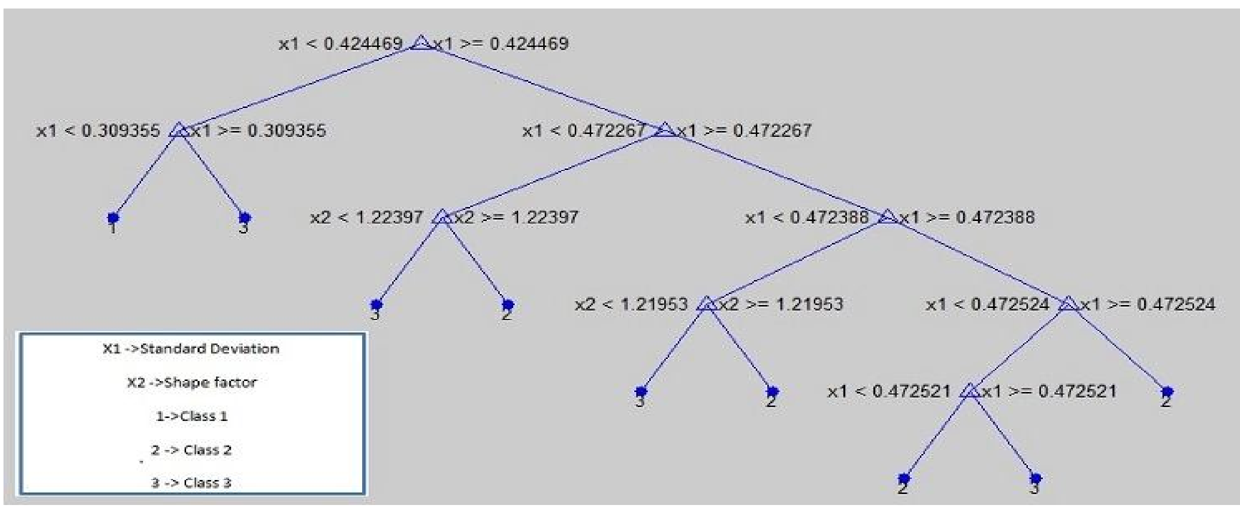


Figure5.3: Classification Tree generated by CART for classification.

5.1.4 Discriminant Analysis

For dimensionality reduction of the data we can map it onto a line. If the training samples in this multi-dimensional data are well-separated and they can grouped into separate clusters in d-space, then projecting them on a line will result in a mixture of samples which will be difficult to identify, in this case our objective will be to find an orientation for separating these projected points which is the objective of discriminant analysis.

Let us say we have c different classes in our training data set so discriminant function will find c-1 different functions to classify the class categories. Here we are representing a d-dimensional space in (c-1) dimensional space and it is assumed that $d \geq c$. Samples $x_1, \ldots\ldots, x_n$, n_1 in the subset D_1 labelled w_1 and n_2 in the subset D_2 labelled w_2. The with-in class scatter matrix is given by-:

$$S_w = \sum_{i=1}^{c} S_i \tag{5.5}$$

where,

$$S_B w = \lambda S_w w \tag{5.6}$$

$$S_w^{-1} S_B w = \lambda w \tag{5.7}$$

$$W = S_w^{-1}(m_1 - m_2) \tag{5.8}$$

and

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x \tag{5.9}$$

Suppose that we define a total vector m and a total scatter matrix S_T given by-:

$$m = \frac{1}{n} \sum_{i=1}^{c} n_i m_i \tag{5.10}$$

and

$$S_T = \sum_{x} (x - m)(x - m)^t \tag{5.11}$$

then

$$S_T = S_w + \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^t \tag{5.12}$$

The next matrix will be between-class scatter matrix and total scatter is given by the summing the within and between-class scatter matrix.

$$S_B = \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^t \tag{5.13}$$

and

$$S_T = S_w + S_B \tag{5.14}$$

Total c-1 discriminant functions are obtained while projecting d-dimensional data to c dimensional space

$$y_i = w_i^t x \quad i=1,\ldots..,c\text{-}1. \tag{5.15}$$

If the y_i are viewed as components of a vector y and the weight vectors w_i are viewed as the columns of a d-by-(c-1) matrix W, then the projection can be written as a single matrix equation

$$y = W^t x \tag{5.16}$$

The samples $x_1,\ldots..,x_n$ project to the samples $y_1,\ldots.,y_n$, which can be described by their own mean vectors and scatter matrices. Thus,

$$\tilde{m}_i = \frac{1}{n_i}\sum_{y \in y_i} y \tag{5.17}$$

$$\tilde{m} = \frac{1}{n}\sum_{i=1}^{c} n_i \tilde{m}_i \tag{5.18}$$

$$\tilde{S}_w = \sum_{i=1}^{c}\sum_{y \in y_i}(y - \tilde{m}_i)(y - \tilde{m}_i)^t \tag{5.19}$$

and

$$\tilde{S}_B = \sum_{i=1}^{c} n_i(\tilde{m}_i - \tilde{m})(\tilde{m}_i - \tilde{m})^t \tag{5.20}$$

then,

$$\tilde{S}_w = W^t S_w W \tag{5.21}$$

$$\tilde{S}_B = W^t S_B W \tag{5.22}$$

Now, objective is to find transformation matrix W which maximizes the ratio of these two scatter matrix. Determinants of the scatter matrices can be used as a scalar measure and it is defined as the product of eigen values i.e., the product of variances in the principal direction. Then we obtain the equation-:

$$J(W) = \frac{|\tilde{S}_B|}{|\tilde{S}_w|} = \frac{|W^t S_B W|}{|W^t S_w W|} \tag{5.23}$$

If we have only two different classes present in our training data then the above equation can be written as-:

$$S_B w = \lambda S_w w \tag{5.24}$$

and

$$S_w^{-1} S_B w = \lambda w \tag{5.25}$$

then, transformation matrix will be given by-:

$$W = S_w^{-1}(m_1 - m_2) \tag{5.26}$$

Algorithm 5.4 Linear Discriminant Analysis $(D = \{(x_i^T, y_i)\}_{i=1}^n)$

Input: dataset defining attributes and class labels

Output: dominant eigenvector

1. $\mathbf{D_i}$ = { x_j^T | $y_j = c_i$, j=1,….,n}, i = 1, 2 // class-specific subset
2. μ_i = mean(D_i), i=1, 2 // class mean
3. $\mathbf{B} = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ // between class scatter matrix
4. $Z_i = D_i - 1_{n_i}\mu_i^T$, i=1, 2 // center class matrices
5. $\mathbf{S_i} = Z_i^T Z_i$, i=1, 2 // class scatter matrices
6. $\mathbf{S} = S_1 + S_2$ // within class scatter matrix
7. λ_1, **w** = eigen(S^{-1} B) // compute dominant eigenvector

Computational or time complexity of discriminant analysis is-: For each c, there are $N_c d$ additions and 1 division. Thus, in total there are Nd+C operations. There are $N(d+d^2)$, the first d is for (x_j-m_c), and second d_2 is for $(x_j-m_c)(x_j-m_c)^T$. There are $C(d+d^2+d^2)$, the first d for (m_c-m), the second d^2 is for $(m_c-m)(m_c-m)^T$, and the third d_2 is for the multiplication between N_c and the matrix. There are $d^3+d^3+d^3$, the first d^3 is for S_w^{-1} and the second d^3 is for the multiplication between S_w^{-1} and S_b, and the third d^3 is for the eigen-decomposition. In general, we have N>d>C, thus the dominated term is Nd^2. Thus, computational complexity is $O(Nd^2)$.

5.2 Comparison of Classification Algorithms

For our pump and motor system data we classified our data sets using all the defined algorithms in section 5.1, but there was a significant difference between the classification results obtained by these algorithms. The efficiency of each algorithm is determined by the key performance indicator or miss-classification rate obtained during testing the classification rules generated by each algorithm. The difference between the classification rate of these algorithms differ significantly in some cases because of the feature value distribution.

For this purpose we choose a data set of 36633 data values matrix with 2 feature values Standard Deviation and Shape factor. Out of these 30% of data values were used of testing and remaining 70% data values for training.

Table 5.1 contains information about the time complexity, miss-classification rate and total execution time (training and testing) of the classification algorithms defined in section 5.1 for 25643 training data values and 10990 testing data values and computational complexity for each classification algorithm.

Figure5.4 and Figure5.5 shows a comparison of total execution time and total miss-classification rate of the above defined classification algorithm.

Table 5.1: Parameters comparison of classification algorithms.

Classification Algorithm	**Time Complexity (Big-oh notation)**	**Miss-classification rate (Key performance indicator)**	**Execution Time (in seconds)**
Discriminant Analysis	$O(N*d^2)$	827	0.0520
Naïve Bayes	$O(m*n)$	508	0.0482
CART	$O(d*n(\log n)^2)$	1	0.1072
Nearest Neighbor	$O(n^{dk+1} \log n)$	1	0.0866

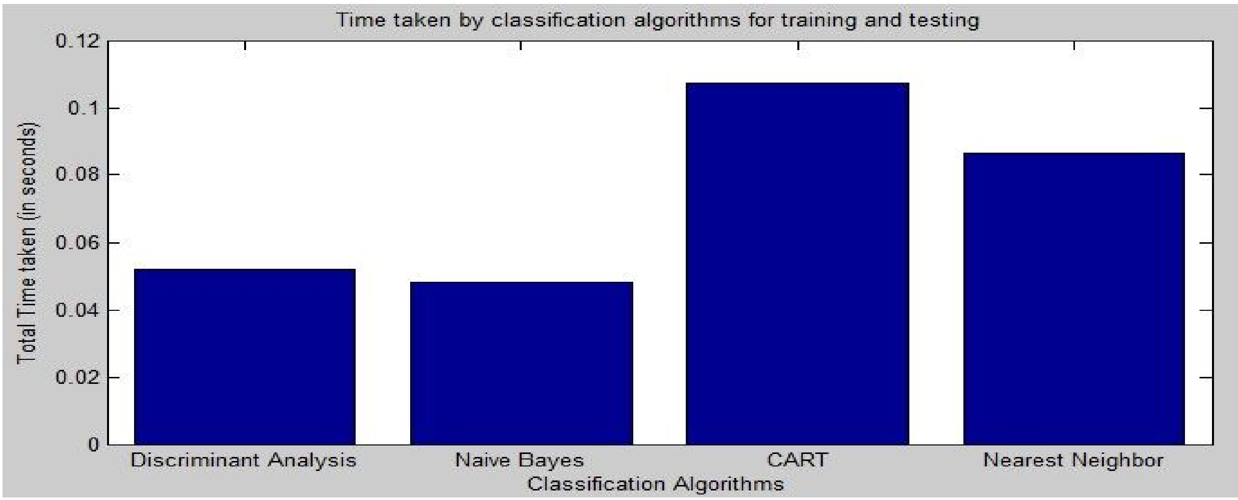


Figure 5.4: Execution Time (Training and Testing) for Classification Algorithms

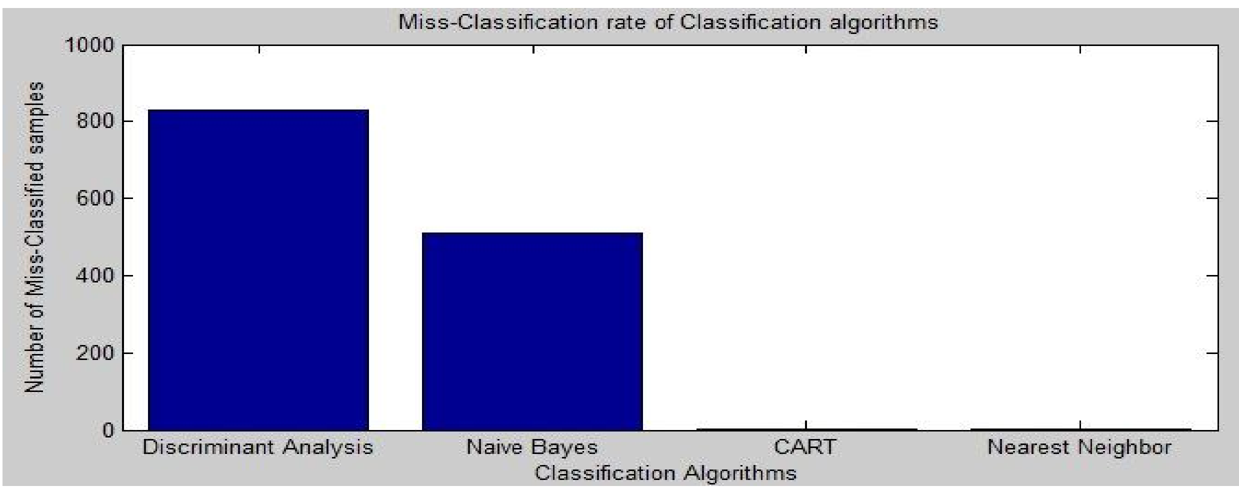


Figure 5.5: Total number of miss-classifications done by classification algorithms.

This significant different between miss-classification rate of Discriminant Analysis, Naïve Bayes and CART, Nearest Neighbor is coming because of non-uniform scatter of the data plots or overlapping regions generated by different classes.

As can be seen in Figure 5.6 there is an overlapping region between Class 2 and Class 3 marked by 'green' and 'red' respectively. So, when we trained and test our data set using discriminant analysis it will first find a function which separate class 1 with class 2 and class 3 combined and then another function which separate class 3 with class 1 and class 2 combined. So because of these overlapping regions there is a large miss-classification rate using discriminant analysis.

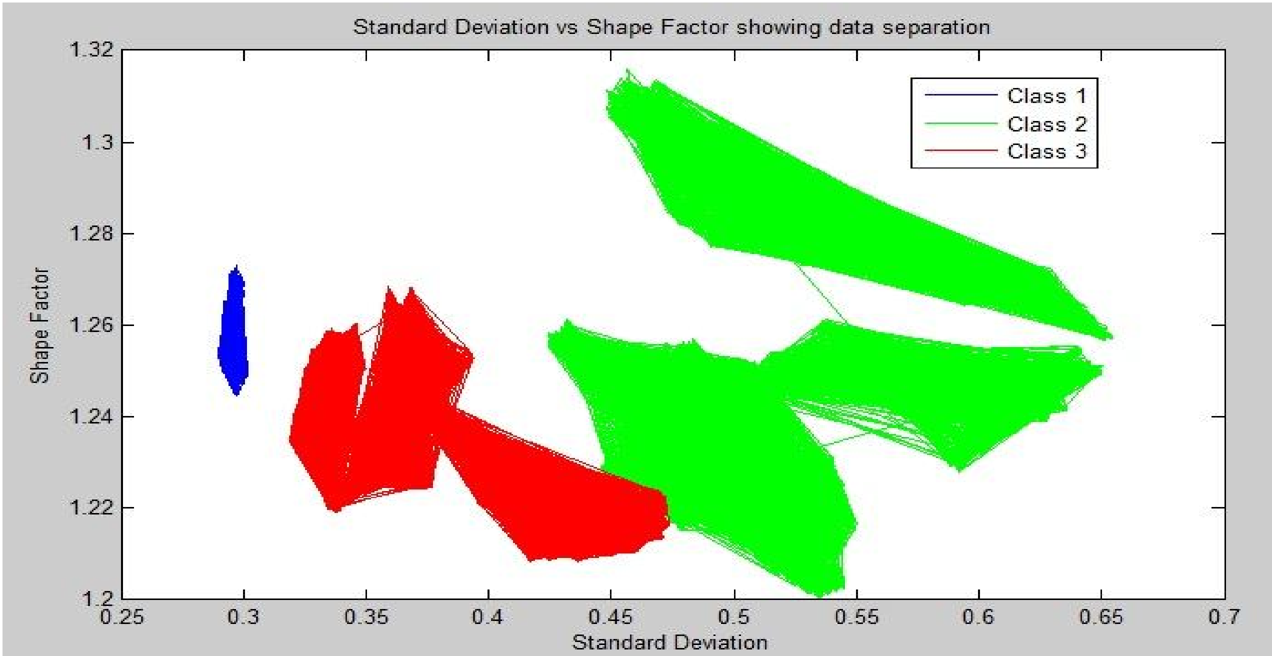


Figure 5.6: Scatter Plot for Standard deviation and Shape Factor

Similarly, in the case of Naïve Bayes classification because of this overlapping region between class 2 and class 3 there were a total of 508 miss-classifications.

But when we consider the decision tree classification algorithms i.e., CART it has miss-classified only 1 data sample. This is because the algorithm treated all the data values as a unique entity instead of considering them as a single group and then by divide-and-conquer method using exhaustive search classified the data sample into respective classes. So, the effect of overlapping regions was eliminated.

Similarly, like decision tree Nearest Neighbor classification algorithm also considered each data set individually while creating the clusters so it also miss-classified only 1 sample.

From the above discussion we can see that CART, Nearest-Neighbor classification algorithm are best suited for this system considering the miss-classification rate.

Considering the computational or time complexity of all the above classification algorithms we can say that Discriminant analysis and Naïve Bayes have better run time than CART and Nearest Neighbor and comparing with execution time which is almost double but still the later three algorithms are fast enough. But these differences in computational complexity is overcome by a negligible miss-classification rate of CART and Nearest Neighbor compared to Discriminant analysis and Naïve Bayes.

We can summarize that for this particular type of data set which has many overlapping regions decision trees and clustering methods of classification are best suited for them then compared to the methods of calculating maximum-likelihood or prior probabilities for the data set being in a particular region and they are also time effective.

Discriminant analysis and Naïve Bayes will overpower decision trees and clustering for a dataset which has non-overlapping boundaries between different class labels as they take nearly 50% of the execution time then that taken by CART and Nearest-Neighbor.

5.3 Visualization Techniques

Visualization and visual analysis play important role in exploring, analyzing and presenting such large time series data. Representing spatio-temporal and multi-variate data from different data sources, from multiple runs requires some valid technique. The heterogeneity of the data presents new chances as well as technical difficulties for visualization research. Visualization and interaction methods are often combined with computational analysis. In this work we present some approaches to represent our time series data so, operator can get a better understanding of the ongoing process.

Since, our work has multi-variate data for different individual process over different time. So, these different processes should be clearly visible to the operator so different conditions occurring in the machine state can be differentiated easily. So, the best method or approach for this purpose is multiple-tone coloring which is an example of integrated overview and detail technique. By showing the time series data as a combination of colors, the actual state of the machine can be more easily and precisely determined. With each different operating condition operating with-in the machine operator can easily and accurately determine different fault occurrences.

Other method or approach which can be used is by scatter plots. Each time series data has some time domain feature related to them and these time feature data will vary with-in certain range for a particular operating condition of the vacuum system. By defining the scatter of the time series feature over time will help identify the operator to determine operating state changing with-in the system.

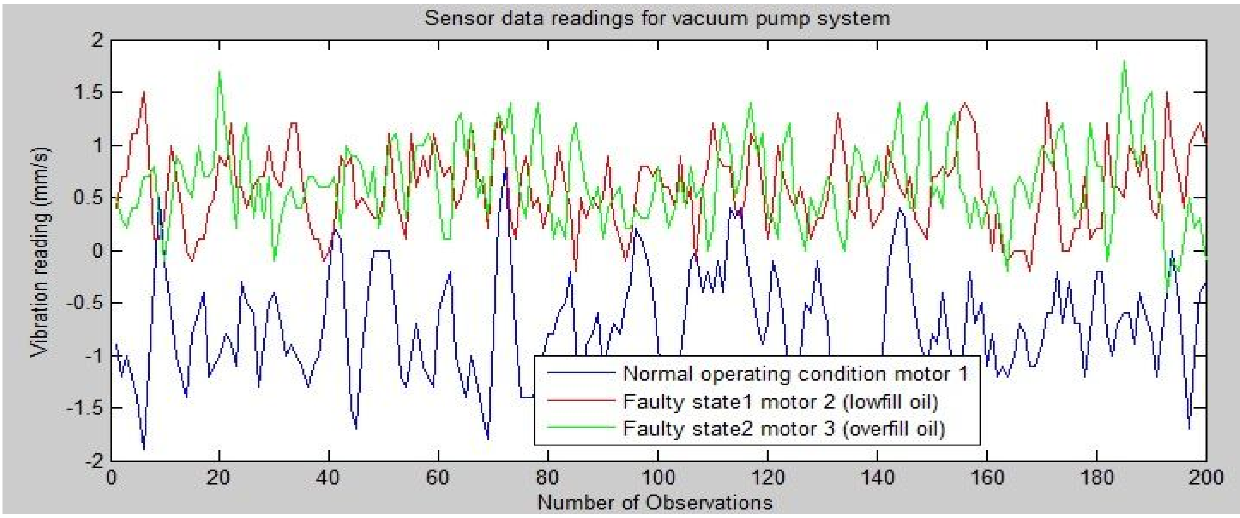


Figure5.7: Three-tone coloring for different vacuum motors showing their current state.

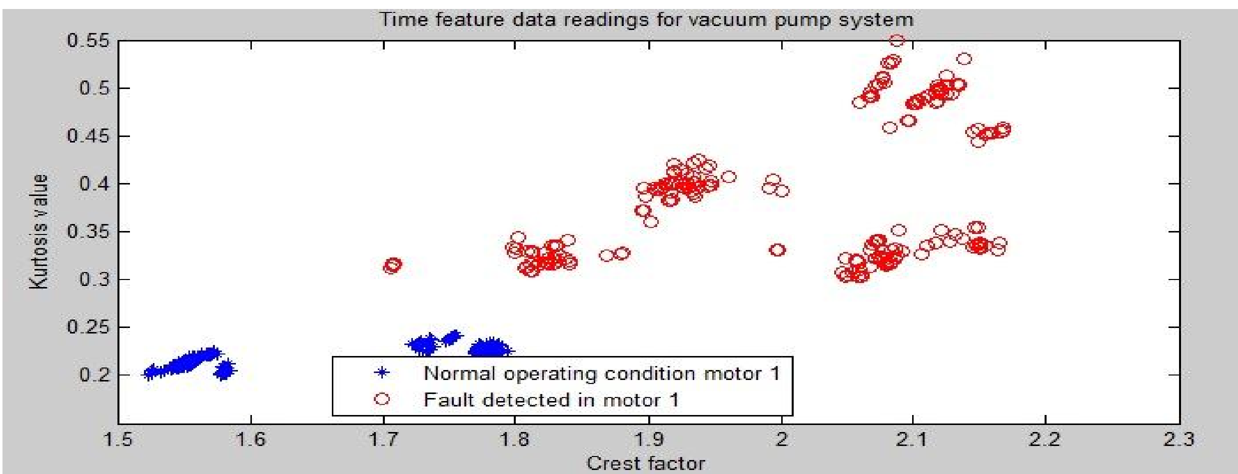


Figure5.8: Scatter plot showing deviation of crest factor and kurtosis value from normal operating condition.

Additionally, there are many other visualization approaches which can be deployed like brushing, zooming, panning or view reconfiguration which will help the machine operator in better visualization the machine process.

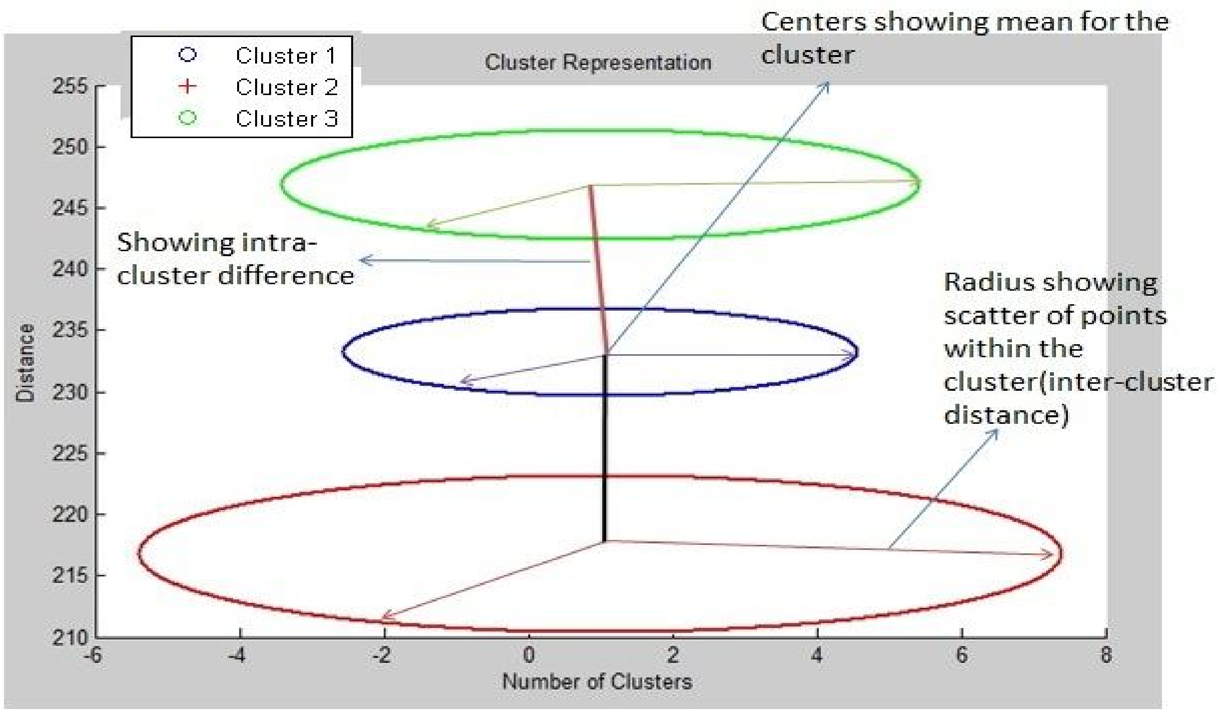


Figure5.9: Cluster representation of the dataset features.

CHAPTER 6

Experiments and Results

In this chapter, we present the experimental steps and results of the proposed work in the thesis. We have implemented different classification algorithms to predict the working state of our pump and motor system and tested them over the data sets of the system. We present the data pre-processing, feature selection, training and testing steps for classification to determine the best combination of features and algorithms.

6.1 Data and Setup

For the purpose of experimentation, the data for pump and motor system was obtained from ABB GISL, Bangalore. The available data sets contains the sensor data obtained from pump and motor system for baseline condition (normal operating condition) and faulty data (abnormal operating conditions) defining the time stamp at which the data was recorded and the corresponding values of the sensors. In order to prepare the data for classification of operating state of pump and motor system, we use the steps and techniques as discussed in Chapter 4.

6.2 Data Pre-Processing

6.2.1 Separation of raw data for feature calculation

In this section, we extract the data values for different operating states of pump and motor system. As our data set contains two different readings for ON and OFF state of pump and motor system. We need to separate these ON and OFF state values for extracting the feature values for our time series data. On and OFF state is obtained at an interval of every 45 seconds alternatively by the pump and motor system. There are time stamps present in our data set by which we can separate these two different states. But instead of using these time stamps we used the changing variance of our time series data to create to different sets of ON and OFF state data to be more accurate.

Every second sensors provided with 3 or 4 readings for the pump and motor system which makes it approximately 200 different values every minute. So every 45 seconds for each and OFF state we will have approximately 150 data values. Analyzing our data sets for three different axis i.e., x-axis, y-axis and z-axis we found that there is a pattern followed by our time series data for the z-axis data set. Refer to Figure 4.1, 4.2 and 4.3 to see pattern formation for x, y and z-axis.

In z-axis we can see alternate lower and higher fluctuations which denote ON and OFF state of our pump and motor system respectively. For calculating the variance we considered a moving window of 150 data points as our sensor will provide approximately 150 different readings in 45 seconds span and calculated the variance for the whole dataset.

After we have calculated the moving variance for our data set, we know that the variance for OFF state of pump and motor system is higher than that of ON state. So for differentiating between the two states for the purpose of splitting our data we considered another moving window of 200 data points (for variances) as our sensor will provide approximately 200 different readings in 1 minute span. We then find the point which has maximum variance out of these 200 data points. This data point with maximum variance is the split point for our ON and OFF state data. We separate out these data points starting from the first point of our moving window up to the point with maximum variance and put them under ON state data and next 150 data points in the OFF state data. Likewise, we repeat this step until all the data points are grouped under ON and OFF state data points. This separation of ON and OFF state data sets based on the variance approximately after every 150 data points can be seen in Figure 4.4.

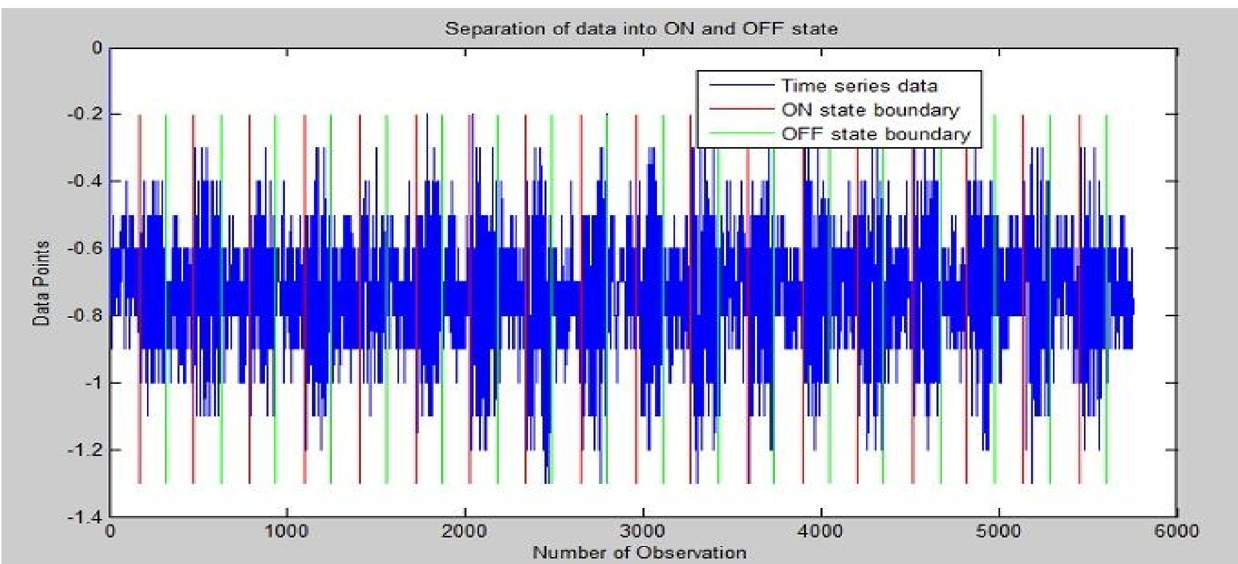


Figure 6.1: Separation of ON and OFF state data by variance difference

Similarly, we separated ON and OFF data states for all the data sets of baseline data and faulty data. Next step is to calculate optimal time series features with the help of which we can define the changing nature of our time series data to correctly determine the current working state and conditions of pump and motor system.

6.2.2 Time Series feature Calculation

The primary purpose of feature calculation step is to process the time series data into a sequence of ordered pair values for the multivariate classification technique. The raw data available is grouped accordingly as defined in section 6.2.1. In this phase, we first order the data temporally and then consolidate the feature extraction information. The details of this have been presented below.

The idea of representing time series data with the help of features is to identify its behavior over a certain period of time. After we have identified certain pattern in these feature values we can easily differentiate between different patterns present in the time series data and different operating state and conditions for our pump and motor system.

The time feature values which we considered for purpose of classification of pump and motor system operating state are-: Peak Value, Root Mean Square, Standard Deviation, Crest Factor, Kurtosis Value, Clearance Factor, Impulse Factor, Shape Factor. We will calculate the time series values for all these features and then select which feature values best separates different classes present in our data.

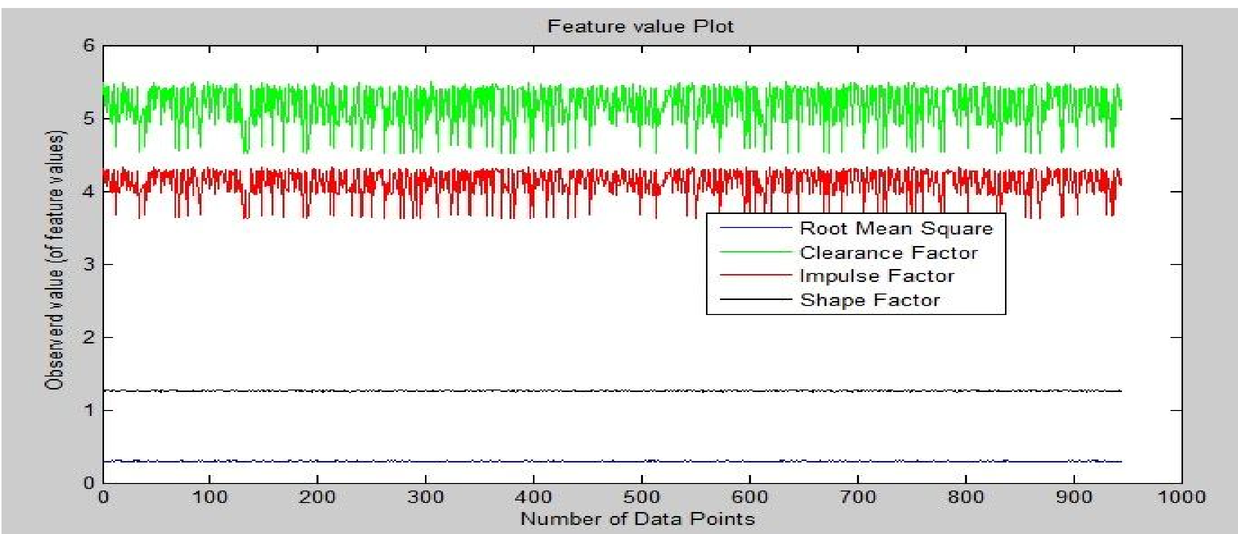


Figure 6.2 Showing feature values variation for moving window size 1350.

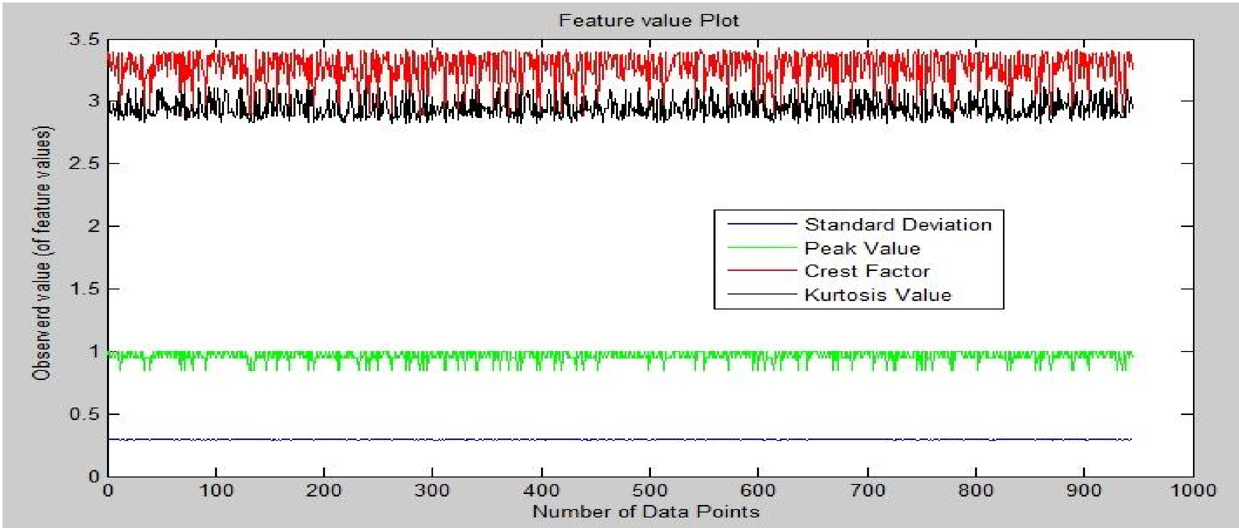


Figure 6.3 Showing feature values variation for moving window size 1350.

Similarly, we obtain time domain feature for all the possible working states or conditions in the pump and motor system and store them in feature value tables for classification.

6.3 Classification of process states

After calculating the feature value for all the data sets and storing them in feature selection table we have a total number of 8 different features with 36633 different feature values each. Out of these 70 percent of the values i.e., 25643 we will use for training our classification algorithms and rest 30 percent i.e., 10990 values for testing the results of our classification algorithms. These feature values contain three different classes which are to be classified based on these feature value datasets.

We used 4 different classification algorithms i.e., Discriminant analysis, Naïve Bayes, CART and Nearest Neighbor considering all the features. Each algorithm was run upto 3-features combinations and the best result for each run was recorded. Miss-classification rates results for all the 4 algorithms are shown in Table6.1-Table 6.3 for Discriminant analysis, Table6.4-Table6.6 for Naïve Bayes, Table6.7-Table6.9 for CART and Table 6.10-Table6.12 for Nearest-Neighbor. These tables shows best classification features for all the 4 algorithms and all the x-axis, y-axis and z-axis respectively. Some more features may also be obtained with classification equal to these features or slightly more than the obtained rates by these features.

Table 6.1: Miss-classification rate using Discriminant analysis for X-axis features.

Total features	Features X-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Kurtosis value	3397	30.91
Two features	Kurtosis value and Shape factor	2207	20.08
Three features	Root mean square, Standard deviation and Shape factor	927	8.43

Table 6.2: Miss-classification rate using Discriminant analysis for Y-axis features.

Total features	Features Y-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Root mean square	1263	11.49
Two features	Standard deviation and Shape factor	827	7.52
Three features	Peak value, Clearance factor and Impulse factor	709	6.45

Table 6.3: Miss-classification rate using Discriminant analysis for Z-axis features.

Total features	Features Z-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Peak value	3614	32.88
Two features	Peak value and Clearance factor	2627	24.31
Three features	Root mean square, Peak value and Crest factor	2589	23.55

Table 6.4: Miss-classification rate using Naïve Bayes for X-axis features.

Total features	Features X-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Standard deviation	2487	22.62
Two features	Kurtosis value and Shape factor	1843	16.76
Three features	Kurtosis value, Clearance factor and Shape factor	2001	18.20

Table 6.5: Miss-classification rate using Naïve Bayes for Y-axis features.

Total features	Features Y-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Root mean square	559	5.08
Two features	Standard deviation and Peak value	423	3.84
Three features	Root mean square, Standard deviation and Peak value	428	3.89

Table 6.6: Miss-classification rate using Naïve Bayes for Z-axis features.

Total features	Features Z-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Peak value	3614	32.88
Two features	Standard deviation and Peak value	3165	28.79
Three features	Root mean square, Standard deviation and Peak value	3119	28.38

Table 6.7: Miss-classification rate using CART for X-axis features.

Total features	Features X-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Impulse factor	1586	14.43
Two features	Root mean square and Impulse factor	131	1.19
Three features	Standard deviation, Crest factor and Kurtosis value	27	0.24

Table 6.8: Miss-classification rate using CART for Y-axis features.

Total features	Features Y-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Root mean square	448	4.07
Two features	Standard deviation and Shape factor	1	0.009
Three features	Root mean square, Standard deviation and Shape factor	1	0.009

Table 6.9: Miss-classification rate using CART for Z-axis features.

Total features	Features Z-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Impulse factor	1555	14.14
Two features	Kurtosis value and Impulse factor	263	2.62
Three features	Kurtosis value, Clearance factor and Shape factor	109	0.99

Table 6.10: Miss-classification rate using Nearest-Neighbor for X-axis features.

Total features	Features X-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Impulse factor	1096	9.97
Two features	Root mean square and Impulse factor	85	0.77
Three features	Root mean square, Standard deviation and Impulse factor	1	0.009

Table 6.11: Miss-classification rate using Nearest-Neighbor for Y-axis features.

Total features	Features Y-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Root mean square	460	4.18
Two features	Root mean square and Clearance factor	1	0.009
Three features	Root mean square, Standard deviation and Clearance factor	1	0.009

Table 6.12: Miss-classification rate using Nearest-Neighbor for Z-axis features.

Total features	Features Z-axis	Total number of Miss-classification	Total percentage of Miss-classification
One feature	Impulse factor	1531	13.93
Two features	Root mean square and Clearance factor	257	2.33
Three features	Standard deviation, Crest factor and Kurtosis value	131	1.19

From the miss-classification tables shown it can be seen that CART and Nearest-Neighbor classification are best suited in this case. CART has only 1 miss-classification out of 10990 data values (test cases) in the case of Standard deviation and Shape factor on Y-axis for two feature combination and Root mean square, Standard deviation and Shape factor on Y-axis for three feature combinations.

Similarly, Nearest-Neighbor has only 1 miss-classification out of 10990 data values (test cases) in the case of Root mean square, Standard deviation and Impulse factor on X-axis for three feature combination and Root mean square and Clearance factor on Y-axis for two feature combinations and Root mean square, Standard deviation and Clearance factor on Y-axis for three feature combinations.

Chapter 7

Conclusion and Future work

In this report, we provide details of the pump and motor system sensor data analysis for classifying the working state of the system based on the incoming sensor data. We use this data to create time domain feature value of the time series data which is further used for classifying into different operating states based on these feature values. We provide details about which operating state has what kind of feature variation over the time interval and classified them into various operating states.

Further, we define some visualization techniques so that operator can effectively see the variation in different operating variables of the pump and motor system data. Since, there can be many operating states and which each operating states many control variables associated so, it is very important that each and every detail of the process is clearly and accurately provided to the operator. These variables fluctuation over the time can be best displayed with the help of time series and the techniques defined in this thesis.

In this work, we concentrated on supervised learning methods in which we knew the class labels and trained our system with already known information. In the future work we will use unsupervised learning methods to train our system and classify different operating states. As in the case of supervised learning if any new operating state is encountered then supervised learning will classify it into already existing states. So, we need some efficient technique which can also determine the new states generated in the system which can be dealt with the help of unsupervised learning methods.

Another approach for the similar work which we will follow in future is based on the text mining on the operator log. Each machine operator enters the working state and condition off the machine over time in the form of text. So, by analyzing this text data we will try to predict different operating states of our machine.

Most important task which will be considered in future work will be predicting the machine operating state prior to its actual occurrence. So, that necessary precautions and actions can be taken prior to any occurrence of the fault in machine so, that any time is not lost in repairing it during the operational time and production gets hindered.

8. References

[1]. Zhihua Wang, Time Series matching: *A multi-filter approach,* Newyork University, Newyork, NY, USA, 2006.

[2]. Leichen**,** *Similarity search over time series and trajectory data*, University of Waterloo, Waterloo, Canada, 2005.

[3]. Omar, M. Hafidz. "Statistical process control charts for measuring and monitoring temporal consistency of ratings." *Journal of Educational Measurement* 47, no. 1 (2010): 18-35.

[4]. Draeger, Andreas, Sebastian Engell and Horst Ranke. "Model predictive control using neural networks." *Control Systems, IEEE* 15, no. 5 (1995): 61-66.

[5]. Spyras G. Tzafestas, Expert Control systems: *An introduction with case studies control systems*, Robotics and Automation, VOL XVII.

[6]. Venkat Venkatasubramaniyam, Raghunathan Rengaswamy, Kewen Yin and Surya N. Kavuri, *A review of process fault detection and diagnosis Part I: Quantitative Model based methods*, 2003.

[7]. Venkat Venkatasubramaniyam, Raghunathan Rengaswamy, Kewen Yin and Surya N. Kavuri, *A review of process fault detection and diagnosis Part II: Quantitative Model based methods,* 2003.

[8]. Venkat Venkatasubramaniyam, Raghunathan Rengaswamy, Kewen Yin and Surya N. Kavuri, *A review of process fault detection and diagnosis Part III: Quantitative Model based methods*, 2003.

[9]. Martin Langkvist, Lars Karlson and Amy Loufti, *A review of unsupervised feature learning and deep learning for time-series modelling*, Applied Autonomous Sensor Systems, 2014.

[10]. Paolo Pareti, *Mining unexpected behavior from equipment measurements*, Uppsala University, 2010.

[11]. Anna Hagenblad, Fredik Gustafsso and Inger Klein, *A comparison of two methods for stochastic fault detection: The parity space approach and PCA*, Department of Electrical Engineering, Linkopings university.

[12]. Somkiat Tangjitsitcharoen and Siripong Damrongthaveesak, *Advance in monitoring and process control of surface roughness*, World Academy of Science, Engineering and Technology, VOL:7 2013.

[13]. Biao Huang, *Bayesian methods for control loop monitoring and diagnosis*, Department of Chemical and Materials Engineering, University of Alberta, Edmonton, Canada, 2007.

[14]. Uche R. and Ebieto C. E, *Process monitoring and control of machining operations*, ARPN Journal of Engineering and Applied Sciences, VOL. 5, 2010.

[15]. Varanon Uraikul, Christine W. Chan and Paitoon Tontiwachwuthikul, *Artificial intelligence for monitoring and supervisory control of process systems*, Engineering Applications of Artificial Intelligence, 2007.

[16]. E.B. Martin, A.J. Morris and J. Zhang, *Process performance monitoring using multivariate statistical process control*, System Engineering for Automation, VOL.143, 1996.

[17]. Panos J. Antsaklis, *Neural Networks in Control Systems*, IEEE, VOL.10, 1990.

[18]. H.J. Feise *Handling of Solids-Transport and Storage* Chemical Engineering and Chemical Process Technology-Vol II.

[19]. Chester, Daniel, David Lamb and Prasad Dhurjati. "Rule-based computer alarm analysis in chemical process plants." In *Proceedings of the Seventh Annual Conference on Computer Technology*, pp. 22-29. MICRO-DELCON 84, IEEE, 1984.

[20]. Rich, Steven H. and V. Venkatasubramanian. "Model-based reasoning in diagnostic expert systems for chemical process plants." *Computers & Chemical Engineering* 11, no. 2 (1987): 111-122.

[21]. Epstein, Steven A. "Reasoning and representation in ritse." In *Artificial intelligence and other innovative computer applications in the nuclear industry*, pp. 751-759. Springer US, 1988.

[22]. Kuipers and Benjamin. "Qualitative simulation." *Artificial intelligence* 29, no. 3 (1986): 289-338.

[23]. Hankins, W. W., J. E. Pennington and L. K. Barker. "Decision-making and problem-solving methods in automation technology." (1983).

[24]. Berndt, Donald J. and James Clifford. "Using Dynamic Time Warping to Find Patterns in Time Series." In *KDD workshop*, vol. 10, no. 16, pp. 359-370. 1994.

[25]. Das, Gautam, King-Ip Lin, Heikki Mannila, Gopal Renganathan and Padhraic Smyth. "Rule Discovery from Time Series." In *KDD*, vol. 98, pp. 16-22. 1998.

9. APPENDIX A

Vacuum Pumps in Solid Conveying System

In this chapter, we present the details about the conveyer system and vacuum pump used in this for which we have analyzed the data for classifying the working states.

A.1 Conveyor System

Conveyor system is a mechanical device used in industries for transferring materials required in manufacturing process form one operation point to another. Conveyers are used for lifting those materials in the industry which are not possible to be transported by human one operating location to another. With the help of the conveyer systems materials can be transferred in very less time form one operational point to another.

Pneumatic systems, makes use of pipes or ducts called transportation lines that transport different materials and the air used to operate them. These materials are such as dust and light dry materials like residue generated during the process flow. These residues, materials needed in process flow or left out materials can effectively be transported using conveyers.

Common type of conveyer system used in transportation of these materials are-:

1. Suction or vacuum systems, creating a vacuum in the pipelines to transport the material with the atmospheric air. The arrangement works at a low pressure, and is used mainly in transportation of lightweight materials.
2. Pressure-type systems, which uses positive pressure to transfer ingredients from one operative point to another. This is an efficient method or technique when the material is to be transported to multiple working points.
3. Combination systems, use suction system to transfer ingredients from multiple operating points and pressure system is used to transport these ingredients to multiple operating locations.

In thesis we considered vacuum systems and analyzed the data for its different operating conditions.

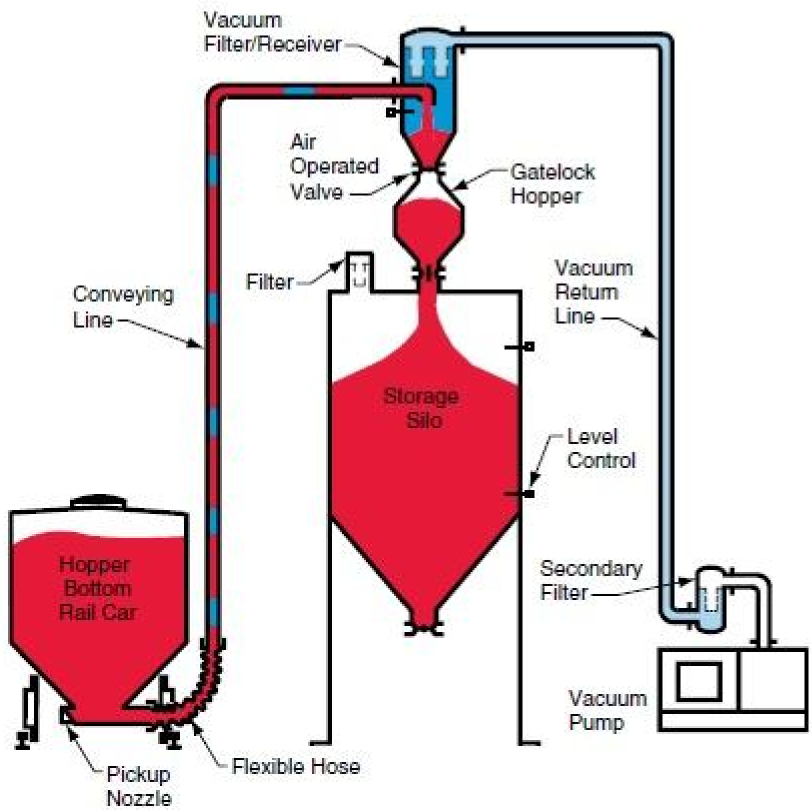


Figure A.1: Vacuum pump conveying system [18].

System shown in FigureA.1 describes the use of vacuum pump in conveyer system for material transportation. The flow of the materials is from low density region towards high density region. The ingredients are passes to the conveyer system continuously by vacuum which is generated with the help of vacuum pump.

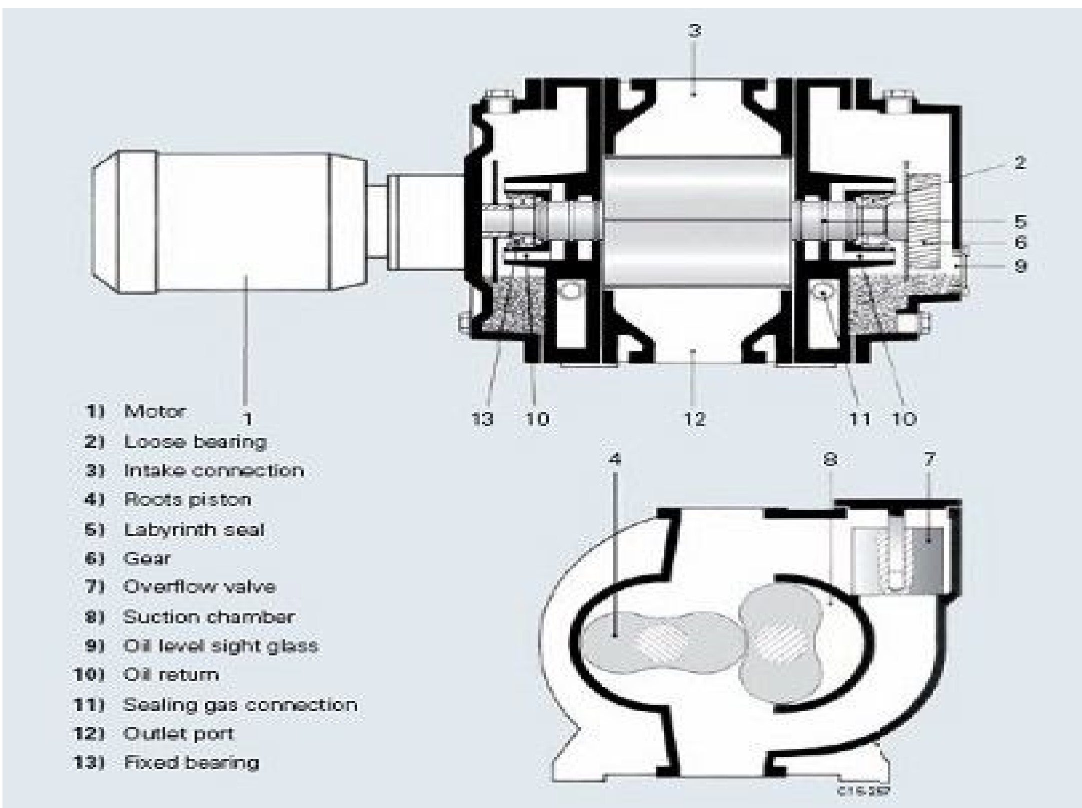
A.2 Vacuum Pump

In this work we considered the vacuum pump system used in solid conveying system for transportation of materials. We collected vibration and pressure data for the analysis of vacuum pump.

Vacuum pump works on the principle of eliminating any gas molecules present thus creating a vacuum for the transfer of materials. Vacuum pump is similar to a compressors in which the release, rather than the intake, is maintained atmospheric pressure. Eliminating air from the bounded system continuously decreases air density within the limited space, triggering the absolute pressure of the residual gas to low. A vacuum is created. A vacuum pump transforms the mechanical drive of a rotation shaft into pneumatic energy by clearing the air present within system. The internal pressure level thus becomes lower than that of the outside pressure. The quantity of energy generated depends upon the volume and the pressure difference generated.

Our work considers the Oil-Lubricated vacuum pumps which have distinct advantages if proper maintenance is provided. They typically deliver a higher vacuum of 20 percent as the oil provides airtight regions between moving parts. Also these vacuum pumps have a life span of about 50 percent more than the oil-less pumps because of the cooling process induced by presence of oil. They also are less subject to corrosion from condensed water vapor. This vacuum pump is a piston pump which can generate relatively high vacuum.

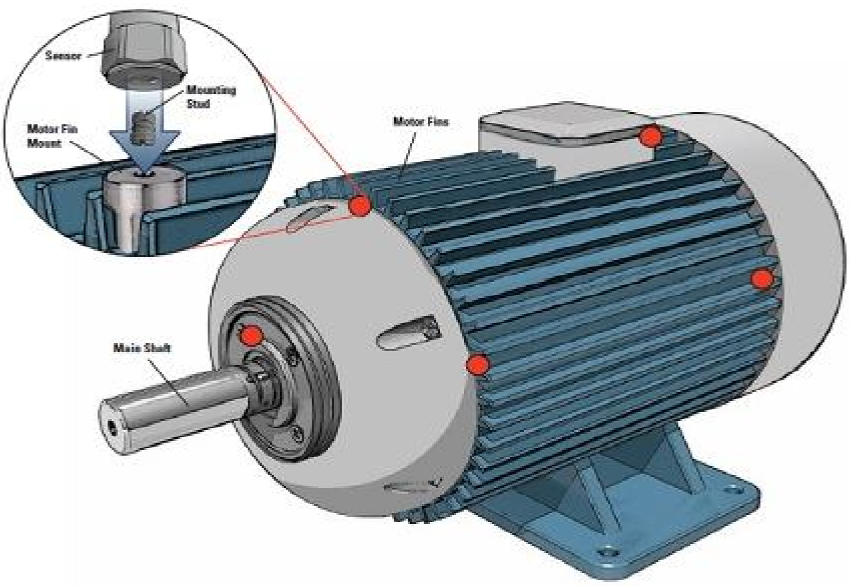
So, the change in the oil-levels of the pump and the air-pressure can create faulty operating conditions which are taken into consideration in our study.



FigureA.2: Internal architecture of piston pump [18].

As the piston moves to produce pressure there are vibrations caused in the vacuum pump and also the change in the oil level in the vacuum pump causes disturbance in the normal operating conditions of the pump. So data for these factors were collected with the help of the sensors attached to the motor system providing operating values of the pump for different operating conditions.

These data values were used to design our system which determines different operating states of the vacuum pump system and guides the operator for the proper functioning of the pump system.



FigureA.3: Showing placement of sensors in pump and motor system [18].

FigureA.4: Sensor reading in the 3-co-ordinate axis [18].