Big Data Analytics and Its Applications in Soft Sensor for Smart Manufacturing

by

Devarshi Kamleshkumar Shah

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 4, 2019

Keywords: Big Data Analytics, Smart Manufacturing, Industrial Internet of Things, Machine Learning, Statistical Modelling, Process Monitoring.

Approved by

Jin Wang, Chair, Walt and Virginia Woltosz Endowed Professor, Chemical Engineering
Q. Peter He, Associate Professor of Chemical Engineering

Mario Eden, Joe & Billie McMillan Prof. & Chair, Chemical Engineering

Selen Cremaschi, Redd Associate Professor, Chemical Engineering

Jesus Flores-Cerrillo, Associated Director, R&D, Praxair Inc.

Abstract

This dissertation presents research performed to develop industrial internet of things enabled testbed and statistics pattern analysis feature-based modelling approach for smart manufacturing.

Advent of internet completely revolutionized the way humans communicate and share information. With rapid growth in internet users, amount of data generation and exchange increased exponentially resulting in ever increasing demand for faster internet. This demand drove multidisciplinary innovations resulting in fast & reliable wireless internet, cheap storage memory and efficient computation resources capable of processing large amount of data. Large amount of data generated by the users, when modeled effectively, offers the potential to make information exchange more efficient and productive. Internet of things is an extension of the same idea but with goal of improving efficiency, productivity and in turn profitability of a business.

Network of things or items capable of exchanging information using some communication protocol is called Internet of things (IoT). Many health, retail and ecommerce businesses have benefited by modelling data generated from these IIoT devices. Combination of IIoT sensors and data modelling using machine learning techniques presents manufacturing industry with immense potential to identify and mitigate bottlenecks resulting into a more efficient manufacturing or smart-manufacturing.

Smart-manufacturing is still at its incipient and much more research is required both at academic and industrial scales. In this work, Industrial internet of things enabled test-bed was developed to demonstrate entire smart-manufacturing project pipeline. It also provides idea about how a normal unit operations can be transformed into smart unit operation which is capable of real time process monitoring. This study provides discussion on sensor selection, micro-controller selection, data transfer and storage architecture, data compilation, noise characteristics over type of internet connections, data mining, possible way of dealing with data synchronization, model based data filtering by introducing new statistical approach. Moreover comparison study of recurrent neural networks, artificial neural networks and hierarchical modelling approach for prediction of flowrate and pump RPM will be presented.

Another important aspect of smart manufacturing that was addressed is development of modelling approaches that can capture relevant information from a complex system while making resulting models more robust, simpler and accurate. A new feature based modelling approach has been proposed for spectrum dataset. In this approach relevant features are engineered from original raw spectrum with the help of different statistics. The proposed approach is compared with different conventional soft sensor approaches. Moreover this comparison study was carried out on four publically available industrial datasets. A Monte-Carlo validation and testing procedure was also devised in order to clearly indicate identify if any modelling approach's performance is better or it just seems better because of bias variance tradeoff.

Acknowledgements

My journey for this dissertation would have never been possible without constant support and encouragement from many individuals. Firstly I would like to express my deepest gratitude towards my advisors Dr. Jin Wang & Dr. Qinghua He for their constant support and guidance. They always believed in me and are epitome of dedication and hard work. Their suggestions and inputs helped me develop versatile & In-depth understanding about problems which in turn made me a better researcher. I would also like to extend my special thanks to Dr. Mario Eden for providing his advice during times of uncertainty, scholarly or personal.

I would also like to thank my other committee members Dr. Selen Cremaschi, Dr. Jesus Flores-Cerrillo & Dr. Aleksandr Vinel for their assistance in completing this work. I want to thank my current and past Wang's group members, especially Kyle Stone, Matt Hilliard, Nathan Roberts and Kiumars Badr for being first line of troubleshooting. For financial support I want to extend my gratitude towards National Science Foundation.

Although no amount of thankfulness and gratitude would be enough, I would like to extend my deepest gratitude to my parents, Jigisha Shah & Kamlesh Shah, for not only supporting during the journey but also for instilling deep values and strong ethics in me. I also want to thank my sister, Chandni Mehta, and my brother-in-law, Bhavin Mehta, for

constantly being best of friends, philosophers, guides and more during all stage of my life. I also want to thank Krupali Shah & Vijay Vanker for their immense support and help in this journey. I would also like to thank my beautiful wife Sayali Barshikar for being all the positivity, smiles and encouragement in my life.

Last but not the least I receive my values, grit, motivation and energy for making positive impact with purposeful innovation directly from spiritual god ("Karma Yogi"). I thank you for always being my inner code of honor.

Devarshi K Shah

Auburn, Alabama

April 16th, 2019

Table of Contents

List of Tables

List of Figures

List of Abbreviations

σ^2	Variance
σ_{NRMSE}	Standard deviation of normalized root mean squared error
K	Kernel function
γ	Skewness
κ	Kurtosis
μ	Mean or average
σ	Standard deviation
AFD	Average of first derivative
ASD	Average of second derivative
BM-PLS	Binary matrix partial least squares
COV	Coefficient of variation
CVA	Canonical variate analysis
DCW	Dry cell weight
DI	De-ionized
DNN	Deep neural network
FSS	Forward stepwise selection
GB	Giga-bytes
gpm	Gallon per minute

HDMI	High-definition multimedia interface
Hz	Hertz
I2C	Inter-Integrated Circuit
ICV	Inverse of coefficient of variation
IIoT	Industrial Internet of things
IoT	Internet of things
iPLS	Interval partial least squares
KF	Kalman filter
K-PLS	Kernel partial least squares
Lasso	Least absolute shrinkage and selection operator
LSTM	Long short term memory
MA	Moving average
MCCV	Monte Carlo cross validation
MCVT	Monte Carlo cross-validation & testing
MHz	Mega Hertz
MIMO	Master out slave in
MISO	Master in slave out
MPE	Mean prediction error
NIPALS	Nonlinear-iterative partial least squares
nm	Nano meter

NMPE	Normalized mean prediction error
NN	Neural network
NRMSE	Normalized root mean squared error
OD	Optical density
OLS	Ordinary least squares
OS	Operating system
PCA	Principal component analysis
PCs	Principal components
PLS	Partial Least Squares
RMSE	Root mean squared error
RMSECV	Root mean squared cross validation
RMSEP	Root mean squared prediction
RNN	Recurrent neural network
Rpi	Raspberry pi
R-PLS	Recursive partial least squares
RPM	Revolution per minute
RWR	Recursively weighted regression
SCL	Serial clock
SDA	Serial data line
SiPLS	Synergy interval partial least squares

SLL	Slope of linear regression line
SM	Smart manufacturing
SPA	Statistic pattern analysis
SPI	Serial peripheral interface
SSH	Secure shell
SSL	coefficient of squared term for second order regression line
SVM	Support vector machine
SVR	Support vector regression
TB	Tera-bytes
UET	Unix epoch time
USB	Universal serial bus
VM	Virtual metrology
W	Watts

Chapter 1. Introduction

Internet of Things (IoT) means communication between physical things with the help of the internet. IoT connects people and computers with these physical things into a digitally connected network. These things measure a variety of properties using different sensors and report the data. If the data collected is modelled correctly, it can give invaluable information that can improve reliability and efficiency as well as reduce the overall cost associated with physical things. Different industries have implemented IoT frameworks and have observed tremendous improvements in overall functionality and profitability of the businesses, thus IoT is heralded as the next industrial revolution. Different names have been used to describe this next generation manufacturing approach, such as smart manufacturing, Industry 4.0 and intelligent manufacturing; however, the essence of these approaches is development of Industrial Internet of Things (IIoT) by incorporating increasingly powerful and low-cost computation and wirelessly networked information-based technologies in manufacturing facilities.

At the heart of this emerging revolution is the availability of large amounts of data and the ability to harness this data for purposeful innovations. The word “Big data” is used to describe such large quantities of data which can be used to generate valuable insights. For data to be considered as big data, it should typically exhibit four key characteristics or the 4 V’s of big data, 1) volume, data having large volumes in memory (several GBs, TBs). 2) Variety, data obtained from a variety of locations or different

types of information from same or different locations, 3) velocity, data obtained not only at fast rates but also obtained at different rates, 4) veracity, data having many uncertainties *i.e.* data influenced by noise from various sources. The field which models such data by solving the challenges associated with 4 V's of big data using a combination of statistics, mathematics and machine learning is called big data analytics.

A chemical process or manufacturing plant can be considered a warehouse of data where a large number of different process measurements are collected and stored every day. However, many times these process measurements are not available due to the nature of the material being handled, (e.g. flow measurements in services that deal with tar, heavy hydrocarbons, or dual phase transfer) nature of the operations (e.g. loss of sensor connection in centrifuge), or by an accident or unknown reason (e.g. loss of sensor mount, unexpected damage to sensors). Operating plants in such situations can be dangerous and may lead to severe catastrophes, moreover such loss of quality variable measurements may force plants to operate sub-optimally and in the worst case plant needs to be shutdown, resulting in loss of production.

Incorporation of IoT sensors along with big data analytics has a high potential to mitigate such problems and make the entire manufacturing process smarter. Fully integrated unit operations with complete real-time knowledge about each of their conditions and limitation will enable overall systems to make more informed optimization and control decisions resulting in a system that is smarter, more efficient and productive than their non-connected counterparts. More details in [1], [2].

Smart manufacturing is still at its incipience and there are still many challenges that needs to be addressed. Some examples of challenges to be addressed are: the interaction

between different equipment is often non-linear with unknown functions, the amount and number of sources for disturbance and noise are prohibitively large, the high cost of getting data to identify different working conditions of several pieces of equipment, the availability of data, the processing capacity, the choice of modelling techniques and their limitations, equipment failure, measurement failure, space constraints *etc*. Despite these challenges, the first step to make a manufacturing process smart is the development of Industrial Internet of Things (IIoT) built with the help of IoT devices.

IIoT devices are sensors, actuators and computers with wireless networks, and most importantly, systems that are small and easy to embed. Although the use of IIoT sensors has been increasing exponentially in the retail and services sectors, their use in manufacturing has been limited. Because of their small size and cheap price, IIoT sensors offer the opportunity to enable manufacturing systems with monitoring of novel properties resulting in data large amount of data collection. With the huge amount of data and the programmability of these IIoT devices, comes the opportunity to shape the data received, to address local redundancy of information, and to improve both the accuracy and precision of measurements locally and across a distributed parameter of unit operations [3]. Therefore, IIoT devices have potential to enable data-driven modelling of different unit operations making them capable of real time monitoring. Such IIoT enabled unit operations form the building block of Industrial IoT for smart manufacturing. Figure 1.1 shows schematic of overall IIoT enabled smart manufacturing process where all the different unit operations continuously talks with a head, *i.e.* data from IIoT enabled unit operations is sent to a computing head. This analytical head contains predictive, proactive models developed using combination of mathematics, statistics and machine learning

techniques in order to make decision which results overall manufacturing process to be more efficient, smart & sustainable.

With industrial IoT still in its infancy, there is not a sufficient understanding on the property, capacity and performance of IIoT sensors to enable accurate simulation [3].

Figure 1.1 Schematic of overall IIoT enabled manufacturing process

Moreover, several attempts and overall frameworks have been proposed [[4]–[6]] but to the best of author's knowledge no case specific research has been published, which informs definitively about sensors, data characteristics, network characteristics, data quality, and ways to develop the monitoring framework. The lack of case specific research was the major motivation to build the IIoT enabled smart manufacturing (SM) testbed presented in this work. Other motivations for this study were to identify IIoT sensors' data characteristics, data challenges associated with data collected SM testbeds and to identify and explore novel ways to predict and monitor key process information.

This study focuses on mitigating issues caused by the traditional sensors and the service they are used for, as discussed above, by combining predictive modelling, IIoT sensors and big data analytics resulting in a smart manufacturing testbed. The testbed can predict important process properties using non-conventional, non-invasive IIoT sensors like vibration sensors (feel) and cameras (visual). The entire IIoT enabled smart manufacturing testbed development pipeline is discussed, and a predictive monitoring and fault identification framework is proposed. Moreover, IIoT sensor data characteristics were studied and a discussion of how the data generated qualifies as big data is presented. Next different predictive modelling approaches (long-short-term memory neural networks, deep neural networks (DNN), partial least squares (PLS) *etc*.) were tested and there performances were compared to establish the importance of incorporating system/process knowledge in model development and applicability. A new application of inverse of coefficient of variation statistic is proposed and implemented for data filtering in frequency spectra. A hierarchical PLS modelling approach for smart manufacturing testbeds, similar to the one presented in this work, is proposed. A new signal reconstruction approach by combining principal component analysis (PCA) and partial least squares (PLS) is developed and presented.

Data-driven soft sensor models have been used to capture many indirect and/or complex relationships [7], [8]. Soft sensors are models that identify the relationships between process property/information, such as product quality [9] which is often difficult to measure, with easily measured properties. Soft sensors are also used to provide prediction of infrequently measured process variables so that faster, more informed control actions can be taken [10], [11]. Information about a novel soft sensor approach

developed by the author is given in Appendix B. The advantages of soft sensor makes it extremely useful and important to achieve smart manufacturing.

Although soft sensors have been extremely useful in correlating properties of interest with easy-to-measure variables there are many limitations [12]. Presence of large number of variables interferes in finding the global optimal model performance and it has been shown that performance of data-driven models can be tremendously improved by selecting only vital variables that strongly relate to the primary variables [10], [11]. However this causes loss of information, and if the relation between secondary variables and properties of interest is weak then overall model performance will be limited in accuracy.

Taking a view of a data driven future (where novel secondary variables will be measured) the author believes that as every physical thing has vibrations or is affected by vibration, absorption spectra (light, sound, *etc*.) will form key secondary variables. Therefore, in order to model the complex relationships between primary variables & secondary variables, while still keeping the dimensionality of the secondary variables low, a feature based soft sensor modelling approach for spectrum data has been proposed.

The work presented in this dissertation is structured as follows. In Chapter 2, physical details and considerations for selection and development of an IIoT enabled tested b is discussed. Chapter 3 provides information about experimental design and IIoT sensor data characteristics. More specifically, how the data generated by the IIoT sensors can be characterized as containing the 4 V's of big data as well as the static noise characteristics of the sensors is discussed. Chapter 4 provides details about initial modelling, primary data analysis and feature extraction or data filtering. Additionally, a

brief description of modelling approaches which have been used in the study is also provided. This chapters shows how model selection and performance can be improved by incorporating primary system knowledge, where a comparison of a full PLS model & DNN is performed. Chapter 5 focuses on the importance of combining knowledge based learning with advanced analytics by conducting a detailed data analysis and data visualization to identify scientific fundamentals underlying the processes. A system knowledge enabled hierarchical modelling approach is presented for the testbed by proposing a novel peak frequency selection criteria and by mitigating noise characteristics. The overall framework is also completed by presenting new PCA-PLS based reconstruction approach. Chapter 6 provides background about different linear soft sensor modelling approach & new feature based soft sensor approach has been proposed for spectrum data. A Monte Carlo validation and testing approach has been devised for systematic comparison of performance of different predictive modelling techniques especially when sample size is small. Chapter 7 provides detailed comparison of the proposed feature based soft sensor approach with other linear approaches using industrial datasets from different fields of science in turn proving versatility and applicability of the proposed approach. Moreover a non-linear extension of the proposed approach is also discussed with examples provided. Finally, Chapter 8 provides a description of the overall framework with a schematic, gives conclusions from the studies, and provides suggestions for future work.

Chapter 2. IIoT Enabled Multi-stage Centrifugal Pump Testbed

A manufacturing plant consists of different unit operations. Each unit operation consists of several different equipment with the best possible synchronization. The health of any given unit operation is dependent on each individual piece of equipment functioning properly. This makes proper functioning of each equipment important and also interdependent. One of the most versatile equipment in manufacturing industry are centrifugal pump, compressors and the associated piping system that move gas or liquid from one location to another. Therefore, for development of the IIoT enabled testbed it was decided to mimic working of industrial multi-stage centrifugal pump so that the developed framework could be widely applicable. Figure 2.1 shows the process setup used for this work. A centrifugal pump is a system with a number of interacting parts, and one of the most commonly known sources of information in any such piece of equipment are the vibrations being produced. The application of vibration data for condition monitoring of machinery or structure has been well documented, such as the detection of faults or defects in gears, rotors, shafts bearings and couplings [13]–[16]. However, their applications for information such as rotor speed and fluid flow rate has not been reported. Therefore, for this work it was decided to collect vibration data from different parts of the pump in the hope that later information regarding different operating stages of a pumps will captured and in-turn be successfully modelled.

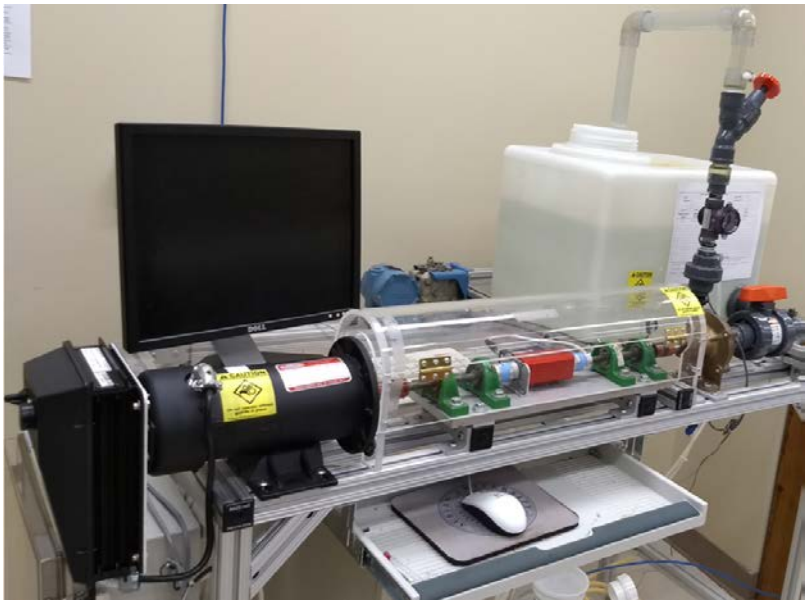


Figure 2.1 multi-stage centrifugal pump setup

2.1 Major goals

1. Build a non-invasive IIoT vibration sensor enabled multi-stage centrifugal pump testbed capable of conducting the designed experiment.
2. Develop data transfer and storage architecture for IIoT testbed mimicking cloud computing.
3. Study data characteristics of IIoT sensors and noise characteristics from wireless networks.
4. Development of framework for monitoring and prediction of different properties of interest or critical quality variables.
5. Sensor fault detection procedure for the testbed.

2.2 Advantages of the Testbed

1. Entire setup is scalable and useful in all the industries as pumps are ubiquitous in manufacturing.

2. Use of non-invasive sensors makes entire proposed approach applicable not only on new sensors but also on existing legacy processes.
3. Use of vibrations for estimating flowrate instead of flowmeters has a very high potential to address safety and un-planned shutdown issues associated with choking prone operations in petro-chemical, oil & gas industries. Many variants of this concept can be adopted to solve a variety of manufacturing problems.
4. Can be applied for real time fault identification & control.
5. Such a setup can act as a layer of safety for many industrial processes.
6. Cheap to assemble and easy to maintain.

2.3 Sensor selection

Two major types of vibration sensors were considered and tested in this work: Piezo type vibration sensors and Accelerometers. Piezo type vibration sensors are sensors that contain a material which, when moved or touched, produces voltage which is measured to identify vibration or motion of the surface on which it is placed. Accelerometers are electromagnetic devices that measure the acceleration force on the sensors which in turn can be used to sense vibration or movements. Figure 2.2 shows the pictures of different piezo & accelerometer vibration sensors tested for this work.

Piezo type vibration sensors are in general the cheapest, require minimal connections and do not require complex breakout boards. However, during the initial testing it was observed that Piezo type sensors had very low sensitivity for centrifugal pump application, and are most likely not suitable for industrial processes if process information buried in the vibration signals is to be extracted & modelled. On the other hand, Accelerometers are slightly more expensive at around $10 in oppose to $1.50 for

piezo, at the time of writing about the study, but are highly responsive, easy to connect

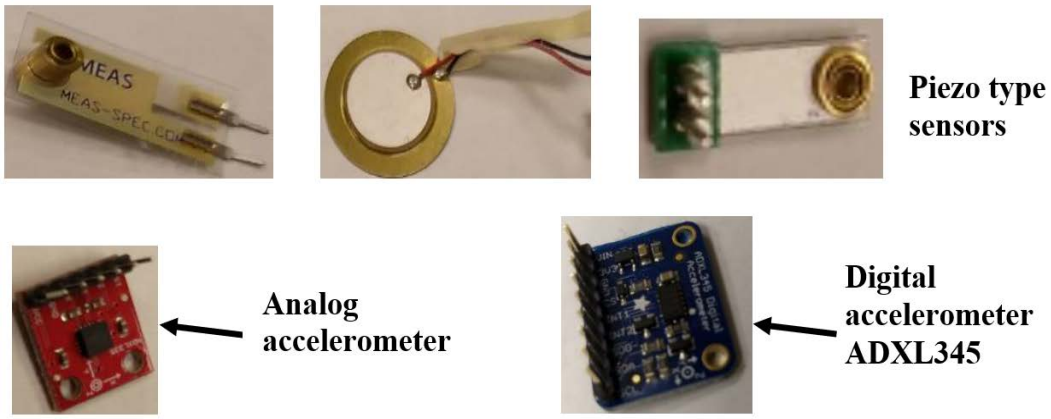


Figure 2.2 Types of sensors tested

either using standard I^2C (Inter-Integrated Circuit) or SPI (Serial peripheral Interface) protocols. Thus it was decided to use accelerometers for this work.

Accelerometers have two main categories: Analog accelerometers, which need an external analog to digital converter (ADC), and Digital accelerometers, which have built-in ADC. Having an external ADC increases manual connection points and in-turn increases sensor failure points while increasing the size of the overall sensor setup. Analog accelerometers are cheaper that their digital counterparts, but when considering overall reliability of the setup, space constraints and overall smaller breakout boards (ADC+sensor), digital accelerometers clearly shines over analog accelerometers and thus were used for this study.

It was decided to use ADXL345 tri-axis digital accelerometer in Adafruit breakout board. Major advantages of using this particular sensor are, it measures components of vibrations in three directions (x, y & z) which provides more information for data analytics, it can use both two wire I^2C or SPI (3 or 4 wire) protocol for communicating with any computing device, its sensitivity is adjustable (+-2g, +-4g or +- 8g), its sampling rate is

adjustable (800 Hz, 1600 Hz, or 3200 Hz), it has built-in low pass filters for lower sampling rates, as well as a wider temperature range (-40^{o}C to 85^{o}C), a smaller size (3mm X 5mm X 1mm), and strong community support [17]. Figure 2.2 shows ADXL345.

2.3.1 I^2C vs SPI Connection Protocols:

Currently these sensors do not contain built-in controller which can directly communicate with cloud servers, so it was decided to use a third party "micro-controller" or computing devices. Details about the micro-controller used for this work will be discussed in the next section. Communication between sensors and micro-controllers can be established using either of the two standard data transfer protocols I^2C and SPI. Both are bus protocols that allow short-distance data transfer, commonly used in electronic devices like smart phones, Televisions, laptops, *etc*. Figure 2.3 shows schematics of both protocols. I^2C system sends bidirectional data over serial data lines (SDA). I^2C needs just two lines, connection serial clock (SCL) lines & SDA. On the other hand SPI is a point to point connection data, where in and out flow happens on two separate lines. It needs at least 4 lines, connection serial clock (SCLK), master out slave in (MOSI), master in slave out (MISO) and slave select. Clock lines for both the protocols are used to synchronize all the data transfer over the respective buses. Multiple devices can be connected to each bus. However for I^2C all the devices can be connected over same two lines while for SPI separate slave select line needs to be added in order to keep track of each device and its corresponding data. Because of its design SPI is faster and easy to interface than I^2C, Highest speed for I^2C devices operates around 1MHz transfer rates which is much slower than 20MHz for SPI. There are many other technical differences and advantages for each of the protocols , but extensively comparing the two protocols is not within the scope of this study, and more details about them can be found in [[18]–[20].

For this work it was decided to use I^2C over SPI because it requires only two hard wire for communication with the computing devices and more devices can be connected on the same wire. This consideration was quite important for future scale up and reliability as more wired connection increases chance of disconnection. Moreover, I^2C communication capacity was fast enough for the application under consideration and sensors used. Although this is just a minor selection preference, and one can easily switch between any communication protocols without making any major change in the approach proposed for data analysis and data-driven predictive models later in the study.

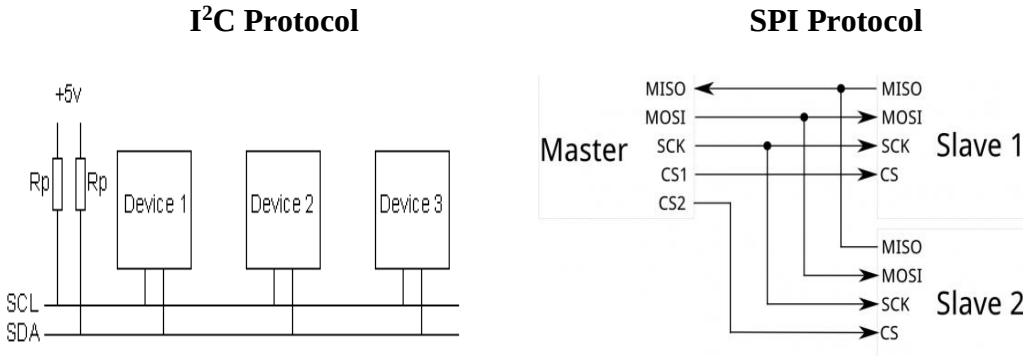


Figure 2.3 Schematics of I^2C and SPI protocols

2.4 Computing device or micro-controller:

Sensors are electro-mechanical devices which sends signals based on the changes in its position, but a master device is required so that each sensor can be controlled based on the user requirement, as well as allowing data tracking and labelling to be achieved for easier data analysis and control. This master device is called a micro-controller, micro as they need to as small as possible again for scale up and space limitations.

For this study it was decided to use a computing device called Raspberry Pi. The major advantages of using this device as a micro-controller are:

1. It is extremely low power drawing- 5 to 7 W of electricity. This can help in reducing electrical costs as pumps are generally working 24/7 and continuous data collection is required.
2. It has small form factor (small in size), which comes in many models and can be as small as 2.6 inch longest dimension. Developers are coming up with still smaller yet power versions of this device.
3. It has no moving parts, which results in a smaller chance of mechanical failure.
4. It can work with multiple types of sensors and other devices, therefore compatibility issues can be alleviated.
5. It is extremely low-cost.
6. It has a huge open source community support, thus troubleshooting for hardware and/ or software bugs can be performed more efficiently.

More details, advantages and disadvantages about the Raspberry Pi (Rpi) can be found in [21]–[24]. Figure 2.4 shows two Rpi models, Rpi zero & 3 model B.

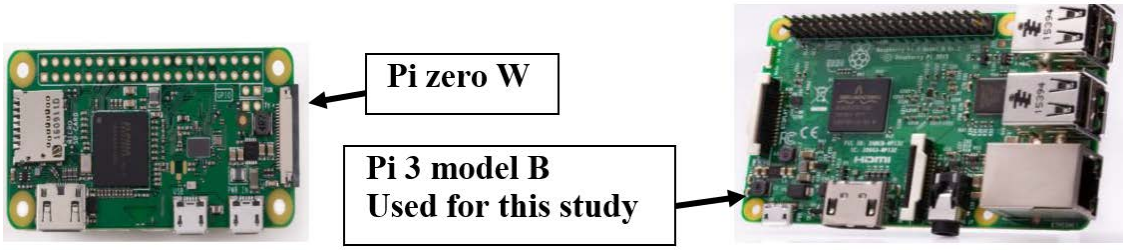


Figure 2.4 Models of Raspberry Pis

For this work it was observed the Raspberry Pi 3 model B version was well suited and ran without any major hiccups. It has built-in HDMI, Wi-Fi, and USB capabilities which are very important for troubleshooting any problems and establishing data reliability during the research stage. However, any other type of micro controller can be

used based on the application preference without affecting any data analysis, data filtering and modelling methodologies presented later in this work.

2.5 Adopted Cloud computing like system architecture:

Designing a complete cloud computing architecture requires much more knowledge of computer science than just coding. For designing this architecture, collaborators from Dr. Skjellum's lab department of computer science and the Network Services department in Auburn University were heavily involved for hardware and software selection. Figure 2.5 shows that the overall design architecture was designed based on subscriber -

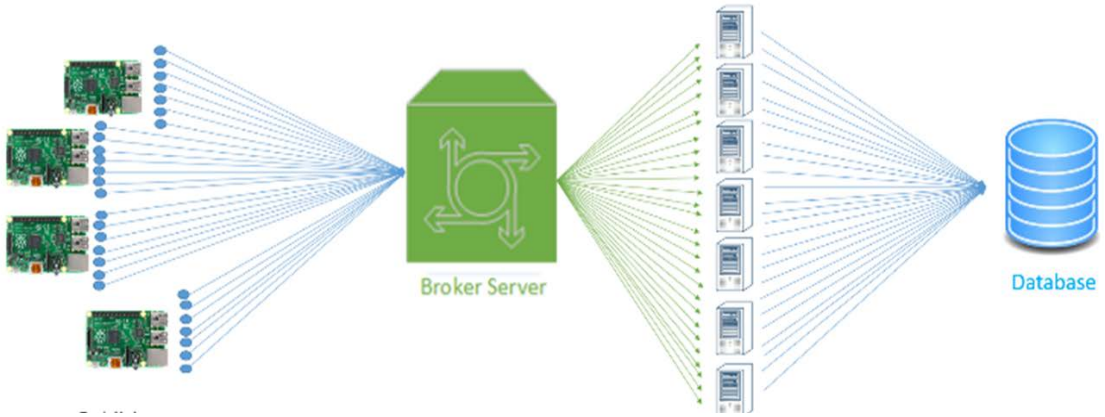


Figure 2.5 System Architecture

publisher pattern, and was adopted to achieve online real-time control of data collection, transfer and storage from and to remote location/servers.

Under the adopted architecture, multiple Rpis received required data from their corresponding sensor. They simultaneously transfers data to a remote server. Data from the server can be accessed from multiple location. It is deliberately sent to different locations in order to mimic safety and redundancy of actual data. Finally, data is stored in a database, and from there any required data is processed, compiled and analyzed based

on business or application requirement. For this study this architecture closely followed. However, in order to incorporate high frequency data, transfer initially data was logged on the broker server and later was accessed from publishers to store on the databases.

All the Rpis were connected to the broker using wireless network and were accessed and controlled remotely making it at par with the future IIoT enabled industrial systems. Issues related to cyber security, network issues & system robustness are out of the scope of this study, as these are issues of lesser relevance for data modelling and analytics. Issues that affect the overall data quality and reliability were considered and are discussed later chapters.

2.6 Experimental setup:

Experiments were performed by mounting tri-axis accelerometers ADXL345 at different locations around the variable drive pump assembly so that vibration signals could be captured and analyzed without making any changes to the original design of the pump (non-invasive).

2.6.1 Pump assembly and working without IIoT sensors:

The pump assembly contains a variable drive motor, pump impeller, impeller casing, coupling, electrical connections, and knobs for changing pump revolution per minute (RPM). The motor shaft and impeller shaft are connected by a coupling. The pump sucks water from a reservoir and pumps it back to the reservoir, and has both a suction valve and a discharge valve, but no bypass. There is one flow meter at the pump discharge. The pump RPM can be adjusted by turning the physical knob. Pump flow can be changed independently either by opening or closing pump discharge valve or by changing pump RPM. The pump RPM and flowrate are continuously indicated and will be used as a base

value for building predictive models. The range of operation for the pump is from 1500 RPM to 2500 RPM. For the discharge valve, the minimum flow at 1500 RPM and at the minimum discharge valve opening, which can be measured by a flow meter reliably, is around 5 gallon per minute (gpm), while the maximum flowrate that can be achieved at 2500 RPM and maximum discharge valve opening is around 16 gpm. The term “around” is used as the centrifugal pumps used in this case have a higher variation in flowrate and can’t be fixed exactly at a particular value.

2.6.2 Sensor – pump assembly attachment:

It was decided to attach 5 ADXL345 on the pump assembly. Sensors locations were selected based on process knowledge so that important vibration signals related to the properties of interest (flowrate & RPM) could be captured. A sensor was mounted on the motor casing, the impeller casing, the coupling joining motor and impeller, the pipe fitting and on the loose end of the pipe. The attached sensors were connected so that they would have maximum contact with the surface of the assembly without affecting its

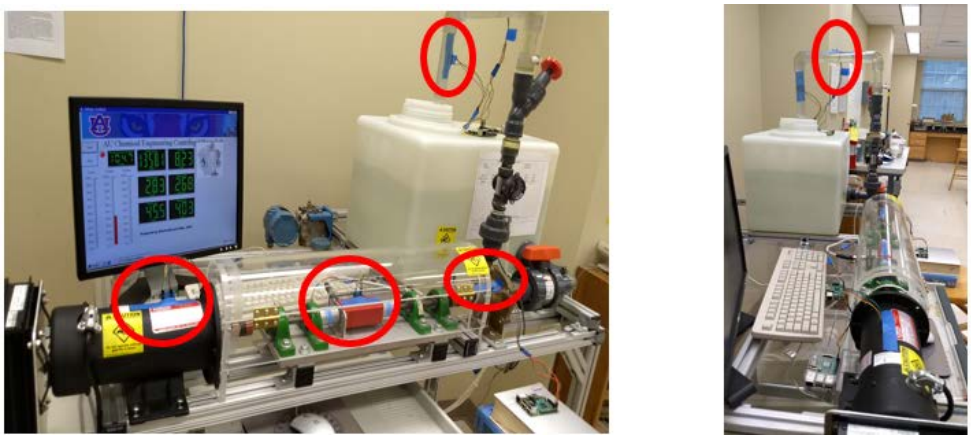


Figure 2.6 IIoT enabled pump testbed (red o: IIoT sensors)

operation. Each sensor along with its Rpi were numbered for tracking the data. Figure 2.6

shows the pump testbed with IIoT sensors. Figure 2.7 shows schematic of the testbed with sensor location.

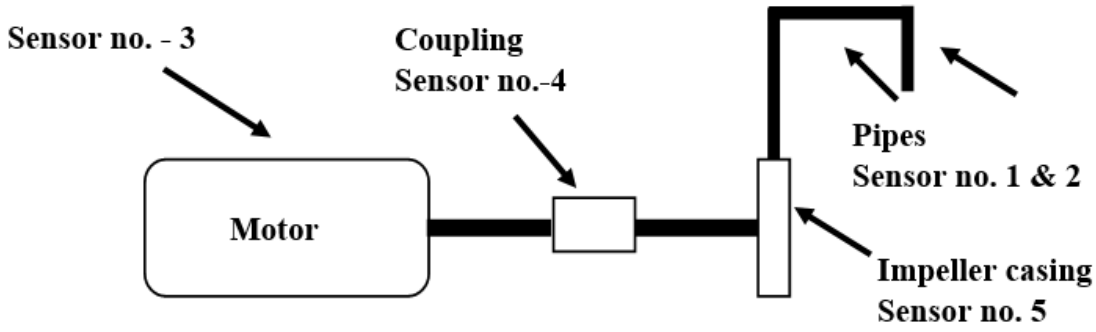


Figure 2.7 Schematic of the testbed with sensor location

2.6.3 Sensor- Pi Connection Data collection and Storage:

Each sensor was attached to individual Rpi using the I^2C device connection protocol. Figure 2.8 shows sensor connection to the Rpi and the pins used for the connection. All Rpis were connected to private and secure wireless networks. They were accessed

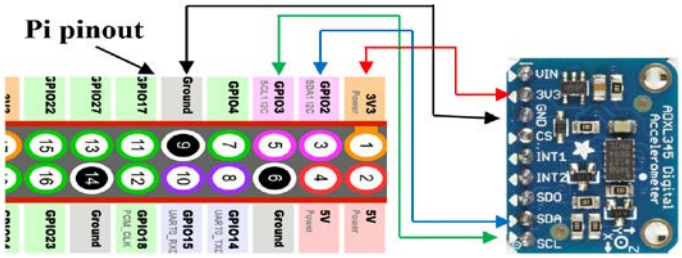


Figure 2.8 Sensor connection to pi: Red: Power, Black: ground, Blue: SDA and Green: Clock

remotely using secure shell (SSH). Software putty version 0.67 was used to establish SSH and access Rpis remotely. The remote server/broker used to access the Rpi was a regular desktop PC with the Windows 10 operating system (OS) with a wired secure university internet connection.

Once the entire setup was ready for data collection, the user commanded the Rpis to start signaling sensors for data collection from the remote server/broker. The data

received by the Pis were sent to the server in real-time. This remote server logged the data it received and stored it in separate files. There, files were manually sent to the separate remote publisher based on the publisher's user requirement, from where the entire data was stored on a relational database, SQLite, which can later be accessed by data analysts and modelers. It can be observed here that the entire data transfer and storage architecture as discussed before was implemented here, and thus this entire setup can be easily scaled up to an industrial level.

2.6.4 IIoT Enabled Testbed other consideration:

The pump used has a drift, so it doesn't operate at an exact fixed RPM. This is common in many centrifugal pumps. Thus when the pump is fixed near any fixed value of RPM it tends to drift at around a 5-7 RPM range, which is acceptable. The flow meter and indicator isn't very accurate, and the resulting measurement is noisy. However these limitation are not prohibitory, and the visible intended change can be observed in order to collect data for framework development.

Chapter 3. IIoT Testbed Experimental Design and Data Characteristics

3.1 Experimental Design:

As previously discussed, one of the major goals of this work was to develop monitoring and sensor fault identification framework by estimating process information like RPM & flowrate in the pipe with the help of vibration signals obtained from non-invasive IIoT sensors. In order to achieve this, the experimental design was drawn with the motive of establishing and identifying a relationship between vibration signals and flowrate. As flowrate of the system can be independently controlled either with pump discharge valve or RPM knob, the relationship between vibration signals & flowrate can be different for distinct setting of discharge valve and RPM know. Moreover, the relationship between RPM and vibration signal needs to be identified.

Based on these considerations, it was decided to collect vibration signal data at different combinations of conditions. First RPM of the pump is fixed at a decided value and then vibration signals are collected at different flowrates. This process was carried out at different RPMs covering entire range of the pump operation. Thus, each combination of RPM & flowrate is considered as a condition. Vibration data was collected at 85 different conditions (big data characteristic = Volume & variety). A list of conditions at which data was collected is given in appendix A.1. For each condition data was collected for 10 mins. For example, pump RPM was fixed around 2000 RPM then using pump discharge valve flowrate is set at approximately 7gpm. Once this condition is fixed, vibration signals from all the sensors were collected for 10 mins and stored.

In order to synchronize start time of all the sensors Unix epoch was used. It is the number of seconds that have elapsed since January 1, 1970, not counting leap seconds [[25]–[27]]. In this study microsecond version of Unix epoch was used.

Existing IIoT testbed only had RPM & pipe flowrate indicators, *i.e.*, no capability of storing them in the memory for modelling and analysis purpose. Moreover, in order to build and optimize real time predictive model one needs record of RPM & flowrate values at corresponding time of vibration. To capture this information, in real time it was decided to take video recording of the RPM & flowrate indicator (big data characteristic = Volume, Variety). Subsequently, this video was used to automatically extract digits using a combination of video image processing & statistics.

It was observed that on average the flowrate value changed every half second; therefore, sampling frequency of RPM & flowrate values was fixed at 3 Hz (big data characteristic = Velocity, different sampling frequency of vibrations & RPM-flowrate). Although this choice is realistic and high enough for most applications other desired sampling frequencies can be used.

3.2 Data Characteristics:

3.2.1 RPM data:

Figure 3.1 shows deviation in RPM values observed due to pump drifting for a representative set of RPM conditions along with big data characteristics which the data represents. Similar behavior is observed for all the conditions. Based on this analysis it was observed that average maximum difference between maximum RPM and minimum RPM is 7. Drift was smaller at lower RPM conditions. Further, this drift is approximately less than 0.5% for smaller RPM (1500-1700). There are 2-3 samples for which the

difference is around 10-12 RPM which is a negligible number of instances. However, this shows big data characteristic of measurement veracity.

3.2.2 Flowrate data:

Figure 3.2 shows histogram of all the flowrate values measured at different RPM conditions along with big data characteristic it represents. It can be clearly seen that flow measurement shows a Gaussian distribution. This distribution can be due to three reason; firstly, centrifugal pump is not accurate and actual flowrate obtained from pump is Gaussian distribution. This is possible, because as stated above, the pump RPM tends to drift and if this drift is Gaussian then flowrate change can also be Gaussian. Secondly, measurements from the flowmeter are not accurate and have a noisy response. Lastly, it could be attributed to combined effect of both reasons stated above. In order to check second reason, stated above, Figure 3.3 shows histogram of pump RPM for condition 2400 RPM and 15 gpm (blue) overlapped with histogram of flowrate 15 gpm (red). It can be observed from the Figure 3.3 that of course change and RPM contributes to the flowrate distribution but there is also flow distribution around each RPM column and which can be attributed to measurement noise. Thus, it can be deduced that change in flowrate is due to both RPM and measurement noise of the flowmeter. Although the behavior of only few conditions was presented here, this behavior was observed across all the conditions.

3.2.3 Vibration sensor data:

Before carrying out final experiments vibration sensors were tested for different sampling rates, it was observed that 1600 Hz sampling rate was less noisy than 3200 Hz. Moreover, based on the pump assembly and IIoT testbed it was unlikely to have need for

obtaining frequency information greater than at 800 Hz. Combining these two results it was decided to use sampling rate of 1600 Hz. The idea of using 800Hz sampling frequency was also discarded to avoid aliasing based in Nyquist frequency criteria[28],

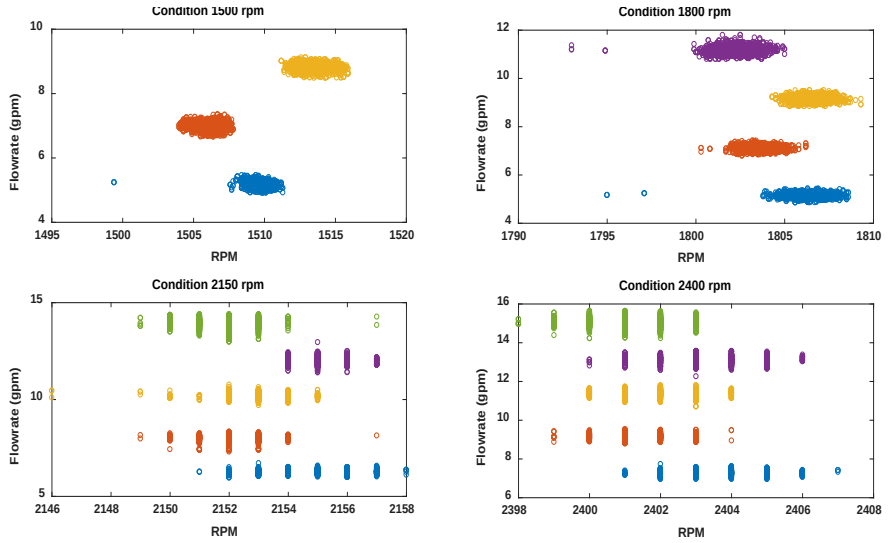


Figure 3.1 Measured RPM Data Distribution (Velocity, Veracity)

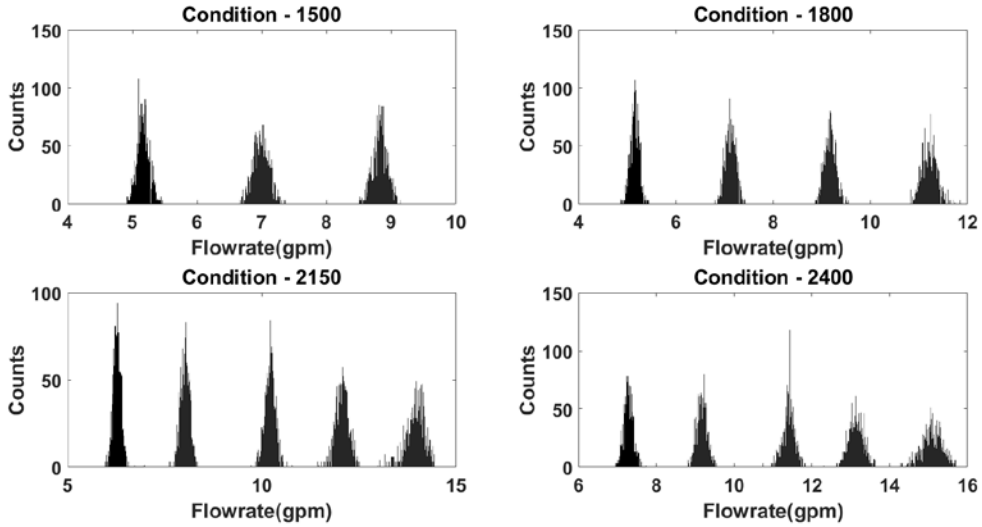


Figure 3.2 Measured Flowrate Data Distribution (histogram) (Velocity, Veracity)

[29]. Frequency of this data is much higher than any other data considered in the study. Figure 3.4 shows mean centered x, y & z direction signals collected from sensor-4

(sensor on coupling) for condition 2400 RPM and 7 gpm. Figure 3.4-(d) shows zoomed x-axis signal.

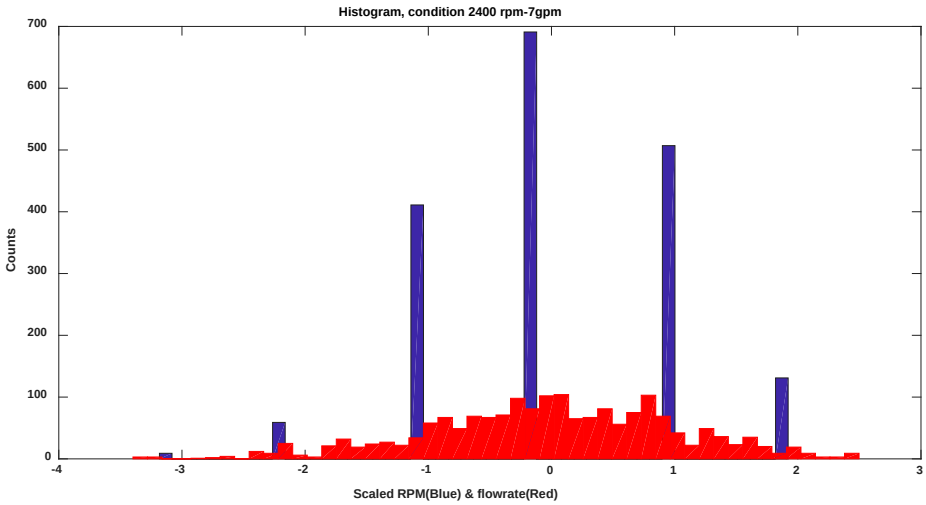


Figure 3.3 Histogram of Measured RPM & Flowrate Distribution (Scaled)

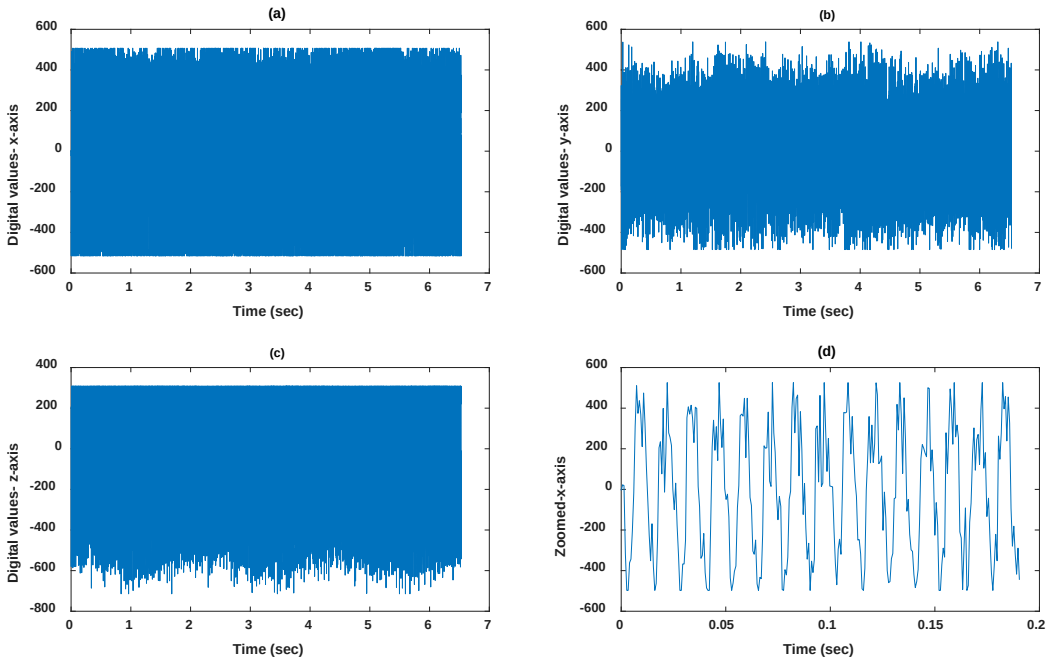


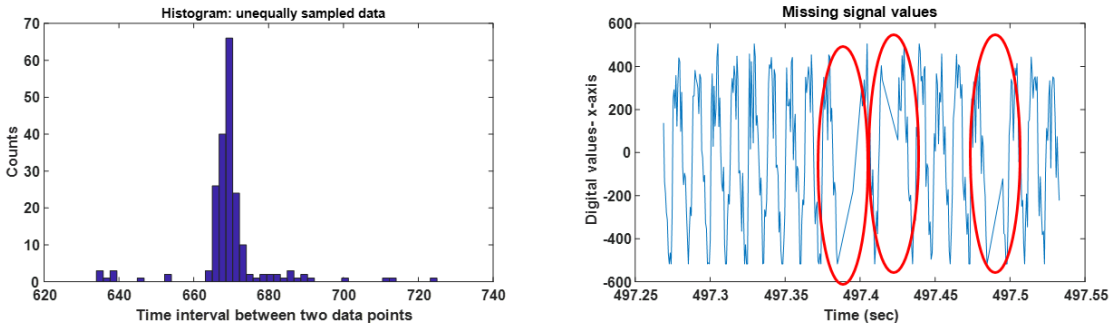
Figure 3.4 Signal from sensor-4 for condition 2400 RPM & 7 gpm. (Volume, Variety, Velocity, Veracity)

Some of the major IIoT sensor characteristics that were observed and can be generalized is discussed below along with specific problems they pose during analysis:

- Extremely noisy high frequency data: this makes data analysis difficult as original data is masked under random noise. Figure 13 clearly shows this characteristic.
- Unequally spaced real-time data: *i.e.* time between two consecutive readings is not constant and can be uncontrollable. Most well-established methods for modelling and data analysis used today requires samples to be equally spaces; especially for time series data. This also limits methods that can be used for modelling. Moreover, the chosen method needs to be fast enough, *i.e.* requiring less computation in order to get quick predictions and monitoring decisions. Figure 3.5-(a) shows representative histogram of sampling intervals between two consecutive data points obtained from sensor-4 for condition 2400 RPM & 7 gpm.
- Large sections of missing data: Large chunks of data go missing or do not get measured for a variety of reasons like sensor failure, lag in connection, traffic on network, varying network speed, *etc*. Pinpointing one source can be difficult and can be a separate research topic which isn't the scope of this study. Figure 3.5 (b) shows events of missing data for a part of signal from condition 2400 RPM and 7 gpm.
- Non-periodic and non-stationary signal: Figure 3.4 clearly shows that just like industrial mechanical rotating systems, the pump assembly used for this study also has non-stationary signal component which made it hard to observe periods in the signal.

The characteristics and issues discussed above with the data shows that the data under consideration has all the four V's of big data *i.e*. volume: ~70 GB of data was collected

and processed; variety: data from different sensors & different locations was collected; velocity: combination of both low & high frequency data needs to be modelled with



(a) **(b)**

Figure 3.5(a) Histogram of time between two consecutive data points (veracity), (b) Missing data from the signal (Veracity)

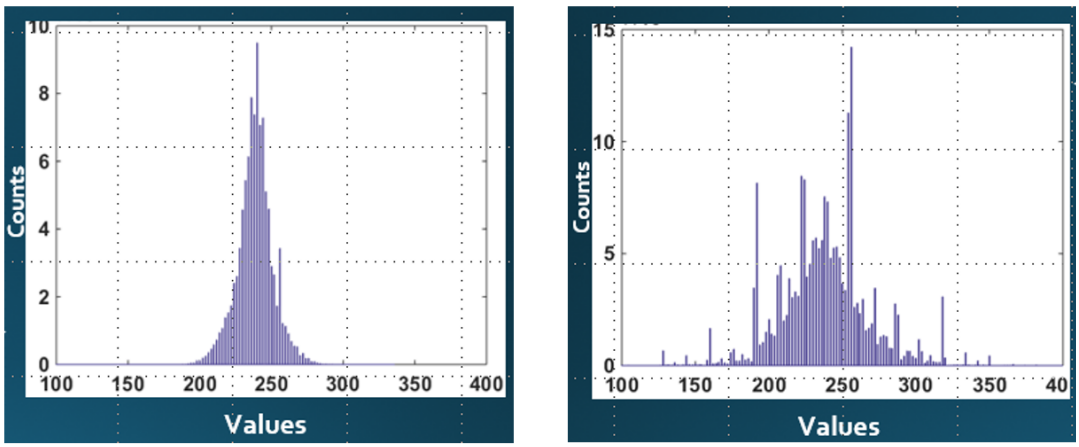
synchronization, & veracity: data collected have multiple layers of noise & different types of veracities like missing signals, unequally spaced, sensor noise, network noise, *etc*.

3.2.4 Static Noise Characteristics:

As stated before, wireless internet technology is fundamental to the concept of smart manufacturing. Therefore, the effect of using this technology on data collected will be of significant importance. To observe impact of wireless connection on data characteristics a simple experiment was designed. A vibration sensor was fixed on a non-mechanical stationary surface. Next, it was remotely accessed using wired internet and data collected was stored on that remote computer. Then, the same exercise was carried out using wireless internet on same sensor on same surface. Figure 3.6 shows histogram of the values collected under wired and wireless networks. As the surface was stationary and non-mechanical, it should ideally give similar Gaussian response for both cases (assuming data has Gaussian noise), however it can be clearly observed that data

collected wirelessly has an added layer of noise than the data collected with wired connection. Moreover, this added layer of noise is also Gaussian. This adds another layer of "veracity" in the data and will be of important consideration for building IIoT enabled SM soft sensors.

These results clearly suggest the importance of the maintaining quality of a wireless network while developing smart manufacturing technologies on a large scale. Moreover,



(a) wired connection **(b): wireless connection**

Figure 3.6 Static noise characteristics

the network's noise characteristic is case specific & needs to be tested for different applications in order to understand and mitigate its influence on the final data, model and framework developed. For our application it turns out intensity of added Gaussian noise characteristics didn't affect to a very large degree and thus reliable information extraction was possible for further framework development.

Chapter 4. Initial Modelling, Data Analysis & Feature Extraction

4.1 Modelling approaches:

In this work it was decided to build models using a linear modelling approach Partial least squares (PLS) and a non-linear modelling approach neural networks (NN). Main reason for choosing PLS is it is a does both dimension reduction as well as regression making it extremely robust against multi-collinearity and it is also commonly used for the scenarios when number of variables are very high [30]–[33]. For this initial modelling number of variables are high and no information about variable's importance was extracted.

Reason for choosing NN is it is considered as universal approximator [34]–[37] *i.e.* NN can model any function irrespective of its shape. Raw vibration signals are time series and to model time series recurrent neural network (RNN) [37]–[39] was used. However as the data considered was very large in size and in-order to mitigate computational complexities of standard RNN, a version of RNN was used called long-short term memory neural network (LSTM)[40], [41].

Some of the important details of PLS, NN, RNN & LSTM are discussed below:

4.1.1 Partial least squares (PLS):

Partial least squares (PLS) regression is a well-known and well established approach for regressing independently measured variables which are highly correlated, have high measurement noise and have high dimensionality. Crudely, PLS is called partial least

squares (& not ordinary least squares (OLS)) because it first extracts orthogonal variables which are called PLS components and then OLS is carried out on these PLS component variables and not on the entire initial dataset. PLS component are extracted such that each of them captures maximum variance in X matrix (original dependent variables) while capturing maximum variance in Y, regressand. Many algorithms have been proposed to carry out PLS two most used are 1) Nonlinear-iterative partial least squares (NIPALS) developed by Wold et al. [42], [43] 2) SIMPLS developed by Jong [[44]]. In this study NIPALS algorithm, briefly reviewed below, was used to build PLS models however SIMPLS can also be used without any changes to the approach. More details about the algorithms and its properties can be found in [[30], [33], [45], [46]].

Let X $\in \mathbb{R}^{n\,X\,m}$ and Y $\in \mathbb{R}^{n\,X\,m1}$, *i.e*. matrix of independent variables and dependent variables respectively. Where n = number of samples, m= number of independent variables and m1= number of dependent variables. If X and Y have linear relationship it can be written as:

$$\mathbf{Y} = \mathbf{XB} + \mathbf{V} \qquad (4.1)$$

Where B = regression coefficients and V = Error or noise matrix with corresponding dimensions. As discussed, PLS components from both X & Y are extracted in an iterative manner such that both X & Y can be represented as below:

$$\boldsymbol{X} = \boldsymbol{TP^t} + \boldsymbol{E} \qquad (4.2)$$

$$\boldsymbol{Y} = \boldsymbol{UQ^t} + \boldsymbol{F} \qquad (4.3)$$

Here T $\in \mathbb{R}^{n\,X\,p}$and U $\in \mathbb{R}^{n\,X\,p}$are the score matrix, P $\in \mathbb{R}^{n\,X\,p}$ & Q $\in \mathbb{R}^{n\,X\,p}$ are the loading matrices of X & Y respectively; where p= number of PLS component

selected for model building. E ∈ $\mathbb{R}^{n\ X\ m}$ & F ∈ $\mathbb{R}^{n\ X\ m1}$ are the error matrices of X & Y generally captures noise behavior in the data. Relation between X & Y is established using equation

$$\boldsymbol{U = TB} \tag{4.4}$$

Finally for estimation of Y equation 5.5 is used, in which F is minimized

$$\boldsymbol{\hat{Y} = TBQ^t + F} \tag{4.5}$$

Figure 4.1 shows schematics of PLS algorithm along with its objective function & constraints.

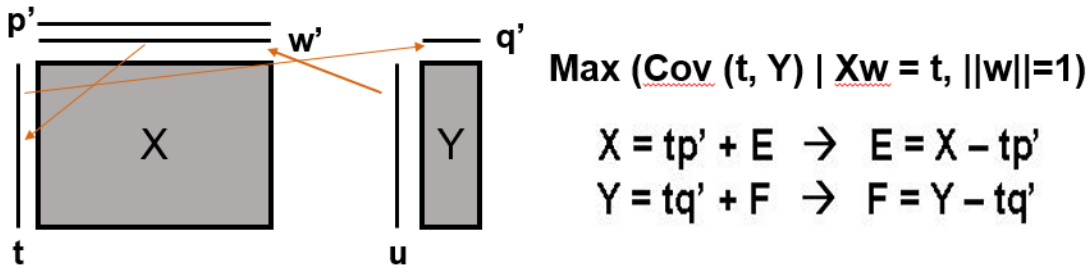


Figure 4.1 Schematics of PLS algorithm

4.1.2 Neural Network (NN):

Neural network models were built with the idea of mimicking human brains. Every time human brain thinks a set of neurons in head lights up. Process of fixing which neurons gets fired on receiving particular type of input is termed as process of learning. Similarly NN model has predefined type and number of neurons and based on fixed set of learning rules data is passed through the network. This allows network to learn and assign a numeric value to each neuron. These values are then combined resulting in a final

mathematical model. NN as a whole is a huge topic in itself and is still hot topic of research.

Four main component of NN model discussed briefly below are neurons, connections & weights, activation function & learning rule. More details about NN can be found in [37].

1. Neurons: All NN consists of input neuron, output neuron & hidden neuron. These neurons are staked together to model different kinds of dataset. Input neurons take raw data as their input process this data based on its nature and pass the updated value to the hidden layers or output layer (in-case there is no hidden layer). Output layer gives final output of the model. Hidden layers combined with activation function enables NN to model complex functions. Without them NN reduces to linear models and its output is simply weight average of input values.
2. Connections & weights: There are several different architecture of neural network models but for most general cases all the neuron on the previous layers are connected with all the neurons of next layers. Each connection is assigned with the weight value learnt during training process.
3. Activation functions: These are the functions which decides state of a neuron in a particular network *i.e.* either it is fired or not fired and if fired to what degree. State of neurons in turn indicates importance of that neuron in the model. Activation function plays major role in imparting NN its ability to capture non-linearity and enables it to be universal approximator. There are several different activation functions suggested in the literature and are used widely [37], [47].

Choice of activation depends on problem at hand and model training and prediction performance. More details about activation functions can be referred from [37], [48].

4. Learning rule: This is the function which decides how a model will learn. This function modifies weights of the models during training stage based on the performance of objective function. Generally objective functions are minimized or maximized and thus generally gradient descent is used in order to reach optimal objective function point. However other learning rules have been proposed as well [37], [49], [50]. Selection of learning rule again depends on problem at hand however for most problems stochastic gradient descent [37], [51] or mini-batch gradient descent [37] approach is used find global minimum.

Figure 4.2 shows schematic of a simple feedforward neural network, where data only moves forward & doesn't recycle. NN doesn't assume anything regarding data and learns data patterns by combination of search and mathematics. In most NN flow of information is always forward to the next layers and thus are also called feed-forward NN.

4.1.3 Recurrent neural network (RNN):

NN have found its extremely successful use for numerous applications in clustering, computer vision, classification, prediction *etc*. Further NN have been applied for all different types of data e.g. image, discrete, continuous, video, binary, time-series *etc*. Several different architectures have been proposed for each type of data. For this study as raw vibration data is a time-series it was decided to use recurrent neural network (RNN), a time-series capturing variant of NN, for modelling this data.

RNNs have internal memory that enables them to remember inputs they received making them ideal for sequential data like time series. In RNNs the information loops

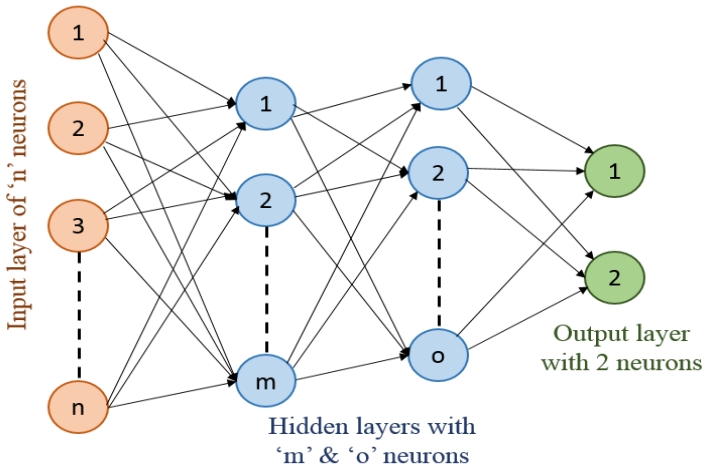


Figure 4.2 Feedforwad Neural Network

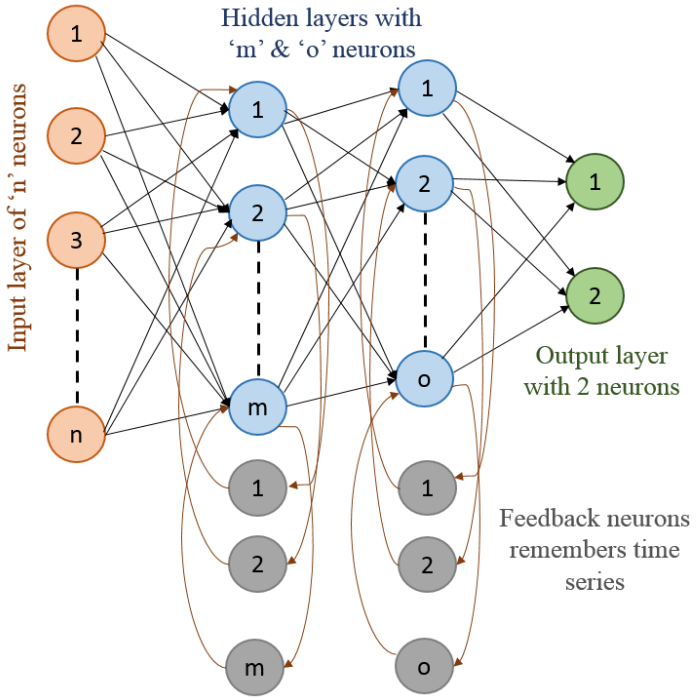


Figure 4.3 Schematics of Recurrent Neural Network

through each neuron *i.e.* each neuron takes current input along with the learnt parts of previous inputs. Another difference between feed-forward NN and RNN is in learning

rule in RNN weights of current as well as previous inputs are updated contributing to its ability to capture temporal patterns. Two main issues of standard RNN are 1) exploding gradients; during model training algorithm assigns extremely high values of weights

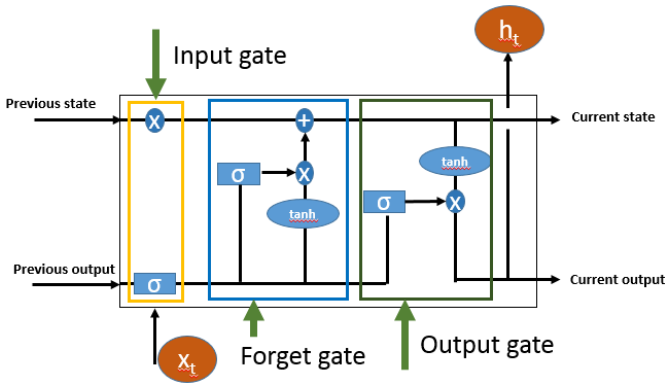


Figure 4.4 Building block of LSTM neuron

without having need for it, 2) vanishing gradients; during model training values of gradients or slope for moving in a particular direction becomes extremely low because of which model loses its ability to learn. Moreover RNNs have only short-term memory usually only up to one previous step therefore RNNs becomes of little use for modelling in this study. However a concept of long short term memory (LSTM) solved these major problems of RNN and was used for initial modelling for the IIoT testbed data. Figure 4.3 shows schematics of standard RNN. There are many more details about each steps of RNNs and interested readers and refer to [37], [40], [41].

4.1.4 Long-Short Term Memory (LSTM):

LSTMs networks are an improvement over RNNs. They have long term memory and thus can be useful for identifying long term patterns. It can read, write and delete information from its memory. Fundamental difference between LSTM & RNN is neurons' building unit itself. Memory can be seen as a gated cell which decides whether

to keep or delete the information based on its importance. Importance of the information is decided by the weights which are learnt as well. Figure 4.4 shows building unit of LSTM neuron, entire time series is passed through this block to generate outputs for each time step or final time step based on requirement. LSTMs neuron has three gates: input gate (determines if any information should be let in), forget gate (determines if stored information should be deleted or not) & output gate (determines if impact of new information should be made on output or not). LSTM gates are in the form of sigmoid *i.e.* between 0 & 1. Problem of exploding & vanishing gradients are solved by LSTMs and generally has relatively less training time. More details about LSTMs can be found in [37], [40].

4.1.5 Neural network training, hyper-parameter & tuning:

For training a neural network first of all a structure or graph of NN is decided, next NN is initialized with random weights, next model calibration data is passed through this NN, each neuron gets its expression value and at the end of the network desired property is calculated. Process of leading up to this stage is called forward pass. Next calculated values are compared with true measured values and an average error value is calculated. This average error values are calculated based on user-defined functions called loss function or cost function. Different cost functions have been suggested in literature [37]. Main objective of loss function is to give representation of model performance. Next is the learning rule or weight update steps generally this is achieved using gradient descent gradient of loss function is calculated with respect to all the weights and biases. This process of updating all the weights is called backpropogation. Fundamentally backpropogation is about understanding how cost function changes for unit change in

weights & biases of the NN. More details about backpropogation & equations related to it can be found in [37]. Weights and biases are updated and again calibration data is passed through the updated network again cost function is calculated and backpropogation is carried out. This cycle is stopped either when desired performance is achieved or when model stopped learning.

Before training any NNs there are many hyper-parameters which needs fixed at the design stage itself, some of the major parameters which were fixed in this study are briefly described below:

- Number of epochs: when all the model calibration data is passed through the network once it is called one epoch. More the number of epochs more the weights are tuned for calibration data. This parameter needs to be tuned as very large number of epochs may lead or over fitting while lower number results into under-fitting. In this study problem of overfitting was avoided by having separate validation set and model which performed best on validation set was used test model's prediction performance. Problem of under fitting was avoided by keeping higher value for number of epoch.
- Number of hidden layers & number of neurons in each hidden layer: This two hyper-parameters are fixed by trial and error. Although there have been few algorithms which have been proposed for automatically identifying them however for most application they are still manually tuned based in experience. In this study they were tuned by trial & error.
- Batch size: One of the draw-back of single batch gradient descent is that weights are updated based on average cost function value which can only be obtained

after all the samples are passed through the network. Thus for every weight update one has to cycle through all the samples and when number of samples are large (big data) this can become time prohibitive. This problem is mitigate by use of mini-batch gradient descent, where entire data is divided into mini batches of data and weights are updated at the end of each mini-batch. It is found that mini-batch gradient descent converges quickly and give similar overall performance as single batch gradient descent. Batch size is optimized to for getting best performance with lesser time.

- Network weight initialization: NN learning starts from random assignment of weights. However some process knowledge is available then they can assigned from user defined distribution. Generally and for this study they are assigned from uniform distribution.
- Other hyper-parameters: there are many other hyper parameters for training a NN like learning rate, decay rate of optimizer *etc*. were kept at its recommended values. More details [37], [52].

Major methods for identifying hyper parameters are manual search, grid search, random search & Bayesian optimization [37], [53]. Each of the approach have its drawbacks and advantages. For this study manual search was used.

4.2 Initial modelling of IIoT testbed:

In order to build a predictive model raw data obtained needs to be cleaned and synchronized for modelling. As discussed before measurement rate of RPM & flowrate is 3Hz & that of vibration signals is 1600 Hz. Moreover in order to make identify any relationship one will have to use many vibration values and relate this extracted property

with corresponding property of interest. After analyzing data it was found that vibration data from sensor-4 was of most relevance for predictive model & was used for all models developed in this study, unless stated otherwise. Synchronization between different sampling frequencies was obtained using Unix epoch time (UET). First UET was identified for measured value of flowrate and RPM, say UET-f. Next the time point closest to this UET-f is identified from vibration sensor raw data, say UET-v. Next total of 800 data points around (400 before and 399 after) the data point collected at time UET-v were selected, these 800 points becomes features for identifying corresponding flowrate & RPM. As discussed before vibration sensor measures vibration in x, y & z directions and 800 data point in each direction becomes input to for model building. Therefore, for each value of flowrate & RPM, 800 values of vibration data point in each x, y z direction were compiled for model development. Moreover samples with missing data were removed as LSTM models can only take equidistant time-series. Data collected at all the conditions corresponding to RPM 1500, 1600, 1700, 1800, 1900, 2000, 2100, 2200, 2300, 2400 & 2500 was selected for model building. Total number of samples were divided into training and test set. Both the sets contains samples from all the condition however care was taken that samples in the test set were collected later in time than the samples in training set were selected. Such division is closer to real world application where model is applied on the real time data which is seen after the model building data is collected. 80 % of the total samples from each condition were included in training set and remaining in test set. Samples in training set are randomized and further divided into calibration (70% samples) & validation set (30 % samples). Total number of samples in

calibration set =48073, validation set = 20604 & test set = 16488. Performance index chosen to monitor model performance is root mean squared error (RMSE):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{Y} - Y)^2}{n}} \tag{4.6}$$

Where n= number of samples.

4.2.1 LSTM model for raw:

As seen before LSTM type RNN have ability to capture the time series. Expectation from the model is it will be able to identify underlying time series pattern and will further be able to establish function relating flowrate & RPM. The designed LSTM network/graph contains an input layer of size 800 X 3 for each sample, followed by two hidden layers of fully connected 200 & 100 LSTM units respectively, followed by one hidden layer of fully connected dense layer (*i.e*. layer of normal fully connected neurons) and output layer of 2 fully connected normal neuron each for RPM & flowrate. Number of epochs = 200, batch size=801 samples. Figure 4.5 shows prediction performance of pump RPM & Figure 4.6 shows prediction performance of pump flowrate using LSTM

Table 4.1 RMSE prediction values from LSTM model (raw data)

Sr. No.	RPM	Flowrate	RMSEp-RPM	RMSEp-flowrate
1	1500	5, 7, 9	669.6520	6.4749
2	1600	5, 7, 9	452.1746	4.2513
3	1700	5, 7, 9, 11	373.7830	3.9521
4	1800	5, 7, 9, 11	353.3748	2.7097
5	1900	6, 8, 10, 12	439.5101	1.8558
6	2000	5, 7, 9, 11, 13	398.6946	2.4441
7	2100	6, 8, 10, 12, 14	270.8760	3.4809
8	2200	6, 8, 10, 12, 14	186.3001	3.9368
9	2300	7, 9, 11, 13, 15	104.5452	4.6394
10	2400	7, 9, 11, 13, 15	89.6497	3.9997
11	2500	8, 10, 12, 14, 16	33.6917	5.0906
Overall			335.9889	3.9961

model with best validation performance. Table 4.1 shows RMSE prediction for all condition at different RPMs. RMSE cross validation (RMSEcv) for RPM = 330.3281 &

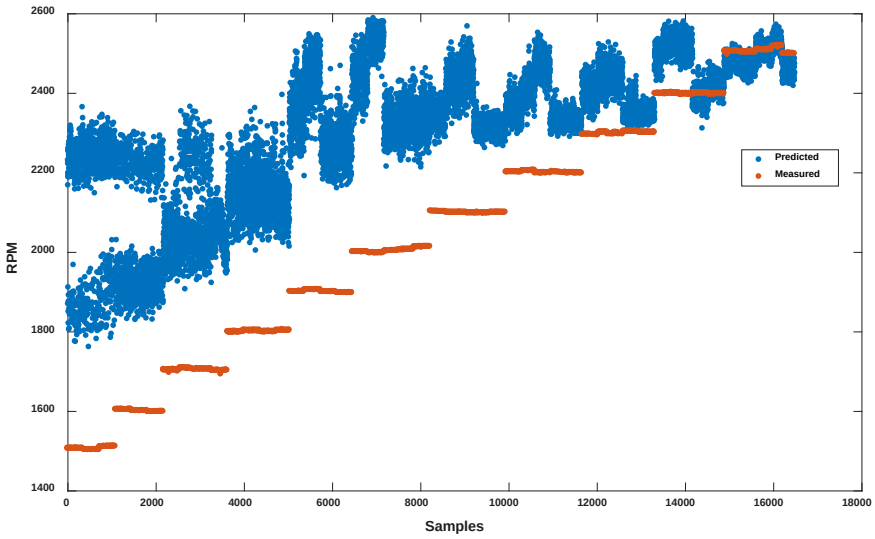


Figure 4.5 LSTM model Prediction performance for RPM

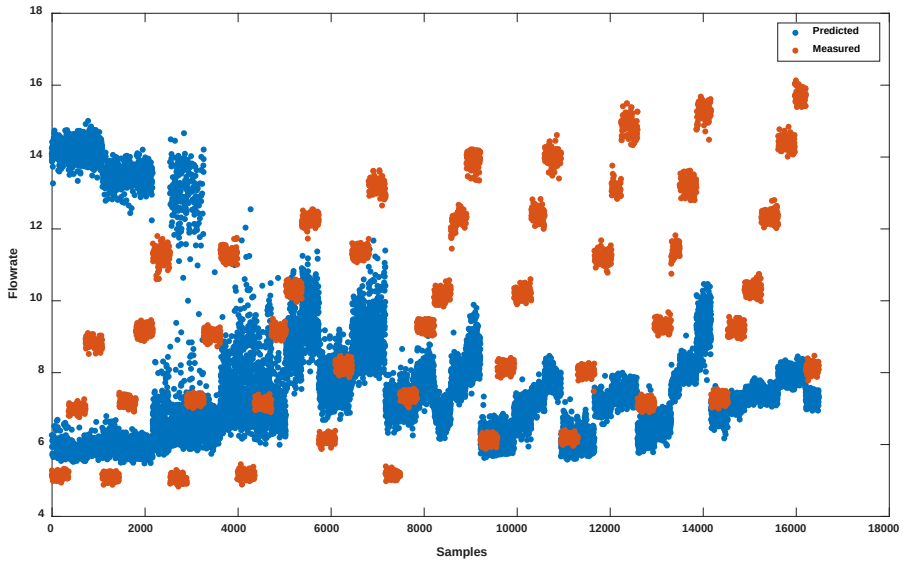


Figure 4.6 LSTM model Prediction performance for Flowrate

for flowrate = 4.0749

Figures clearly shows that model performance is extremely poor and cannot be used for any applications. Bad performance of this model can be attributed to high noise in the

data. Moreover even though samples with large number of chunks of missing data were removed before modelling data points are still unequally spaced which may act as noise in the LSTM structure as it needs equally spaced data. With Neural networks there is no guarantee and some different architecture of LSTM may give improved results however time required for training such models is prohibitively high moreover with such a multidimensional long time series there is reducing this time generally will be difficult. Apart from requirement of extremely costly computational resources it can also be deduced that rote application of machine learning or deep learning approaches on available data without any system knowledge may force users to not only select complex models but also may lead to misleading conclusions, like in this case one may conclude that no relationship exists. Thus it is imperative to include system knowledge and identify how engineering/science principals related to the system or process being modelled by data. Moreover such knowledge will further filter the big data and improve model performance.

4.2.2 PLS model for raw data:

It is obvious that linear PLS model won't be able to capture necessary information for good prediction performance as raw vibration data is non-linear. However, PLS modelling for raw vibration time series data was merely carried out for comparison completeness. Here X-matrix of LSTM model was unfolded to form matrix of size N X 2400 where N is number of samples in each data set. *i.e*. 800 data points from each direction was stacked column wise. Thus each data points will act as a variable. PLS can handle correlated data very efficiently by extracting orthogonal variables. Number of

orthogonal variables to be included in the model was optimized using validation set, details discussed before. Separate models were built for RPM & flowrate as it was

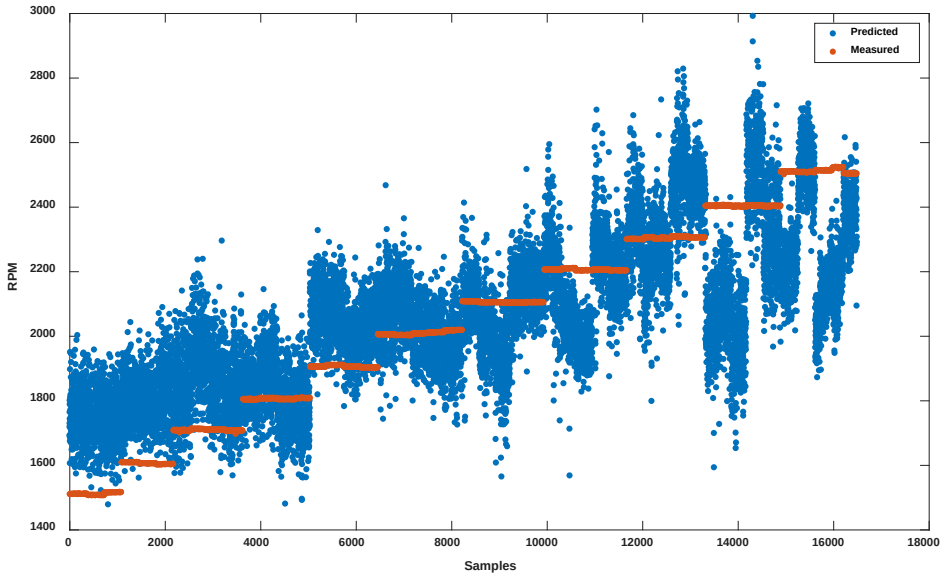


Figure 4.7 PLS model Prediction performance for RPM (raw data)

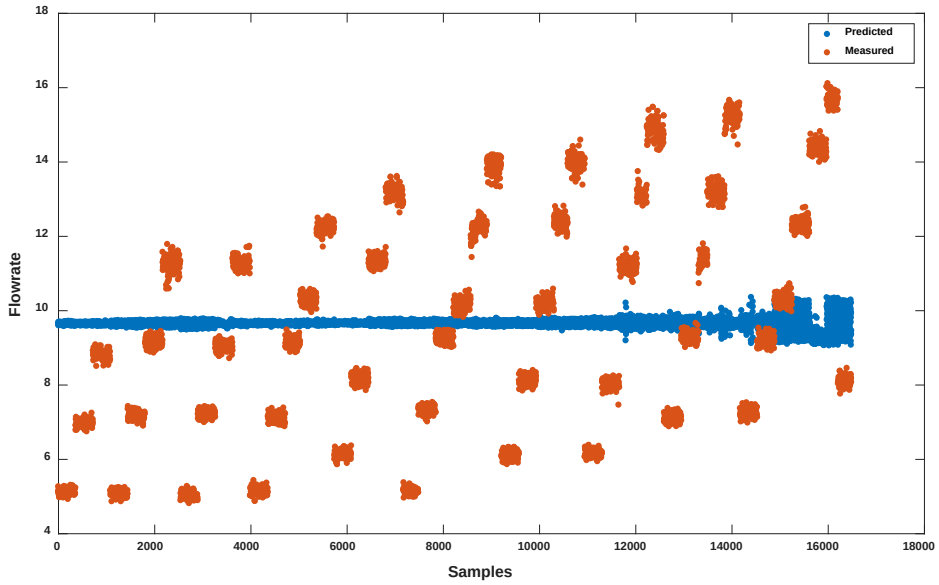


Figure 4.8 PLS model Prediction performance for Flowrate (raw data)

observed that individual models in general performed better than combined. Figure 4.7 & Figure 4.8 shows RPM & flowrate prediction performance for their respective models.

Table 4.2 shows RMSE prediction for all condition at different RPMs. For RPM model RMSEcv = 205.2732 & PC selected = 8, for flowrate model RMSEcv=2.9692 & PC selected = 1. Results for PLS model were extremely poor as expected. At this point based on the results of raw data it is clear that raw vibration data cannot be used directly for prediction of desired property. Moreover data needs to be processed somehow in order to establish if the collected data has enough information that can be extracted modelled

Table 4.2 RMSE prediction values from individual PLS model (raw data)

Sr. No.	RPM	Flowrate	RMSEp-RPM	RMSEp-flowrate
1	1500	5, 7, 9	253.9196	3.0637
2	1600	5, 7, 9	220.7506	3.0097
3	1700	5, 7, 9, 11	193.3502	2.7451
4	1800	5, 7, 9, 11	97.3151	2.7825
5	1900	6, 8, 10, 12	162.7069	2.3329
6	2000	5, 7, 9, 11, 13	100.5365	2.7992
7	2100	6, 8, 10, 12, 14	123.9796	2.7675
8	2200	6, 8, 10, 12, 14	156.4336	2.8737
9	2300	7, 9, 11, 13, 15	158.7514	3.0588
10	2400	7, 9, 11, 13, 15	303.6994	3.3019
11	2500	8, 10, 12, 14, 16	286.8554	3.6691
Overall			195.9700	2.9654

while overcoming big data challenges.

4.3 Feature extraction using frequency analysis:

Vibration signal obtained from the testbed is extremely complicated, there are no clear identifiable periods and also there is possibility that signal is non-stationary. In order to identify degree of non-stationarity and major components of the signal it was decided to use frequency domain analysis of the signal. Generally for equally spaced signals frequency domain analysis is carried out by Fourier transform and FFT is the

most common algorithms which is used for the same. However when the data points are unequally spaced FFT is fails and cannot give reliable frequency spectrum.

To address this issue three approaches that were considered are discussed below:

- Interpolation: In this approach, signal is interpolated at equally spaced fixed instances in time using unequally spaced values around that time [54], [55]. Next, fft is carried out on interpolated signal to obtain frequency spectrum of the signal. Signal can be interpolated using different approaches like simple linear interpolation, pchip (piecewise cubic hermite interpolation polynomial) interpolation, spline interpolation (cubic spline) *etc*. generally spline and linear interpolation techniques are used for cyclic signals. Figure 4.9 shows comparison of three interpolation approaches on a randomly chosen vibration signal. From the Figure 4.9 it seems any of the approach are suitable for the application. But on closer observation it was observed that none of the approach gave acceptable results for when there were gaps in the data. Figure 4.10 shows comparison of interpolation approaches for missing data section. This result highlights one of the most important and inherent problem with interpolation techniques *i.e*. Interpolated signals are not reliable and may not show true representation of actual signal. This issue is more likely to create as signals gets more complicated. Furthermore interpolation also causes loss of information as signal is resampled at some fixed time intervals.
- Signal binning: In this approach, a time interval is fixed. Next signal values that falls within that time interval is replaced by a value representative of that interval, often central value is used [56], [57]. Once this new binned signal is

obtained, fft is carried out to obtain frequency spectrum. Just like interpolation binning has many disadvantages. It will reduce the sampling rate of the signal and thus Nyquist frequency is also reduced. It will cause loss of information as only representative value is chosen. Thus for complex signals like signal from

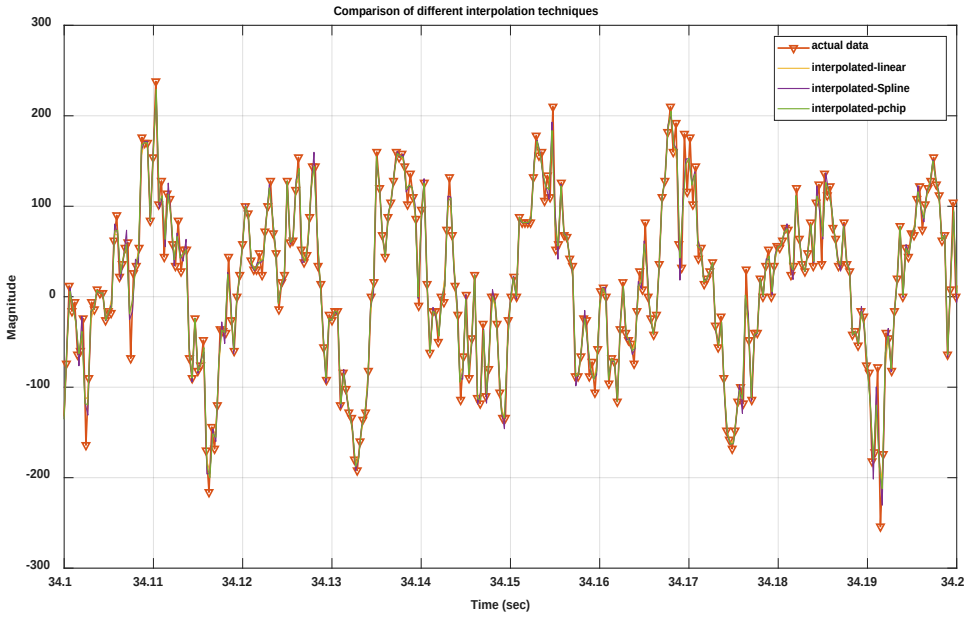


Figure 4.9 Interpolation Approaches Comparison

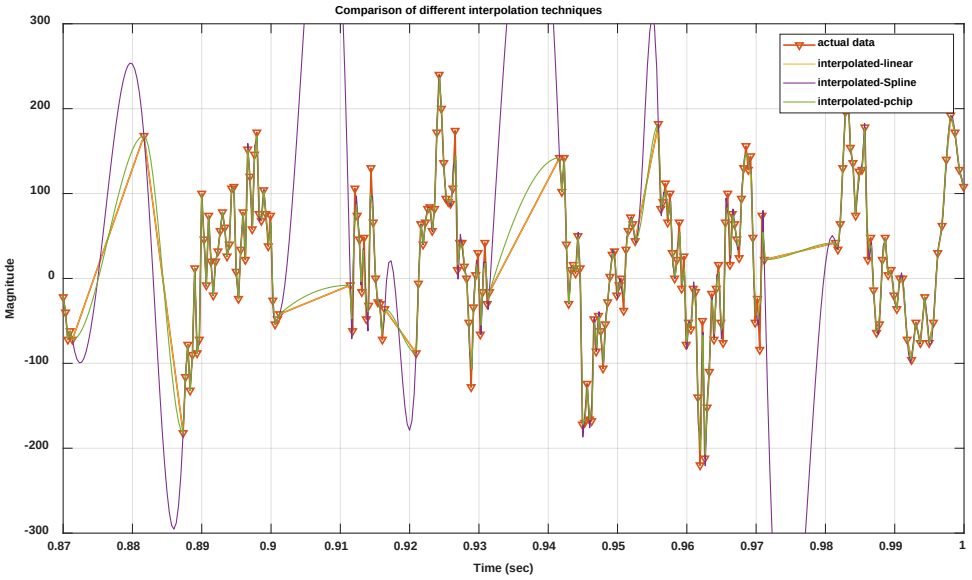


Figure 4.10 Interpolation approaches comparison during missing signal

pump assembly it might lose higher frequencies information resulting into incorrect or unusable spectrum.

- Lomb's algorithm: This approach doesn't required signal to be equally spaced to extract power spectrum density. It a robust approach that uses the measured signal values directly and therefore is immune to information corruption and reduced sampling rate. Because of these characteristics for this study it was decided to use Lomb's algorithm for deriving frequency spectrum from the vibration signal. Brief review of Lomb's algorithm is given below.

4.3.1 Lomb's algorithm review:

Output of this algorithm is power spectrum density of the signal under consideration. It doesn't require samples to be equally spaced and when they are equally spaced mathematics of the algorithm reduces to Fourier transform [58], [59]. This approach is subset of least-squares spectral analysis and algorithm is widely used in astronomy community. This approach assumes signal as a function (P) of amplitudes of cos and sine function, frequencies and time of sampling *i.e.* P (a, b, f, t). Function P is assumed to have sine and cosine parts to it and is written as:

$$\mathbf{P}(\mathbf{a},\mathbf{b},\mathbf{f},\mathbf{t}) = \mathbf{a} * \mathbf{cos}(\mathbf{2} * \mathbf{pi} * \mathbf{f} * \mathbf{t}) + \mathbf{b} * \mathbf{sin}(\mathbf{2} * \mathbf{pi} * \mathbf{f} * \mathbf{t}) \qquad (4.7)$$

Function P is then fitted on the data or signal under consideration using least square approach such that cos and sin terms are made orthogonal to each other at sample time t. *i.e*. cross cos-sin terms in the partial derivative terms are made zero. Cos and sin terms are made orthogonal by shifting the signal in time and identifying delay shift τ (offset). This shift τ is chosen to guarantee the time invariance of the computed spectrum. Any shift in time measurement results in an identical shift in the offset. Moreover, choice of τ ensures that a

maximum in the periodogram occurs at the same frequency which minimizes the sum of squares of the residuals of the fit of a sine wave to the data [59], [60]. Two equation that identifies τ and power spectrum $S = \sum_{i=1}^{N} P^2$ are [58], [59]:

$$\boldsymbol{\tau = \frac{1}{2 * (\omega)} (arc \tan(\frac{\sum_{i=1}^{N} \sin(2 * (\omega) * t_i)}{\sum_{i=1}^{N} \cos(2 * (\omega) * t_i)})} \quad \textbf{(4.8)}$$

$$\boldsymbol{S(\omega) = \frac{1}{2 * \sigma^2} \left(\frac{\left[\sum_{i=1}^{N} X_i * \cos(\omega * (t_i - \tau))\right]^2}{\sum_{i=1}^{N} cos^2(\omega * (t_i - \tau))} + \frac{\left[\sum_{i=1}^{N} X_i * \sin(\omega * (t_i - \tau))\right]^2}{\sum_{i=1}^{N} sin^2(\omega * (t_i - \tau))} \right)} \quad \textbf{(4.9)}$$

Where σ= variance of the signal, N= number of observations, ω=2*pi*f and X=signal values

For complete derivation and more mathematical details please refer [58], [59], [61]. For the above version of Lomb's algorithm signal X should be mean centered. If there is some ambiguity about mean of the signal generalized Lomb's algorithm can be used [[62], [63]]. In this study signal will be mean centered and Lomb's algorithm stated above will be used. Major advantages of this approach are conversion of signal into its equal spaced form is not required. Periodogram thus obtained can be converted to frequency spectrum by linear transformation of square-root of power spectrum, based on sampling frequency. As signal is unequally spaced value of Nyquist frequency is very high and thus problem of aliasing shouldn't pose any serious problems [61].

4.3.2 Signal pre-processing and feature extraction:

End goal of retrieving spectrum information is to model it for RPM & flowrate prediction. Therefore it was decided to retrieve spectrums corresponding to the measured RPM & flowrate values. Just as in the case of raw vibration data modelling, unix epoch time (UET-f) in micro secs corresponding to measure flowrate 7 RPM value is identified. Next time closest to UET-f is searched for in the raw vibration signal data (UET-v) and raw data valued is identified. This identified value forms mid-value of 801 data points, *i.e*. 400 data points before mid-value & 400 data points after mid-value, that are used for extracting reliable frequency spectrum using Lomb's algorithm. In absence of large missing sections of data overall time interval for getting one spectrum is around 0.55 sec. This will cause some overlapping of data points among different segments but as vibration signal was collected at around fixed value of flowrate and RPM spectrum resulting spectrum will give information related to that corresponding value of flowrate & RPM.

Because of the above design each value of flowrate & RPM will have corresponding frequency spectrum. It is to be noted that for all samples before extracting the frequency spectrum using Lomb's algorithm time values were forced to start from 0. Moreover to reduce spectral leakage and obtain smoother spectrum, mean-centered signal is passed through a window function [15], [64]–[67]. In this study Hann window function was used. Data points are feed to algorithm and power values, for frequencies from 1 to 800 Hz with resolution of 0.2 is obtained. This power spectrum is denoted as S. Power spectrum is converted to normal frequency spectrum using following transformation $Amp = cf * sqrt\left(\frac{S*f_S*2}{N}\right)$ where cf = correction factor for using window function (2 for Hann window [66], [67]), S=power spectrum, f_s=sampling frequency and N= number of

data points used for getting spectrum. Figure 4.11 show spectrums of 500 randomly selected samples corresponding to different conditions. Optimization of frequency resolution, range of frequencies to be included for modelling and choice of specific window function for reducing spectral leakage was carried out based on combination of thorough search and working system knowledge of the testbed.

4.4 Model development with primary system knowledge:

Primary system knowledge about testbed and vibration signals provides more understanding about the underlying process such as which frequencies has larger amplitudes *etc*. on closer observation of Figure 4.11 it can be observed that there are different patterns corresponding to different conditions. When samples from all the different conditions are observed together the data underlying is extremely complex and thus it was decided to use a modelling approach extremely successful for learning and

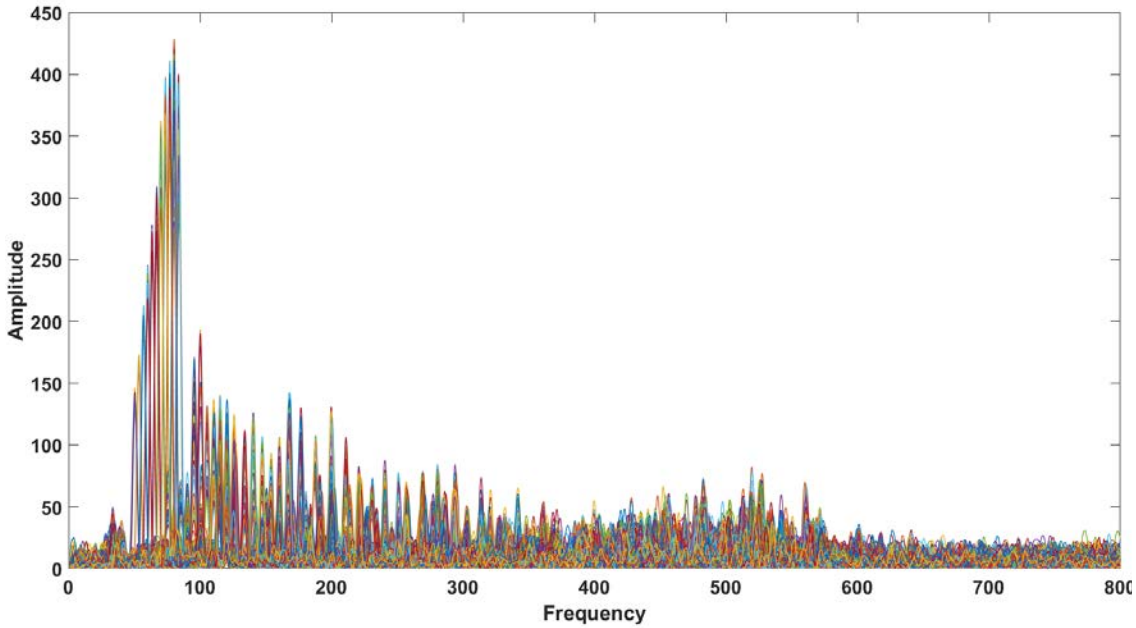


Figure 4.11 Spectrum using Lomb's Algorithm

remembering non-linear complex patterns *i.e*. deep neural network. NN model was build

using entire spectrum from all three direction. On combing spectrum from all three directions number of variables involved in modelling is very large (~12000), as PLS is known to be robust against high dimensional data and also carries out dimensionality reduction it was deiced to build PLS model as well to compare its performance with NN and to check if performance improvement by using complex, resource intensive method like NN is justified. PLS model developed at this stage is called full PLS model in this study.

For NN model & full-PLS model, data collected at all flowrate condition for RPM conditions from 1500 to 2500 at 100 RPM intervals was included for training & testing. Here instead of using raw vibration signals frequency spectrum extracted from the signals were used. Frequency spectrum from x, y & z direction is stacked column wise to form one row (sample) of the X matrix. For each direction frequency spectrum was obtained at frequencies from 1 to 800 Hz with resolution of 0.2 Hz, *i.e*. amplitude value at 3996 frequencies, thus combining x, y & z direction results in matrix with total 11988 variables. To mimic real world scenario last 20 % of chronologically collected samples of all the conditions were included in test set. First 80 % of chronologically collected samples, of the conditions mentioned above, are included in training set. The training set was then randomized and separated into validation set & calibration set. For conditions evident outliers were removed manually from the test set. Number of sample in calibration set=48073 X 11988, validation set = 20604 X11988 & test set = 16488 X 11988.

4.4.1 Full partial least squares (PLS) model:

This model is named as full PLS model because samples from all the conditions were simultaneously included for modelling. Two separate PLS models were decided to be built for RPM & flowrate. Number of principal components to be included in the model was decided based on model validation performance. Calibration, validation & test datasets are same as described in above section. Here data was mean-centered for model building. Figure 4.12 & Figure 4.13 shows RPM & flowrate prediction performance of full PLS model using frequency spectrum. RMSE prediction values are given in Table 4.3 for RPM model RMSEcv = 12.9111 & principal components selected=29, for flowrate model RMSEcv = 0.7425 & PC selected = 13. However this performance is still not satisfactory at all for any real world application and another modelling approach or more stringent feature extraction needs to be carried out.

Even though full-PLS model's performance is poor it is much better than LSTM model & PLS model with raw data. It can be deduced that there is improved relationship between extracted feature frequencies' amplitude and RPM & flowrate than with raw time series data and RPM & flowrate. The poor full PLS model performance is understandable and expected. One of the reason for this improved relationship can be attributed to relationship between spectrum of the vibration signal from the pump & pipes and water flow inside the pipe and/or different mechanical vibrations originating from several different moving parts of the pump. It is well known as RPM changes vibrating frequencies of associated with different parts also changes which in turn shifts peaks generating from different pump components. For example suppose a peak corresponding shaft bearing is observed at 500 Hz for pump running at 1500 RPM. Now if pump RPM is

increased or decreased than frequency at which this particular peak arises will shift as well. Therefore on staking samples/spectrum of different RPM conditions row wise cause's

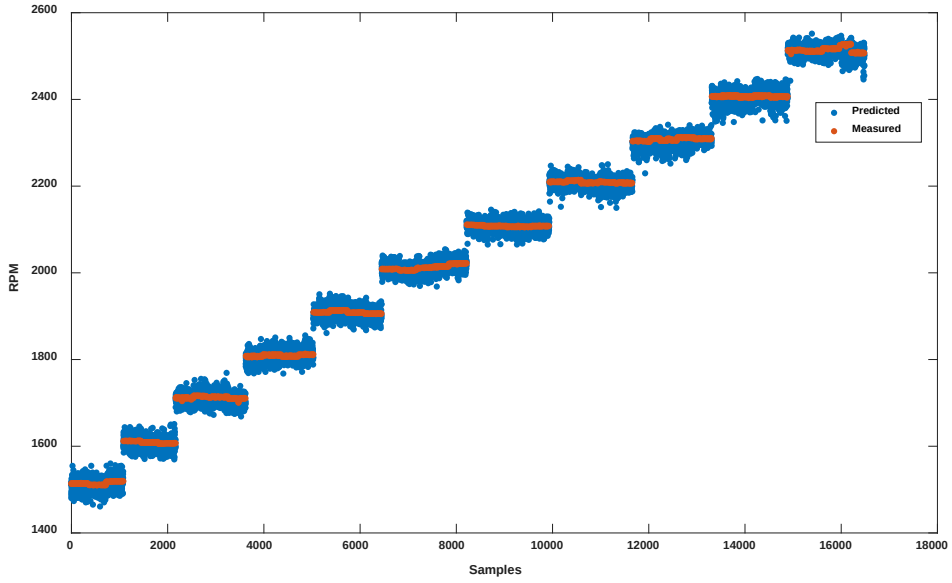


Figure 4.12 Full PLS model RPM Prediction performance (Spectrum)

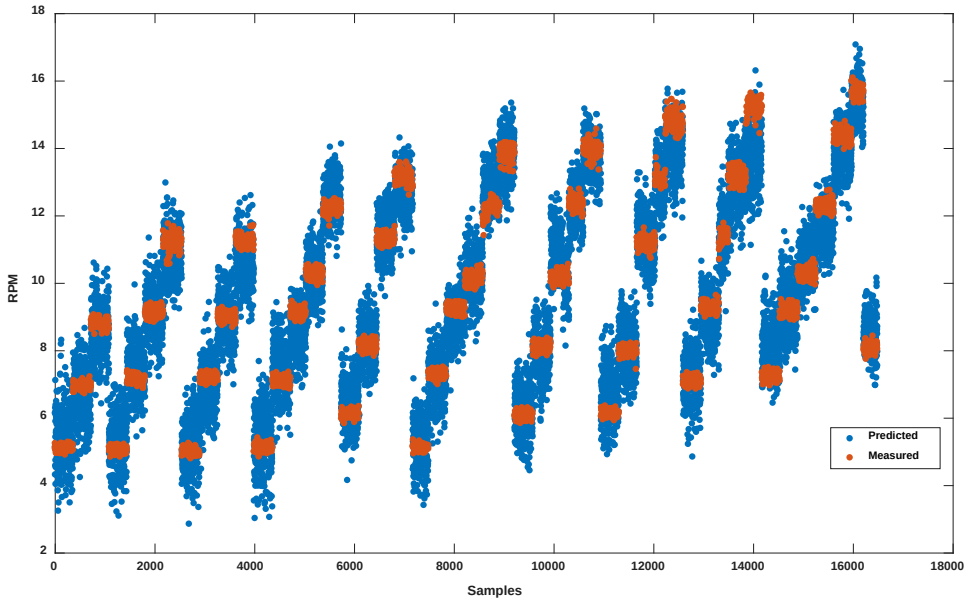


Figure 4.13 Full PLS model Flowrate Prediction performance (Spectrum)

frequencies from different sources to overlap. Same analogy can also be applied for flowrate in the pipe *i.e*. as flowrate in the pipe changes vibrations frequencies may shift in an

unexpected manner. This poses a serious problem for flowrate prediction as on pump system level impact of load/flowrate on intensity of vibrations from different rotating parts of the pump is modelled and overlap of frequencies means overlap of sources. This makes extraction of relevant information (identification of intensity of a fixed vibration source) extremely difficult for linear modelling approach like PLS. It is also possible that inherently multivariate relationship between intensity of vibration sources & flowrate is non-linear.

Table 4.3 RMSE prediction values from individual PLS model (spectrum)

Sr. No.	RPM	Flowrate	RMSEp-RPM	RMSEp-flowrate
1	1500	5, 7, 9	14.2223	0.8005
2	1600	5, 7, 9	13.0963	0.7224
3	1700	5, 7, 9, 11	12.9760	0.7481
4	1800	5, 7, 9, 11	13.9338	0.8432
5	1900	6, 8, 10, 12	12.9654	0.6458
6	2000	5, 7, 9, 11, 13	11.8840	0.6601
7	2100	6, 8, 10, 12, 14	11.9783	0.7084
8	2200	6, 8, 10, 12, 14	11.8524	0.7585
9	2300	7, 9, 11, 13, 15	13.4272	0.8039
10	2400	7, 9, 11, 13, 15	15.7561	1.1005
11	2500	8, 10, 12, 14, 16	12.9700	0.7462
Overall			13.1675	0.7850

Apart from that X-matrix also contains very noisy irrelevant information making modelling more challenging. Thus to extract such complex relationships between flowrate and vibrations by learning spectrum patterns without any knowledge about the underlying process fundamentals and any knowledge based treatment to the data it was design a complex deep NN model.

4.4.2 Deep neural network model:

As discussed before NN are known as universal approximator *i.e.* it can model any possible function and learn complex patterns. Thus it was decided to design a deep neural network for simultaneous prediction of flowrate & RPM. Hyper parameter of the model

were tuned by trial & error. Designed NN has 4 hidden layer neural network with 4000, 2000, 1000 & 300 neurons respectively, it also has input layer of 11988 neurons and output layer of two neurons each corresponding to RPM & flowrate respectively.

Table 4.4 RMSE prediction values from individual Deep NN model (spectrum)

Sr. No.	RPM	Flowrate	RMSEp-RPM	RMSEp-flowrate
1	1500	5, 7, 9	19.0692	0.2228
2	1600	5, 7, 9	16.4467	0.2686
3	1700	5, 7, 9, 11	10.1307	0.2384
4	1800	5, 7, 9, 11	10.3035	0.2877
5	1900	6, 8, 10, 12	10.0722	0.2323
6	2000	5, 7, 9, 11, 13	8.5300	0.2816
7	2100	6, 8, 10, 12, 14	7.5849	0.3691
8	2200	6, 8, 10, 12, 14	10.9346	0.3294
9	2300	7, 9, 11, 13, 15	14.0663	0.4382
10	2400	7, 9, 11, 13, 15	13.2381	0.5478
11	2500	8, 10, 12, 14, 16	13.5008	0.3456
Overall			12.2108	0.3440

Activation function for hidden neurons was kept as RELU (Rectified linear unit) [48]. Model with best validation performance was saved and was used for testing. Number of epoch = 100, batch size = 600. Adam optimizer [69] was used with hyper parameter set as default *i.e.* learning rate = 0.001, beta_1 = 0.9, Beta_2 = 0.999, epsilon = none, decay = 0.00 & amsgrad = False. RMSE was used to determine model performance. Figure 4.14 & Figure 4.15 shows prediction performance of RPM & flowrate by NN. Table 4.4 shows RMSE prediction value for different RPM conditions. RMSEcv for RPM= 11.0301 & for flowrate = 0.2796. Results show model performance is significantly better for flowrate than full PLS model however performance of RPM model is similar to PLS model.

It can be deduced from the results that NN was able to learn & identify unique patterns corresponding to different conditions along with amplitudes. Relationships between different overlapping vibration sources and flowrate can be extracted using a

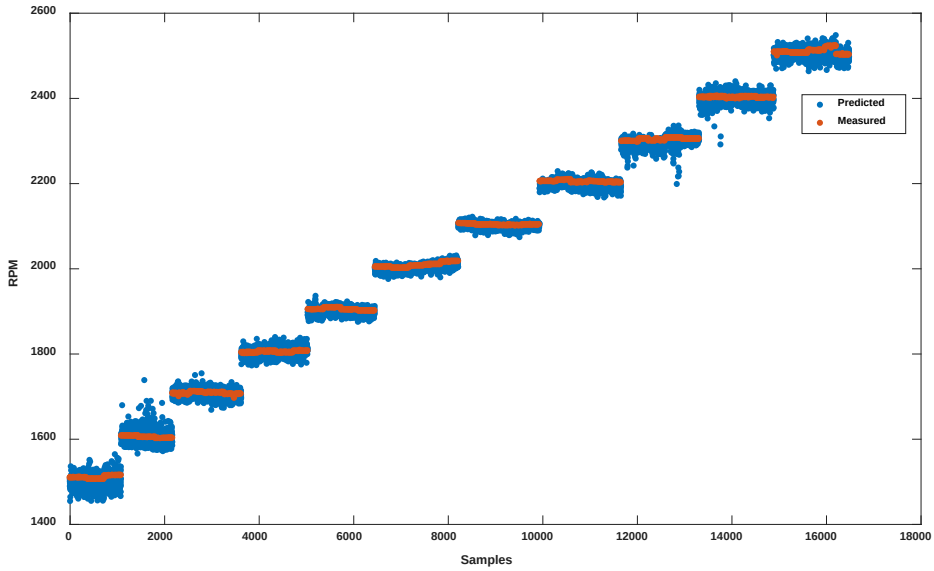


Figure 4.14 Deep NN model RPM Prediction performance

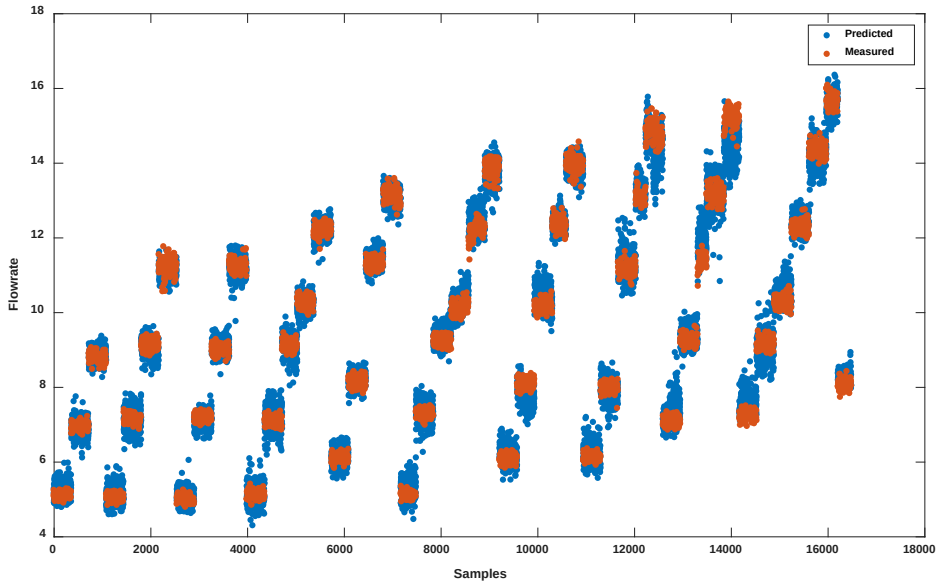


Figure 4.15 Deep NN model Flowrate Prediction performance

complex non-linear function, in other words NN learns pattern of vibration signals along

with its amplitude originating from different sources corresponding to each flowrate & RPM condition. However no major improvement is seen for RPM prediction this suggests that relationship between RPM & intensities of frequencies is linear, this can also be deduced from physical understanding of the pump RPM, *i.e*. as pump RPM increases vibration frequency shifts and thus position of the peak will remain fixed. Therefore deviation in RPM prediction can be attributed to inherently noisy data which results into noisy spectrum.

Even though NN performs fairly well there are several drawbacks for applying it at an industrial scale. NN has very large number of modelling parameters which makes model interpretability extremely difficult. Model interpretability is of utmost importance while working on industrial systems as possibility of unknown failure is high and well interpretable model will be extremely helpful for troubleshooting. As model is extremely complex, training such models is difficult as large amount of data is required in order to get fairly accurate NN model. This makes entire process extremely computationally intensive. For real-time applications it is important use models that can be updated easily, as manufacturing plants changes over time, and that are lighter. Moreover until this point data didn't explain anything about the underlying scientific fundamentals that causes the relationship with vibrations in the system. Thus in order to address these short comings, to identify scientific fundamental reason for the relationship and to develop overall lighter model, it was decided to carry out in-depth data analysis, identify underlying patterns and to finally develop a modelling approach which is more accurate, easily interpretable, includes only important modelling information (*i.e*. simpler) and is computationally efficient by incorporating system knowledge, overcoming data & noise

characteristics with machine learning modelling. In the next section detailed data visualization & its interpretation with scientific fundamentals is discussed. Next system information is combined with statistics & data is filtered a hierarchical modelling approach is presented for modelling for RPM & flowrate and finally noise characteristics are incorporated to further improve the overall model performance.

Chapter 5. Detailed data analysis, data filtering model development & signal reconstruction for monitoring framework.

5.1 Detailed data analysis:

5.1.1 Spectral analysis & data mining:

Figure 5.1 shows spectrum obtained when pump was under different conditions, for easier comparison only one spectrum for RPMs 1500, 1800, 2000 and 2400 are plotted. Spectrums from fixed RPM condition was plotted with a fixed color. It can be observed that there are clear separate peaks for condition of RPM. Thus RPM can be modelled with the data. Further, spectrum of each color are grouped with the same color and overlapping of different colored spectrum is not observed. It suggests that flow doesn't induce any frequency shift in the vibration signal. Moreover, multiple peaks at different frequencies are observed even for fixed RPM signal. Most of these higher magnitude peaks are at fairly lower frequencies and thus it can be concluded that these peaks corresponds to different rotating and static vibrating parts of the pump assembly and not to fluid vibrations in the pipe due to flow. Another component of spectrum is amplitude value at a given frequency, Figure 5.2 shows zoomed view of spectrum shown in Figure 5.1 from 40 Hz to 220 Hz. Here spectrums of same flowrate are assigned same color so that spectrum behavior at different pump RPM can be observed. It can be observed that amplitude values for fixed RPM have different amplitude value however no clear relationship can be observed. This requires for data mining so that it can be first establish that there is

some relevant pattern within the data which can be modelled and be related to desired property. After several consideration it was decided to use principal component analysis (PCA) [32], [68]for developing understanding about underlying multivariate pattern.

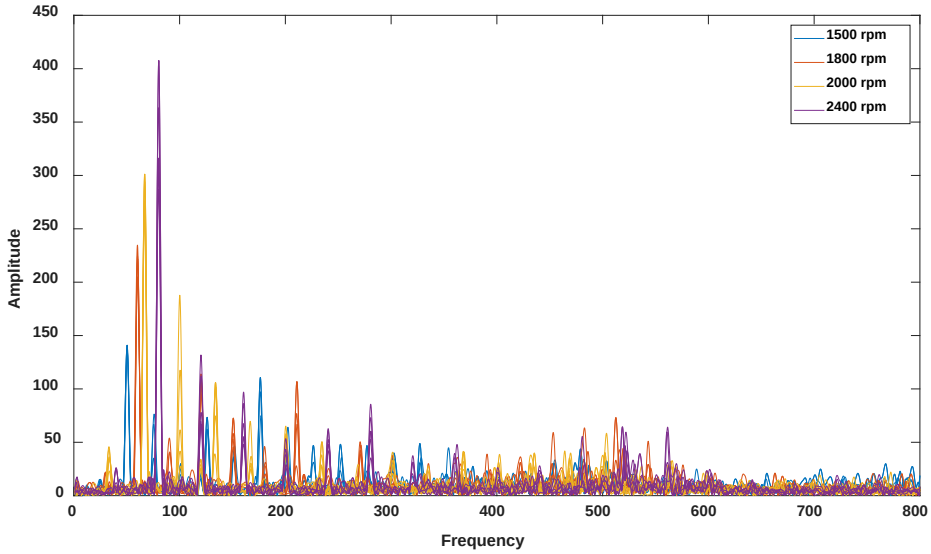


Figure 5.1 Spectrum of all flowrates at different RPM, same color spectrum corresponds to same RPM condition

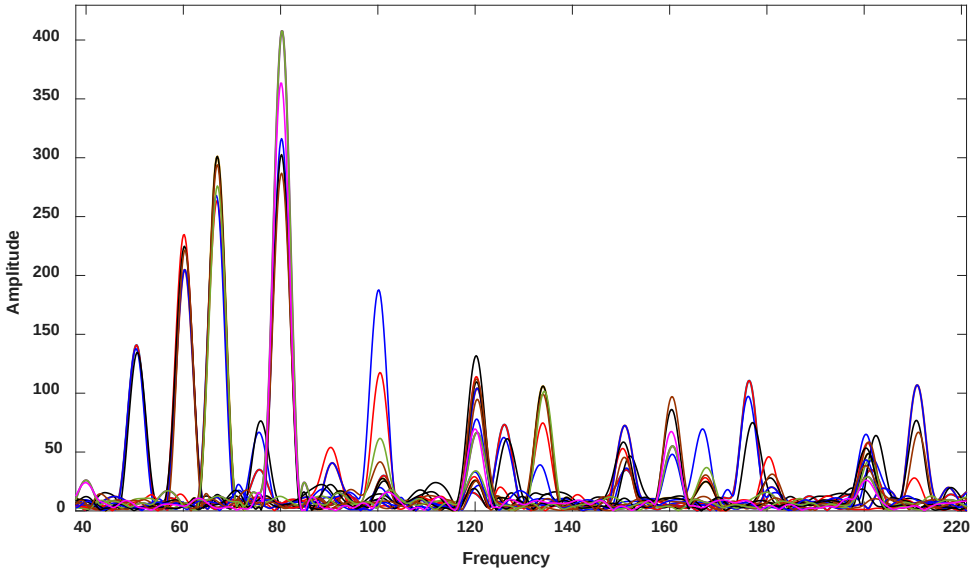


Figure 5.2 Zoomed Spectrum of all flowrates at different RPM, same color spectrum corresponds to same flowrate condition

5.1.2 Principal component analysis (PCA) for data mining:

PCA is a multi-variate un-supervised learning & variable reduction technique which decomposes the set original correlated variables into set of uncorrelated components. These components are derived such that they capture maximum variance of the original data in descending order *i.e*. First component will capture maximum variance from the original data, next second component will capture second highest amount of variance and so on. Another way to understand this is that PCA rotates axis of the original high-dimensional data such that maximum variance of original data is captured by new orthogonal lower dimensional variables. Maximum possible number of components that can be extracted is equal to number of variables in the original data matrix. However for most analysis only few of them are selected based on amount of variance explained by that particular component and problem at hand. This decomposition can be expressed as:

$$\boldsymbol{X = TP^t + E} \tag{5.1}$$

Where T = scores matrix & P = loading matrix. In this study NIPALS algorithm was used for decomposition. Analysis of these independent variables, scores, can help in identifying underlying pattern in the high dimensional original data. Further importance of each variable in each score can be observed from a loading matrix. For more details please refer [32], [68].

First, Columns of X contains values of amplitude at fixed wavelength was build. Rows of matrix X are spectrums samples obtained for different condition (combinations of RPM and flow). PCA was carried out on data collected at fixed RPM *i.e*. Separate analysis was carried out for fixed RPM condition. Multiple samples from each condition are included so that most important variations even under high noise value can be

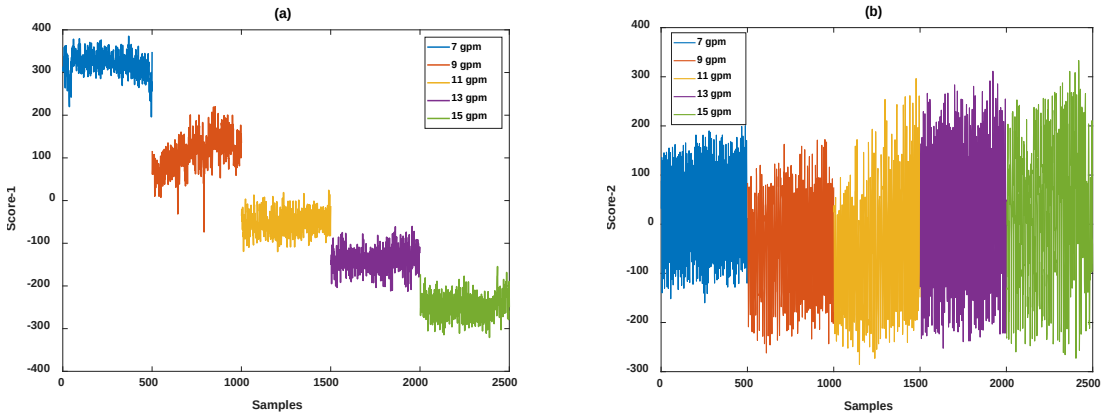


Figure 5.3 Condition 2400 RPM (a) Score-1, (b) Score-2 comparison

captured. Figure 5.3 shows score plot comparison for different flow conditions for 2400 RPM, first 500 points from each condition were chosen to construct a matrix X. PCA was carried out on mean centered matrix. It was not scaled as all the variables have same unit. It can be clearly seen that scores-1 values are very different for different flowrates, Figure 5.3 (b) also indicates presence of some difference between signals captured for different conditions. PCA score pattern can be interpreted using fundamental scientific idea of wave formation in mechanical system *i.e.* whenever there is change in load on a freely vibrating object with fixed amount of energy (*i.e.* fixed RPM) amplitude of vibration may increase of decrease depending upon the relation with load and natural vibrating frequency. In this particular case it can be explained as flowrate increases, load on pump assembly increases and so does the load on several pump components, therefore when pump was running at 2400 RPM with load increasing (*i.e.* flowrate increasing), overall vibration amplitude of several rotating and static vibrating parts of the pump changes. PCA captures this multivariate changes in frequencies and suggests that overall amplitude change is decreasing as flowrate increases. Higher principal components (PCs)

i.e. pc 2, 3 also extracts similar information and shows finer changes or differences across different flowrate conditions. Figure 5.3 (b) shows availability of such differentiating information observed in PC 2 and next few PCs also shows some components of differentiating information which can be used for modelling as in order to accurately predict flowrate, information from other principal components needs to be extracted and modelled. This results indicated that there is enough variability in the data with respect to flowrate which can now be modelled to build predictive model for constant flowrate monitoring.

5.2 System Engineering Enhanced Hierarchical Modelling Approach:

With the help of detailed data analysis and multivariate visualization techniques, underlying scientific fundamentals relationship between RPM & flowrate with vibration signals was identified. In order to build model which is simpler & more accurate it was decided to incorporate additional system knowledge by understanding application aspects of the testbed. Generally variable drive pumps run in stages of different RPM, not continuous RPMs, called operation stage based on the output flowrate requirement. Therefore for such applications an ensemble of different models corresponding to separate stages of operations can be used. Based on this idea an approach of hierarchical modelling for RPM & flowrate prediction is proposed. Under this setup a spectrum is first passed through an RPM predicting model. Once the RPM is identified then the spectrum is passed through the flowrate model corresponding to that RPM or that stage of operation.

In the previous results it was observed that RPM prediction was noisy and was quite inaccurate. Taking help of system knowledge, physics of pump operations & data

modelling techniques a binary matrix approach is proposed in this work which has ability to accurately identify pump RPM value.

5.2.1 Binary matrix approach for RPM prediction:

In spectral analysis section Figure 5.1 shows that there is a very clear relation between RPM and frequency value, not amplitude, at which largest peak appears. Amplitude of peak didn't matter and depended on load of the pump. Thus a model can be designed which identifies max frequency in the spectrum and predict corresponding RPM value. However due to extremely noisy and partially non-stationary signal maximum peak value corresponding to RPM might move around an actual frequency this problem was addressed by reinforcing the identified frequency with other frequency values around it. In order to incorporate peak frequency without adding any bias with its number value a binary matrix approach is proposed in this work. Steps are described below:

Vector of zeros of length 962 points was generated, each point corresponds to individual frequency. As it was observed that shaft coupling frequency for maximum RPM was under 100 hz therefore even if resolution between two integer frequencies is 0.2, total number of variables will be 501. Moreover after careful examination it was observed that spectrum from x & z have clear information about RPM. As a part of making model more robust to noise it was decided to augment binary matrix with information from x & z direction both & therefore 962 binary variables were selected where first 500 variables corresponds to frequency of x directions & remaining to z directions.

Highest frequency value corresponding to largest amplitude was identified, for spectrums from x & z directions.

Variable corresponding to the frequency is replaced with 1. 10 variables, 5 on either side, adjacent to the identified was replaced with 1 as well. This is done to reduce effect of noise and can be seen as an optimization parameter. Above points are carried out for all the samples to obtain final binary matrix of 0s & 1s.

PLS model is calibrated using this binary matrix with calibration set =48073 X 962. Number of PCs were optimized with validation set= 20604 X962 & test set = 16488 X 962.

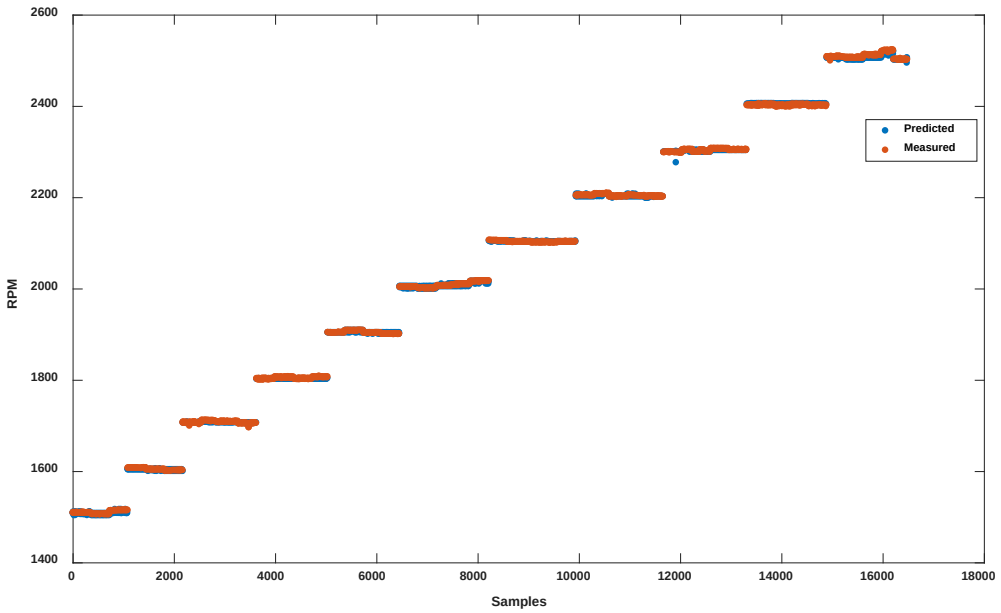


Figure 5.4 Binary matrix model RPM Prediction Performance

Figure 5.4 shows prediction results by binary matrix approach for all the cases considered for modelling. Number of PC chosen =5. It can be clearly seen that model RPM prediction using this approach is extremely accurate and will be able to identify RPM accurately for next stage of flowrate prediction.

5.2.2 Relevant frequency identification & data cleaning:

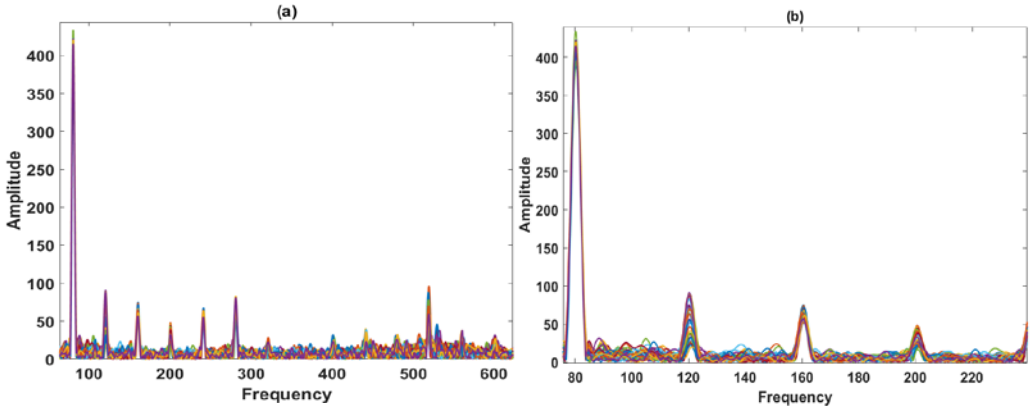


Figure 5.5 Spectrums at fixed condition 2400 RPM & 7 GPM

Before modelling for flowrate system & scientific fundamental knowledge was used to filter and simplify data by removing non-contributing information. For hierarchical modelling each flow rate model corresponds to corresponding RPM (or small range of RPMs) thus to identify uninformative variables conditions corresponding to fixed RPM needs to be observed. Figure 5.5 (a) shows spectrum of different samples for fixed condition *i.e.* 2400 RPM and 7 gpm. Figure 5.5 (b) shows zoomed view of frequencies between 70 and 210 Hz. From these figures it can be observed that for fixed condition *i.e.* fixed RPM and flowrate spectrum extracted have few relevant peaks and lot of irrelevant noisy frequencies. These frequencies do not add any information about flow but rather induces noise to the model and interferes in identifying optimal hyperplane. In ideal case noisy frequencies should have amplitude value zero but in practical cases noisy frequencies always have some amplitude. Moreover there might be presence of frequencies which has non persistent peak closest example of the such frequency can be seen in Figure 5.5 (b) even though condition is fixed *i.e.* fixed flowrate and fixed RPM peak value at around 120 Hz changes from higher 90 to low 14.34 it can be deduced that

this frequency may not be useful in extracting flow information. In order to identify peaks with sufficient information not drowned by noise & other uncertainties and after careful consideration of noise behavior in the data, presented in previous sections, author proposed a statistic, inverse of coefficient of variation. Hypothesis it tests is if the amplitude sample at a given frequency is corresponds to the noise distribution $N \in (0, \sigma^2)$ or not.

Use of coefficient of variation (COV) has been happening since quite long time and it is also known as relative standard deviation. COV will give standardize measure of dispersion of a frequency distribution and is defined as ratio of standard deviation and absolute value of mean. However use of Inverse of COV (ICV) have never been proposed or been used for such applications to the best of this author's knowledge. ICV given relative power a peak will have with respect to the amount of uncertainty, due to noise & non-stationarity, it carries for modelling. In this study ICV will be used in order to identify frequencies whose information is not muzzled by noise. Higher the value of ICV more the information with respect to uncertainty at given frequency. As ICV is a statistic & by central limit theorem it follows Gaussian distribution.

To filter unwanted frequencies, all spectrum for fixed RPM & fixed flowrate conditions are stacked together and ICV is calculated at each frequency under consideration. For example all amplitude values at fixed frequency of say 200Hz are used to calculate mean & standard deviations of the amplitudes, these are further used to calculate ICV at that frequency if the ICV value is greater than threshold then it will be considered as relevant frequency otherwise it will be discarded. This exercise is carried out at all frequencies & also on all combinations of flowrate & RPM conditions. As noise

distribution $N \in (0, \sigma^2)$, threshold value for making decision was fixed at 2.4. Value greater than 2.4 indicates that user is ~98% confident that the ICV value doesn't corresponds to the noise distribution. In other words the amplitude value observed at a given frequency has relevant information and ICV corresponds to the distribution other than zero mean Gaussian noise distribution. Different threshold value can be chosen depending on data under consideration and model requirements. Figure 5.6 shows zoomed view of ICV vs frequencies plot calculated for a fixed condition 2400 RPM and 7 gpm for all three x, y and z direction signals. Significant number of variable will be eliminated giving room for adding more relevant information for better prediction with cheap computation.

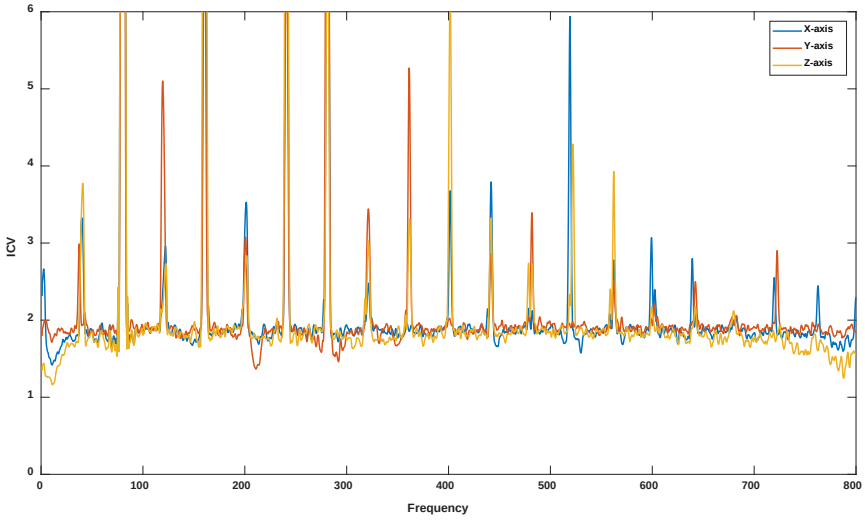


Figure 5.6 ICV values for x, y, & z direction data for 2400 RPM & 7 GPM

5.2.3 Flowrate model prediction:

After developing approach for removing uninformative frequencies, second stage of hierarchical modelling was designed. Under second stage of hierarchical modelling flowrate prediction for several stages of pump operations is carried out. In order to build

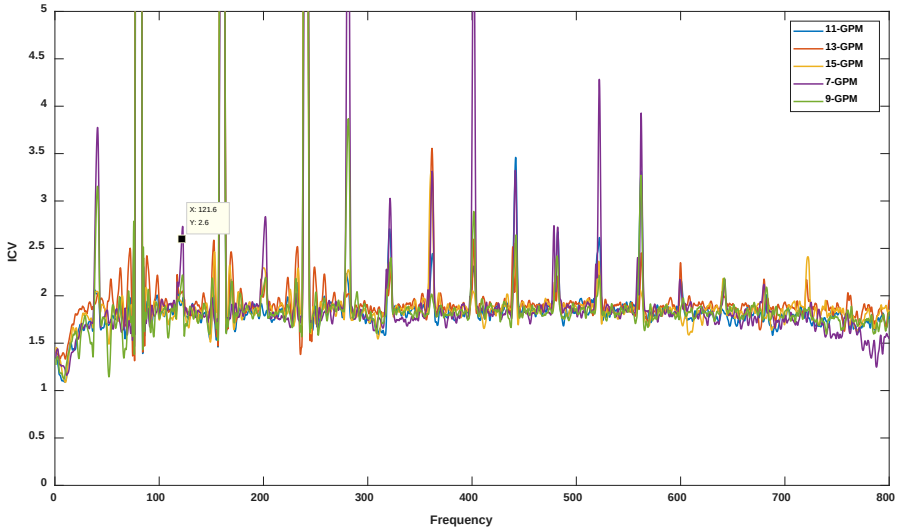


Figure 5.7 ICV values of z direction data from 2400 RPM & all flowrate

these flowrate models, first of all relevant frequencies for fixed RPM condition but different flowrate conditions are identified and combined. A matrix containing all the sample spectrums of amplitude at combined relevant frequencies corresponding to fixed RPM & different flowrate conditions is built. Logic behind combing relevant frequencies is, as change in flowrate corresponds to change in pump load itself it is possible that some frequencies that are not expressed for one condition will get expressed for other condition and this may result into different values of ICV. Figure 5.7 shows ICV plot for z direction for different flow condition. In this study amplitude values corresponding to relevant frequencies from all three direction stacked column wise & thus spectrum variables are available in all three, x, y, & z directions. Different flowrate models corresponding to different stage of pump were built. Number of samples & number of variables included in calibration set, validation set & test set for different models is given in Appendix A.2. Details about RMSE prediction, RMSEcv & PC selected is given in Table 5.1. Figure 5.8 show prediction performance of all the flowrate models under consideration in one plot, individual

performance plot are shown in appendix A.3. Removal of noisy information not only improves performance but also makes model simpler, significantly improves computational time & have better model stability. As number of samples for each model reduced cross-validation used for this modeling was Monte Carlo Cross validation (MCCV). Calibration & validation division ratio = 50% number of Monte Carlo runs= Total number of training samples/12.

5.2.4 Improved modelling by incorporating system noise behavior:

Results obtained by hierarchical modelling for flow & RPM prediction are quite good and useable for many applications, however on closer observation it can be seen that prediction are still noisy. In the data characteristics sections it was shown that measurement, both RPM & flowrate, used for model building are noisy and thus prediction performance can only be as good as measurements, moreover on observing vibration sensor data characteristics it was seen that a layer of Gaussian noise gets added to the signal measurements. Thus in this approach, noise behavior studied in earlier section for both RPM & flowrate measurements & vibration signals was used to further treat the data and mitigate its effect. Noise behaviors for different measurements were classified as zero mean Gaussian noise. As both RPM & flowrate measurement & vibration signals have Gaussian noise it was decided to use moving average of 6 samples for RPM, flowrate & spectrum for fixed condition samples. Fixed RPM condition samples corresponding to all flowrates are combined for model building. Use of moving average mitigates noise effect in both spectrum & RPM & flowrate measurements. Number of samples in calibration, validation & test set is given in Appendix A.2. Details about performance of the model, RMSE prediction, RMSEcv & PC selected, built after incorporating system & noise knowledge is given in Table 5.2.

Figure 5.9 show prediction performance of all the flowrate models under consideration in one plot, individual performance plot are shown in appendix A.3.

Table 5.1 Validation & Prediction performance of Hierarchical model

RPM model	RMSEcv	PC selected	RMSEp
1500	0.2037	8	0.2145
1600	0.2094	8	0.2453
1700	0.2351	8	0.2792
1800	0.2938	11	0.3224
1900	0.2379	8	0.2361
2000	0.2938	10	0.3527
2100	0.2694	7	0.3019
2200	0.3072	9	0.3476
2300	0.3275	10	0.3692
2400	0.3463	8	0.3482
2500	0.2855	8	0.2839
Overall			0.3099

Table 5.2 Validation & Prediction performance of Improved Hierarchical model

RPM model	RMSEcv	PC selected	RMSEp
1500	0.1126	10	0.14570
1600	0.1008	12	0.16026
1700	0.1230	12	0.16351
1800	0.1365	12	0.18399
1900	0.1190	13	0.14455
2000	0.1510	15	0.26346
2100	0.1553	11	0.23118
2200	0.1654	16	0.27044
2300	0.1748	14	0.24761
2400	0.2124	12	0.22254
2500	0.1687	10	0.19869
Overall			0.21470

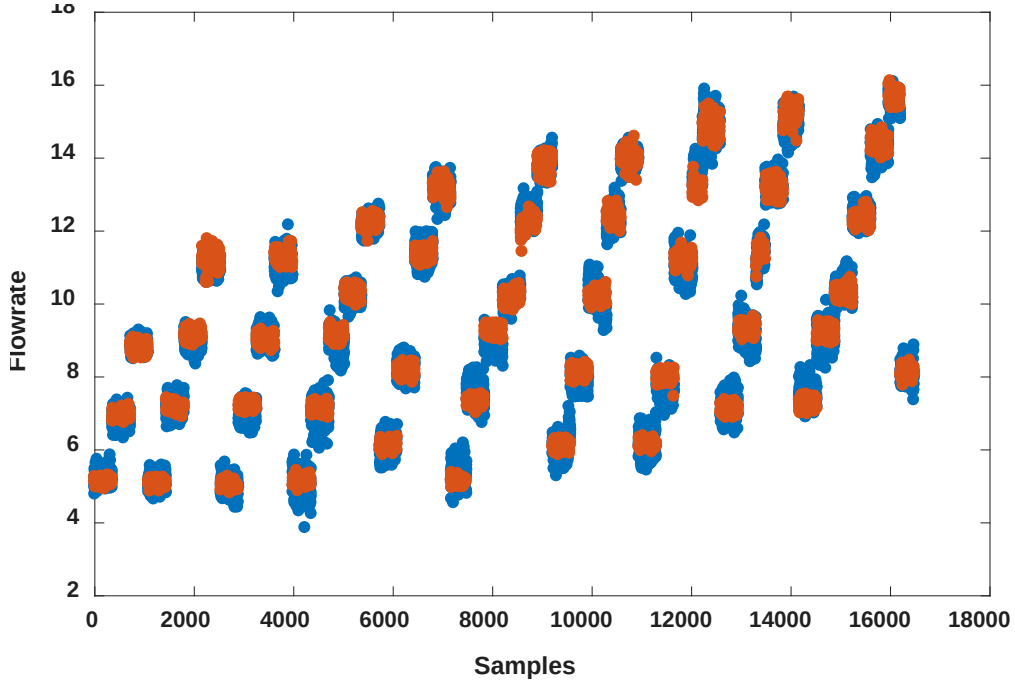


Figure 5.8 Hierarchical Model's Prediction performance for all the test samples.

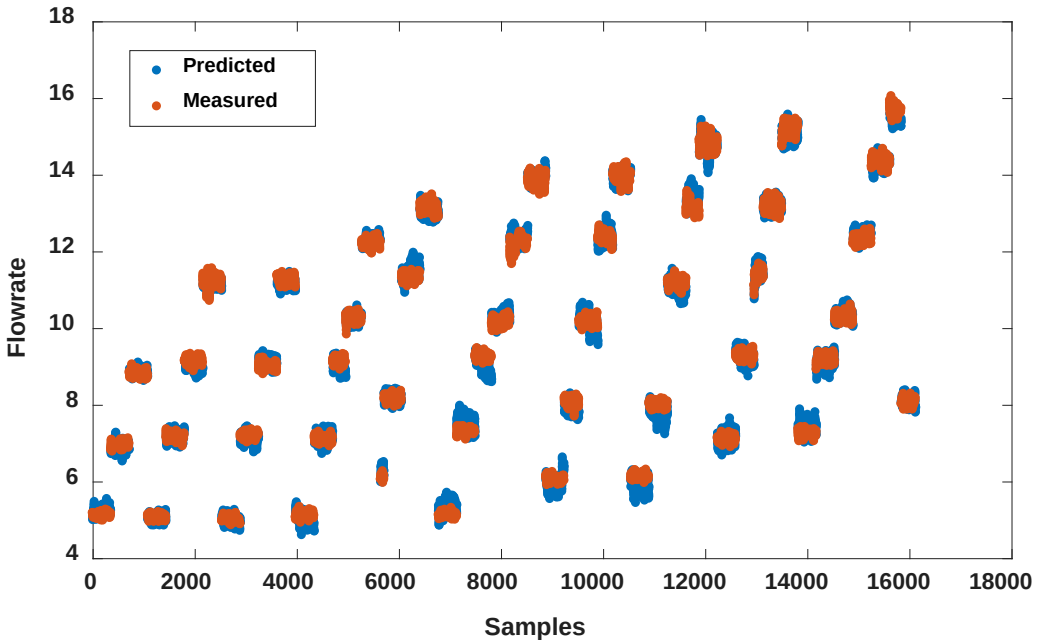


Figure 5.9 Improved Hierarchical Model's Prediction performance for all the test samples.

5.3 Signal reconstruction for fault detection:

It has been seen as part of IIoT sensor characteristics that there is very high possibility of getting missing signals, length for which signal goes missing are random & of varying length, in worst case scenario sensor stops responding. Identification of this missing spectrum will be an asset for identifying if the fault has occurred in a particular sensor. if the duration during which signal is not reported is longer then prediction of flow can't be reported during that time which can be a problem if its value is further used for any control systems or even for decision making by operator. Generally problem of missing signal can be solved by just putting one more sensor in the same location but this would further add capital and computation cost. Also space available for sensor placement might be limited and it is might be impossible to use two sensors. Thus to deal with such situations, in this section a signal reconstruction process is proposed which utilizes information of other sensors and reconstructs missing signal. As data from sensor-4 has been used extensively for flowrate prediction. For the purpose of demonstration signal for sensor-4 will be reconstructed & its reconstruction results will be presented.

Idea of this reconstruction approach stems from the idea of vibration propagation. Vibration signal from one location tends to propagate to different parts it is connected with, vibration wave may tend to get distorted and but still have information about waves from other locations. In pump assembly under consideration, vibration data was collected from 5 different location. Based on relation between vibration amplitude and flowrate, signal from sensor-4 was enough to give good prediction results. Although as vibration signal propagates to different locations, signals from other sensors can have information about sensor-4 vibration signals. Assuming, that signals from other sensor do have

information about signal from sensor-4 then a model can be built that can identify this relation and reconstruct signal from sensor-4. Which will be highly desirable.

This reconstruction approach combines properties of PCA & PLS to obtain missing signals. Firstly, spectrum samples for fixed RPM condition & all flowrate conditions from all 5 sensors were stacked row wise to build an augmented matrix, variables of this matrix were identified using approach explained in data cleaning section. For model building care was taken that samples for all sensors were collected at same time corresponding to flow rate, this ensures that sensor 1-5 also includes samples which have information about each other's vibrations. Once augmented matrix is obtained steps for building reconstruction are given below:

- Augmented matrix is mean centered & PCA is carried out. Number of PCs were decided based in amount of variance explained. Scores & loading matrix are saved for further use.
- Once scores of PCA model are obtained, scores of sensor 1, 2, 3 & 5 are stacked column wise *i.e.* these scores are combined to form independent variables for PLS model *i.e.* X-matrix of PLS model.
- Scores of sensor 4, from PCA, forms Y-matrix or dependent variables for PLS model building. Scores of sensor 4 were predicted as signal for sensor 4 needs to be reconstructed.
- Once X & Y matrix are ready for PLS model building. Different PLS models for each y vector or PCA score of sensor-4 is calibrated using same score matrix of sensors 1, 2, 3 & 5 built in previous steps. Number of PCs to be included is identified based on % variance explained or using Cross-validation.

- Once PCA & PLS models corresponding to all scores of sensor-4 is built than these models in combination will be used for spectrum reconstruction.

Steps for sample reconstruction:

- It is assumed that sample for sensor-4 isn't available for a given time interval. For the same time interval using samples from remain sensors & stored loading values of PCA model, corresponding scores are obtained.
- Obtained scores are arranged in variable form compatible with save PLS models. Using this scores values of scores corresponding to sensor 4 is predicted.
- These predicted scores are arranged multiplied with stored PCA loading matrix which will result in to a re-constructed spectrum.

Schematic of this entire process is shown in Figure 5.10.

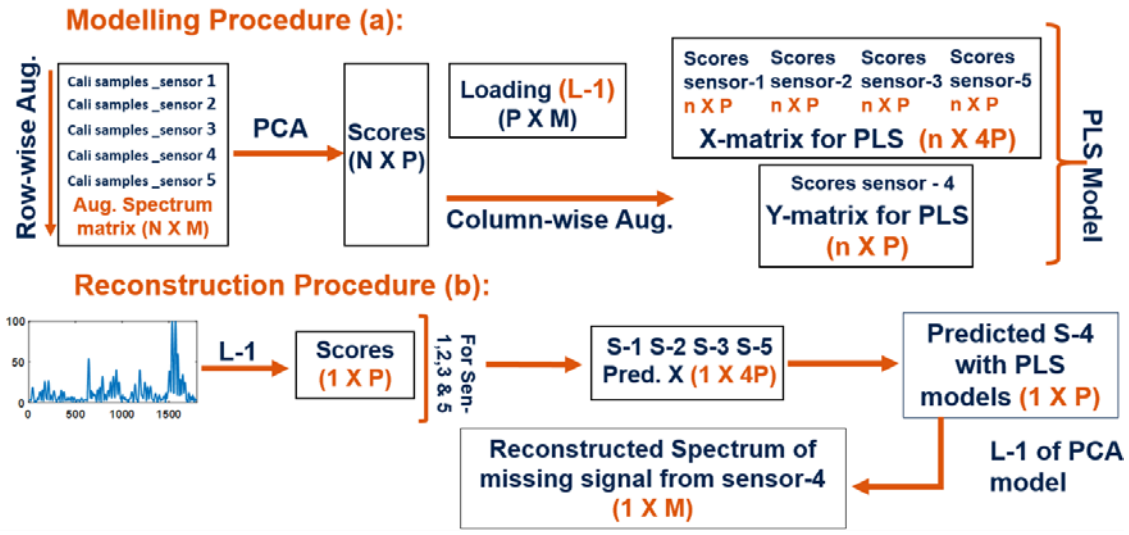


Figure 5.10 Schematics of (a) modelling & (b) reconstruction procedure for PCA-PLS based re-construction approach

Above approach was applied on different RPM models and tested on using test samples. Figure 5.11 (a) & (b) shows reconstruction results for model corresponding to

1700 RPM & 2500 RPM, respectively. RMSEP between original & reconstructed signals are 4.5 & 4.6 for 1700 RPM & 2500 RPM cases, respectively. This indicates that proposed approach is applicable on entire range of testbed operations. Use of PCA scores for extracting relation between signal of sensor-4 & other sensor significantly reduces number of variables involved & makes entire approach computationally feasible & efficient. If PLS model was built just by using spectrum of sensors then number of variables will be very large and it might get difficult to identify a correct information under such noisy signals. The reconstructed signals can be further used to predict flowrate during time of missing samples. Figure 5.12 (a) & (b) shows flowrate predictions for 1700 RPM model & 2500 RPM model using reconstructed signals, respectively. It can be seen that for both cases prediction performance is in comparison with prediction by original spectrum. For higher RMPs where relationships gets non-linear it can be models for reconstruction can be replaced with some non-linear models like ANNs *etc*. to get more accurate predictions.

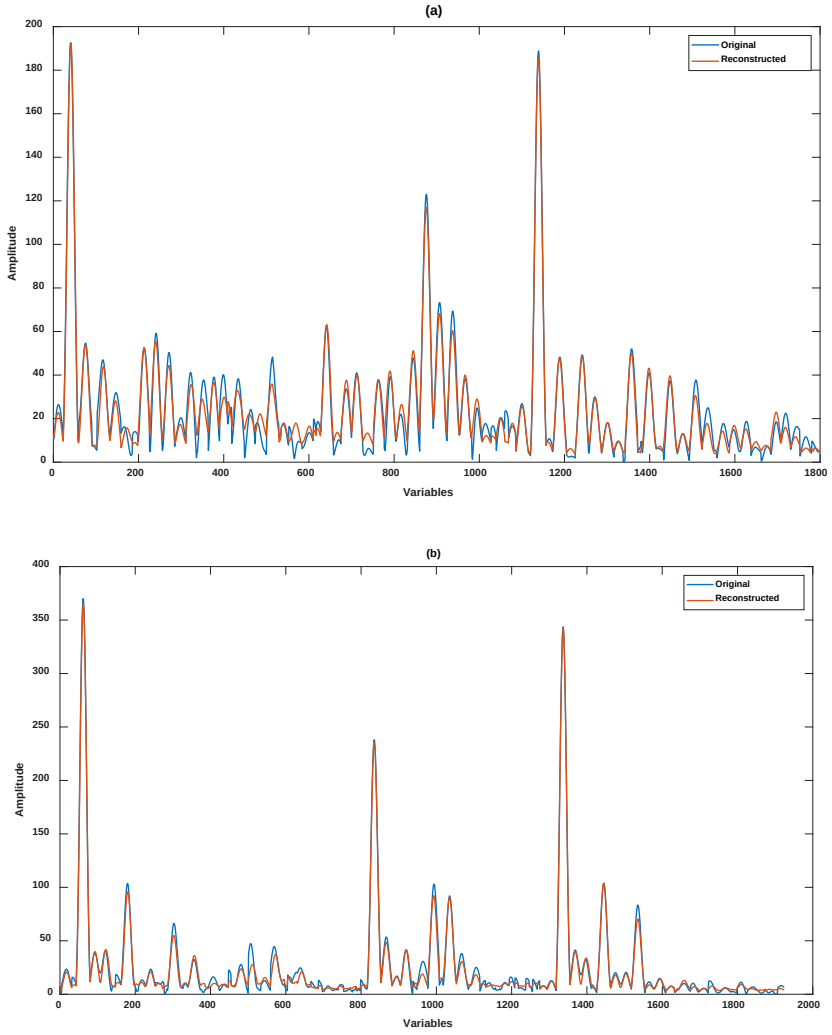


Figure 5.11 Reconstructed spectrum for (a) 1700 RPM & (b) 2500 RPM Conditions

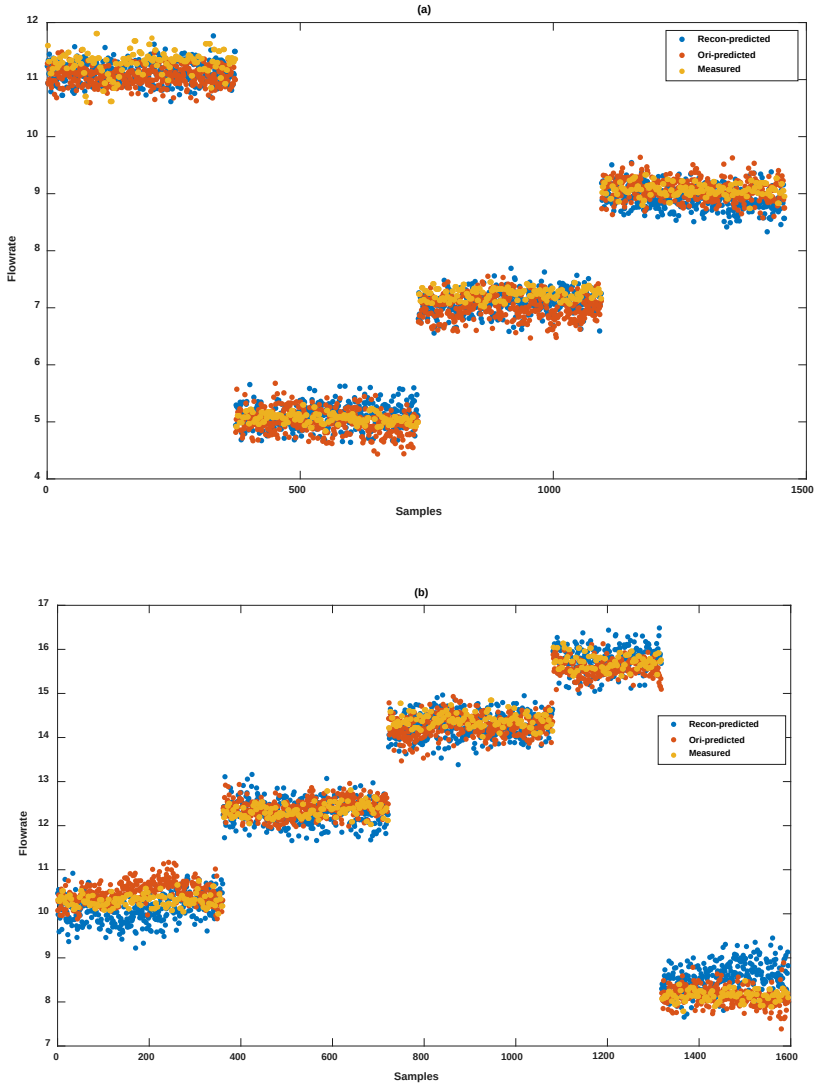


Figure 5.12 Prediction using reconstructed spectrum: Recon-predicted=Prediction using reconstructed spectrums; Ori-predicted= prediction using original spectrum

Chapter 6. A feature-based soft sensor for spectroscopic data analysis

One of aspects about data modelling that hasn't been focused much in the past and will be extremely crucial for new generations of big data analytics is the situation where direct relationships between easily measured data or dependent variable & variable of interest or predicted variable isn't clear and modelling with dependent variables directly resulting in poor prediction performance. Another major challenge for better model performance for future big data analytics is distribution of available data may not follow Gaussian distribution, which is a basic assumption for many modelling techniques, and that it is extremely noisy. A feature based approach proposed in this section for spectroscopic data, author believes such data can be extremely useful for future big data modelling for different applications as there are several types of spectrums possible and almost everything physical has energy, frequency & vibrations.

Spectroscopic techniques such as near-infrared (NIR) and UV/Vis spectroscopies have gained wide applications in the last few decades due to their advantages over other analytical techniques, such as non-invasiveness and low pre-treatment requirement. Beyond their traditional applications in analytical chemistry, spectroscopic techniques have been applied in many different fields to determine properties such as octane number [70], moisture content [71], active chemicals in a samples [72], and microorganism

concentration [73]. In order to correlate the spectroscopic readings of a sample to its properties of interest, multivariate regression models, also known as soft sensors, are often developed, which usually utilize multivariate statistical methods such as multiple linear regression (MLR), principal component regression (PCR), partial least squares (PLS) [42], [74] and canonical variate analysis (CVA) [75]. Interestingly, although CVA identifies directions of maximum correlation between response and regressor variables while PLS may theoretically include directions that are irrelevant to response variable(s) [75], [76], PLS is the most commonly used soft sensor platform and there seems no research showing the advantage of CVA over PLS for soft sensor modeling. In addition, when variable selection is implemented prior to soft sensor modeling, it is expected that the variable selection process would exclude regressor variables/features that are irrelevant to the response variable(s). Therefore, for this study it was decided to use PLS as the modeling backbone for the proposed approach, although it is straightforward to extend the method to CVA based soft sensor. Meanwhile, nonlinear soft sensors that utilize artificial neural network (ANN) or kernel-based methods such as support vector regression (SVR), kernel-PLS (KPLS), *etc*. have also been proposed in the literature [7], [77]. As most spectroscopic datasets have relatively small sample size (e.g., three out of four datasets used in this work have only 21-36 training samples and 16-28 validation samples), they are not sufficient to train a good NN model. Therefore, in this work, we examine KPLS based nonlinear soft sensor. It has been shown that KPLS is a very effective soft sensor approach competitive with other kernel-based approaches such as SVR [78]–[80]. Other advantages of KPLS include its robustness and straightforward generalization, and ease of tuning of the parameters [81].

For absorption spectroscopic measurements, absorbance values at different wavelengths correspond to light absorbed by different components of a sample as illustrated in Figure 6.1 where the NIR spectra of a pharmaceutical tablet dataset are shown. It can be seen from Figure 6.1 that there are many clear absorption bands of the active pharmaceutical ingredient (API) from 600 to 1800 nm. Since the number of variables, which equals to the number of wavelengths where absorbance are measured, are usually large, substantial number of samples are required for building robust models. However, in

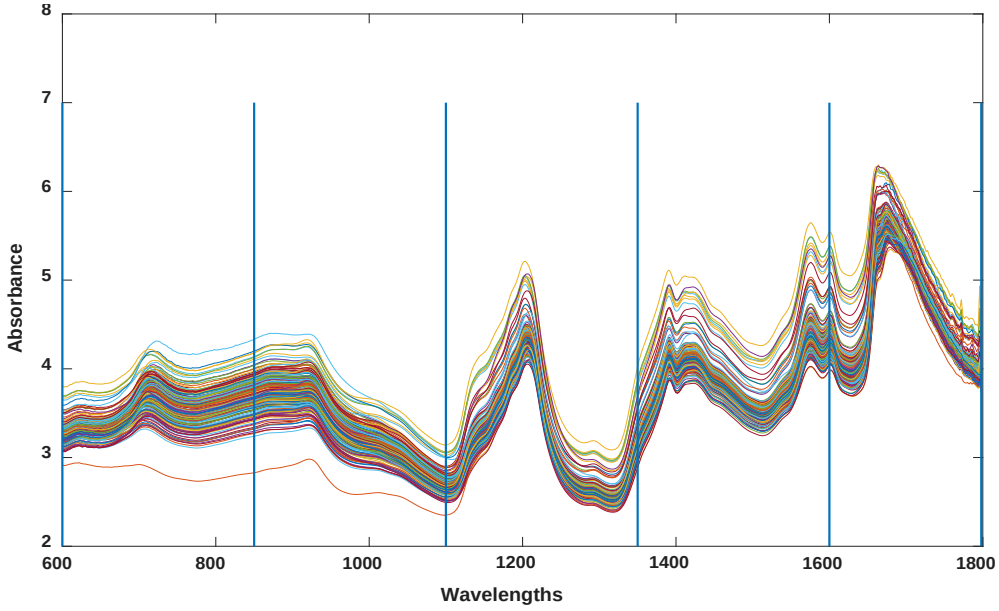


Figure 6.1 NIR Spectra of Pharmaceutical Tablets

some applications, the number of samples are limited. In those cases, the so-called "curse of dimensionality" would affect the predictive power of the model, where insufficient number of samples (compared to the number of variables) are used to build the model. On the other hand, it is well known that absorbance values at different wavelengths are not equally important in building such models. In addition, as shown in Figure 6.1, the absorbance of adjacent wavelengths offer similar information – because the general

features of molecular spectra are of continuous bands. In other words, spectroscopic data contain large number of redundant or highly correlated spectral variables. Although multivariate regression methods based on dimension reduction approaches such as PCR and PLS have inherent capability of handling large number of correlated variables, it has been shown that variable selection, when combined with multivariate regression, can significantly improve the soft sensor's prediction performance, reduce the model complexity, as well as provide better insight into the nature of the process/system of interest [7], [11], [82], [83]. The goal of variable selection for spectroscopic data is to identify the subset of wavelengths that are closely related to the interested properties of a sample such that the model built using the subset of the wavelengths can better estimate the properties for new samples. Another potential benefit of variable selection is to eliminate measurements at wavelengths containing significant noises for better accuracy and performance of the soft sensor models [84].

Due to the benefits mentioned above, variable selection is viewed as a critical step in spectroscopic chemometrics model development and has drawn significant interest in the last few years in different areas of applications [85]–[90]. Even though variable selection, when done properly, often improves the spectrum model prediction performance, it does carry some limitations. As shown later in this study, variable selection can produce soft sensor models that are sensitive to the choice of training and validation data, *i.e.*, data used for model calibration. Due to the noises and unknown disturbances contained in the training data, the wavelengths selected to optimize the calibration performance based on the training and validation data may be "tilted" to overfit or capture the noise or unknown disturbances contained in the calibration data. As a result, the model prediction

performance may deteriorate significantly when model is extrapolated or applied to new samples. In fact, this limitation is not unique to spectroscopic chemometrics models; instead, it is true to all data-driven soft sensor models, which is in essence a balance between model accuracy and robustness. To help address this limitation, a new feature-based soft sensor approach is proposed by adapting the statistics pattern analysis (SPA) framework developed for process monitoring [91], [92]. In the SPA enabled feature-based soft sensor modelling approach, the whole sample spectrum is divided into segments, and different features of each spectrum segment, instead of the spectrum readings themselves, are engineered so that property of interest can be explained in a better way resulting in a better performing, more robust soft sensor model. As the variables are engineered from raw data these variables have more relevant information with respect to property to be predicted. Moreover when the set of such engineered variables are assembled then information relevant for property prediction contained in the whole spectrum can be extracted with significantly less number of variables. The performance of the proposed method is extensively tested in this work and is compared with a full PLS model utilizing all raw variables, a shrinkage method least absolute shrinkage and selection operator (Lasso) [93], an interval based variable selection method synergy interval PLS (SiPLS) [94], [95]. In addition, nonlinear KPLS based soft sensor applied to the original full spectra, as well as to the SPA features has been explored. Their performances is examined and their pros and cons are discussed. Four datasets from different fields, including agriculture, petroleum, pharmaceutical and biochemical, are chosen to show the versatile applicability of the proposed feature-based approach. For consistent and fair comparison, Monte Carlo validation and testing (MCVT) procedure is

proposed and three MCVT-based performance indices for evaluating the performance of different soft sensor methods across different datasets. It is worth noting that the MCVT procedure and the MCVT-based performance indices are generally applicable for model comparison in other applications.

6.1 Brief review of PLS, LASSO, SiPLS & Kernel-PLS for spectroscopic modelling:

6.1.1 Partial least squares using full spectrum (Full PLS model):

In this case, PLS algorithm described in previous chapters was used. X matrix is built using raw NIR/UV/Vis spectrum data without any data filtering or variable selection. X matrix has dimension of $n \times p$, where n and p are the numbers of samples and wavelengths, respectively. Each row of $\boldsymbol{X}$ corresponds to a single spectrum of absorbance measured at different wavelengths of a sample. The physical or chemical properties of the samples are used to construct the dependent variable matrix Y with the dimension of $n \times$ m where the number of properties is m. In this study the variables in X & Y are centered by subtracting their means & scaled by its standard deviation. For all the cases on one property was predicted however the proposed can be applied for other multiple property case as well.

6.1.2 Least absolute shrinkage and selection operator (Lasso):

In Lasso minimization objective function is added with L^1 regularization on the regression coefficient *i.e.* it adds a penalty for larger value of coefficients forcing them to as small as possible. Because of the nature of regularization in the case if lasso it forces many coefficients to zero during optimization process making it an approach with inherent relevant variable selection capability. Objective function minimized in Lasso is given below [93]:

$$J(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta}}{\mathbf{argmin}}\left(\|\boldsymbol{y} - \boldsymbol{X\beta}\|_2^2 + \boldsymbol{\lambda}\|\boldsymbol{\beta}\|_1\right) \quad \textbf{(6.1)}$$

Where λ is a nonnegative regularization parameter to be optimized during model calibration or cross-validation. More detailed discuss on Lasso algorithm can be found in [93], and comparison of Lasso to other variable selection methods for PLS-based soft sensors in [96]. In this work, the lasso function from Matlab Statistics and Machine Learning Toolbox is used.

6.1.3 Synergy interval-PLS variable selection approaches (SiPLS):

One of the early work under this category is called interval partial least squares (iPLS) [94]. In this method, a whole spectrum is divided into *N* non-overlapping segments of the same size (except the last segment) as shown in Figure 6.1. For each segment, a PLS model is developed. In order to find the segment that provide the best performance, the procedure is repeated with different value of *N*'s, and a standard cross-validation approach such as RMSECV (root mean squared error of cross-validation) can be used for performance evaluation. The segment with the best performance on the validation data, together with the optimal parameters such as the number of principal components (PCs), are used to develop a final PLS model for prediction on the test data. Case studies have shown that the performance of iPLS is comparable, rather than outperforming, other variable selection based methods such as principal variable (PV), forward stepwise selection (FSS) and recursively weighted regression (RWR). Since then, several variations of iPLS have been proposed, including backward interval PLS (biPLS), Moving window PLS (mwPLS), and Synergy interval PLS (SiPLS) [94], [97]. In biPLS, the spectrum is divided into *N* segments and a PLS model is developed by

leaving out one interval at a time. Intervals are removed until last, best performing interval is identified. In mwPLS, PLS models are calculated based on moving window approach, size of the window is fixed and window with best performing model is considered for future predictions.

SiPLS is one of the specialized case of interval PLS or interval based variable selection approaches. In this study SiPLS was chosen as it is best performing approach [94]. Compared to iPLS where only a single interval is used for model building, SiPLS identifies best combination of 2, 3 or 4 different segments for the final model based on which combination gave best validation results. The tuning parameters for SiPLS include the number of segments the spectrum to be divided into (*i.e.*, *N*), the number of segments to be included in the model, and the number of principal components (PCs) to be retained in the model. In this study, only results obtained using SiPLS is compared as it was shown that SiPLS outperforms other interval based approaches [94]. The SiPLS Matlab code used in this work was downloaded from www.models.life.ku.dk/iToolbox. But was updated to incorporate a proposed Monte Carlo validation & testing approach (MCVT) presented later in this study.

6.1.4 Kernel-PLS:

Many variations of Nonlinear PLS have been proposed in literature [98]–[101]. Although they can be mainly classified in two categories [100], in category-1 observed independent variables are projected on to non-linear surface while keeping inner relation between scores linear. In category-2 linear relationship between the scores is replaced by modelling non-linear relationship between Independent variables and dependent variables. In this study it was decided to use method belonging to categaory-1 as these

methods can be easily implementable and thus easily scalable, generally computationally less demanding and capable of modelling complex non-linear relations [78], [100].

Kernel partial least squares is an approach under category-1. For kernelizing linear PLS algorithm each point or sample X is mapped onto a higher dimensional feature space using a non-linear function ($\Phi(X)$). A linear regression function ($f(\Phi)$) is constructed in this higher dimensional space corresponding to the non-linear mapping function ($\Phi(X)$). This mapped data appears as a dot products of the mapping functions in dual space and thus by applying "kernel trick" dot product of mapping function can be replaced by kernel function ($K = \Phi\Phi^T$). This approach is similar to kernel-PCA and was used to derive first K-PLS algorithm [78], [79]. A completed derivation and discussion of K-PLS from PLS algorithm using mapping function and constructing linear function corresponding to mapping function is given in [79], [102], [103].

NIPLS algorithm used in this comparison [78], [79], [100]:

$$\boldsymbol{t} = \boldsymbol{K}\boldsymbol{u} \tag{6.2}$$

$$\big|\big|\mathbf{t}\big|\big| \rightarrow \mathbf{1} \tag{6.3}$$

$$\boldsymbol{c} = \boldsymbol{Y}^{\boldsymbol{T}} * \boldsymbol{t} \tag{6.4}$$

$$\boldsymbol{u} = \boldsymbol{Y}\boldsymbol{c} \tag{6.5}$$

$$\big|\big|\boldsymbol{u}\big|\big| \rightarrow \mathbf{1} \tag{6.6}$$

Where $K = \Phi\Phi^T$, K=gram matrix using kernel function

For the above algorithm gram matrix is centered so that bias value is zero. K is centered by $K = (I_n - \frac{1}{n}1_n 1_n^T)K(I_n - \frac{1}{n}1_n 1_n^T)$ operation. Y was scaled to 0 mean and

unit variance. In this study as statistics were already identified and showed better performance with significantly less number of variables it was decided to use identified statistics as independent variables X. X was transformed using Gaussian kernel function, standard deviation as tuning parameter. Gaussian kernel used had form:

$$\mathbf{K} = e^{-(\frac{x-x_i}{2*\sigma^2})} \tag{6.7}$$

Where x_i is are considered as landmarks or points which gives basis for similarity between sample x with x_i. σ is the tuning parameter of the Gaussian kernel. Other kernel types like, polynomial kernel was also tested but it was observed that results from Gaussian kernel were better than others. σ was tuned using cross validation. K-PLS code written in matlab was incorporated with a proposed Monte Carlo validation & testing approach (MCVT) presented later in this study.

6.2 The Proposed Statistics pattern analysis (SPA) enabled feature-based soft sensor:

A good variable selection methods result in reduced models that are simpler, more robust and provide better prediction performance. However, when the training samples are not properly selected or there are not sufficient samples to cover the entire range of the properties to be predicted, the variable selection may be biased towards the covered property region while the extrapolability of the model can be poor. This can be evaluated by comparing the performance of the model on the validation samples to that of the test samples. A significant deterioration of performance on the test samples compared to that of the validation samples would indicate such a deficiency in variable selection. To

address this potential issue while still significantly reducing the number of variables, a feature-based soft sensor using statistics pattern analysis (SPA) is proposed.

Statistics pattern analysis (SPA) is a process monitoring framework developed by [91], [92]. In which statistics of process variables, instead of the process variables themselves, were monitored to determine the process operation status. SPA offers many advantages such as effectively addressing process nonlinearity and non-Gaussianity, non-synchronized batch trajectories, *etc*. Its effectiveness and performance in process monitoring have been demonstrated in multiple case studies [91], [92].

In the original SPA based process monitoring approach, the statistics are calculated along the time dimension and PCA is performed on the statistics for fault detection and diagnosis. There is no response variable involved. Also, the statistics cannot be obtained on an individual sample. There must be a group/window of samples in order to estimate the statistics. In the proposed SPA feature-based soft sensor, the statistics are calculated along the variable (*i.e*., wavelength) dimension and the statistics are correlated to response variable(s) through PLS. In this case, statistics is estimated based on an individual sample and the properties are estimated individually for each test sample. Specifically, as shown in Figure 6.2, in the SPA-based soft sensor each spectrum is first divided into s non-overlapping segments, which is similar to SiPLS; then f different features are extracted from each spectrum segment for each sample, which are raw spectrum readings without any scaling. The extracted features, such as the mean, standard deviation *etc*. are used as the regressor (totally $s \times f$ features for each sample) to build the soft sensor model. With n samples, the dimension of X would be $n \times (s \times f)$ and the dimension of Y would be $n \times 1$ for a single property, or $n \times m$ for m properties. Both X

and *Y* are auto-scaled to zero mean and unit variance for PLS modeling. The spectrum segmentation intervals (or number of segments), statistics used for model building, and number of PC's for PLS are optimized based on cross validation. In this way, information from the whole spectrum will be utilized for model building, but with significantly reduced number of variables. The schematic diagram of the proposed SPA enabled feature-based soft sensor approach is shown in Figure 6.2.

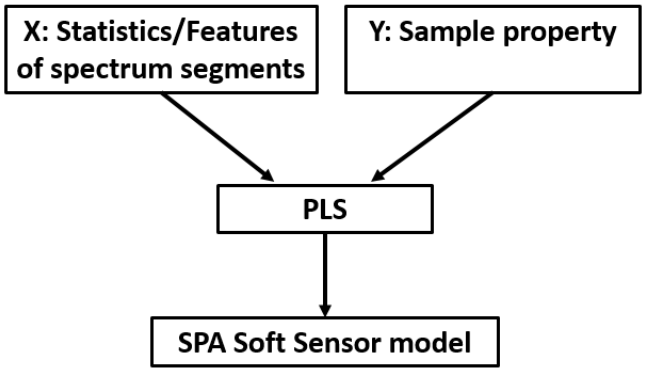


Figure 6.2 Schematic of SPA feature-based soft sensor

There are several benefits associated with the SPA feature-based soft sensor. First, it utilizes the information from the whole spectrum to build the soft sensor model, which provides better model robustness; second, by extracting features of the spectrum segment in each interval, which involves computing the average of certain functions of absorption at different wavelength, it reduces the effect of noise; third it offers the flexibility to utilize nonlinear features and higher order statistics to better capture the nonlinear relationships between sample absorbance spectrum and property; and finally it enables users to be creative and engineer combinations of several types of lower order and higher order features which synergistically produces more robust, better performing predictive models. In addition, a nonlinear regression method such as KPLS can be used in place of

PLS to further capture the nonlinear relationships, if any, between SPA features and properties of interest.

In this study, 8 different features/statistics are considered as candidate features to be modelled; four well-known statistics included aremean (μ), standard deviation (σ) or Variance (σ^2), skewness (γ), kurtosis (κ). Four other features that engineered for this study are average of first derivative of spectrum over an interval (AFD), average of second derivative of spectrum over an interval (ASD), slope of linear regression line (SLL) and coefficient of squared term for second order regression line (SSL). Mean and standard deviation represents the overall change in spectrum for a given interval size. Skewness, kurtosis and coefficients of regression lines provides information about different aspects of shape of the spectrum in an interval. Average of first and second derivatives give rate of change and rate of rate of change of absorbance spectrum with respect to wavelength. Note that the first and second derivatives of the absorbance spectrum, instead of spectrum itself, have been used for spectral analysis [104]–[106]. In this study best combination of features was identified by forward addition procedure.

6.3 Monte Carlo Validation and Testing procedures & MCVT-based performance indices:

To systematically test the proposed method and compare its performance with full PLS, Lasso and SiPLS models, a Monte Carlo validation and testing (MCVT) procedure is proposed, which is based on Monte Carlo cross-validation (MCCV) [107], but adapted for the purpose of comparing performances across different methods. The MCVT procedure is outlined in Figure 6.3. In each outer (*i.e.*, prediction) MC loop, the MC sampling approach is applied on the full dataset to partition it randomly into a combined calibration-

validation set and a test set based on specified proportions (e.g., proportions in Table 6.1). In each inner cross-validation MC loop, the MC sampling is applied on the training-validation set to generate a calibration set and a validation set, again, based on specified proportions such as those given in Table 6.1. A soft sensor model is built based on the training set and the validation set is projected onto the model with different model parameters to obtain a series of performance indices (*i.e.*, normalized root mean squared

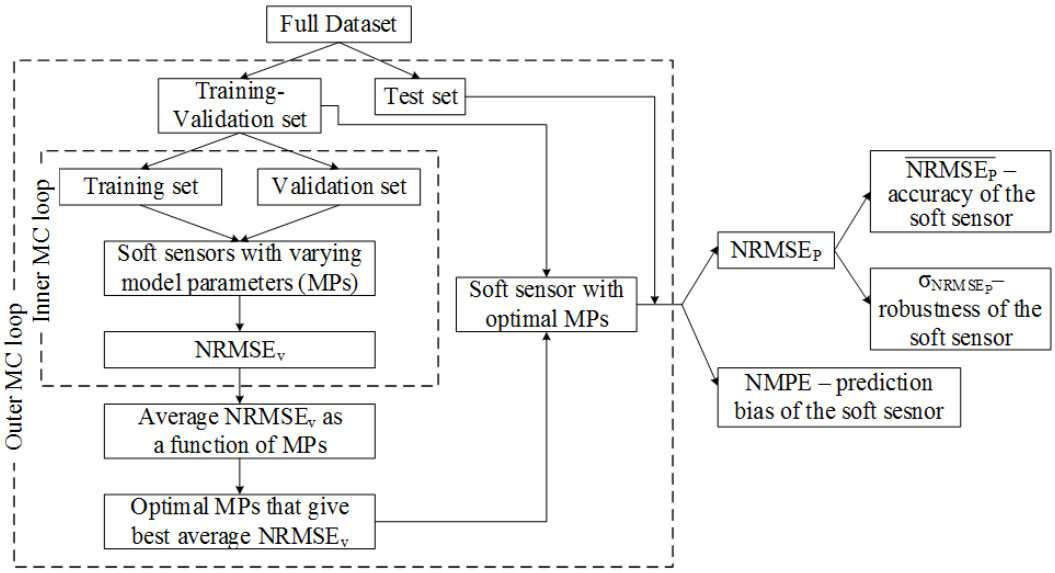


Figure 6.3 Flow diagram of the proposed Monte Carlo validation & testing procedure for comparing different soft sensor methods

errors ($NRMSE_V$) as defined in Eqn. (6.8) but they can be other indices) as a function of model parameters (MP's). Table 6.2 lists MP's to be optimized for each soft sensor method. It is worth noting that the spectrum partition/segmentation is an important parameter to be tuned or optimized for interval based methods SiPLS and SPA. Table 6.2 also indicates that the calibration processes of SiPLS and SPA are more computationally intensive than those of full PLS and Lasso based soft sensors. $NRMSE_V$ has a dimension of $p_1 \times p_2 \times \cdots \times p_n$ where p_i is the number of discrete values of model parameter i to be evaluated.

The inner MC loop is repeated M_V times (which is 100 in this work) to generate M_V $NRMSE_V$'s, which complete the inner MC loop. The $NRMSE_V$'s are averaged over the M_V MC runs to generate $\overline{NRMSE_V}$. The MP's (e.g., number of PC's, *etc*.) that result in the lowest $\overline{NRMSE_V}$ is used to build a prediction model using the combined calibration set. The test set is then projected onto the prediction model to generate the performance index $NRMSE_P$. The outer MC loop is repeated M_P times (which is 25 in this work), resulting in $M_P \times M_V$ inner (calibration) MC loops, to generate M_P $NRMSE_P$'s. The mean of $NRMSE_P$'s (*i.e.*, $\overline{NRMSE_P}$ as defined in Eqa.()) then can be used to evaluate the accuracy of the method while the standard deviation of $NRMSE_P$'s (*i.e.*, σ_{NRMSE_P} as defined in Eqn.(6.10)) can be used to assess the precision, or robustness/consistency of the method. Other performance indices can also be included, such as normalized mean prediction error ($NMPE$ as defined in Eqn. (6.11)) for quantifying prediction bias.

Table 6.1 Division of data into training, validation and test subsets

Dataset	Training (%)	Validation (%)	Test (%)	Total (%)
Corn	36 (45.0%)	28 (35.0%)	16 (20.0%)	80 (100%)
Gasoline	27 (45.0%)	21 (35.0%)	12 (20.0%)	60 (100%)
Pharma	263 (40.2%)	196 (29.9%)	196 (29.9%)	655 (100%)
Co-culture	21 (44.7%)	16 (34.0%)	10 (21.3%)	47 (100%)

Table 6.2 Parameters to be optimized for all methods

Method	Parameters to be calibrated	No. of parameters to be calibrated
Full PLS	No. of PC's	1
Lasso	λ	1
SiPLS	No. of segments the spectrum is divided into; Segments used in the model; No. of PC's	3
SPA	No. of segments the spectrum is divided into; Statistics/features used in the model; No. of PC's	3

6.3.1 MCVT based performance indices:

As mentioned previously, the following three MCVT-based performance indices are proposed to evaluate the performance of each soft sensor in this work as given below:

- **Normalized root mean squared error ($\boldsymbol{NRMSE}$) as percentage of the measurement range:**

$$\boldsymbol{NRMSE} = \frac{\sqrt{\frac{\mathbf{1}}{\boldsymbol{n}}\sum_{i=1}^{n}(\boldsymbol{y}-\hat{\boldsymbol{y}})_i^2}}{(\boldsymbol{y_{max}}-\boldsymbol{y_{min}})} \times \mathbf{100}\% \tag{6.8}$$

- **Average $\boldsymbol{NRMSE}$ ($\overline{\boldsymbol{NRMSE}}$):**

$$\overline{\boldsymbol{NRMSE}} = \frac{\sum_{i=1}^{M}\boldsymbol{NRMSE_i}}{\boldsymbol{M}} \tag{6.9}$$

- **Standard deviation of NRMSE ($\boldsymbol{\sigma_{NRMSE}}$):**

$$\boldsymbol{\sigma_{NRMSE}} = \sqrt{\frac{\sum_{i=1}^{M}(\boldsymbol{NRMSE_i}-\overline{\boldsymbol{NRMSE}})}{\boldsymbol{M}-\mathbf{1}}} \tag{6.10}$$

- **Normalized mean prediction error (NMPE) as percentage of the measurement mean:**

$$\boldsymbol{NMPE} = \frac{\sum_{i=1}^{n}(\boldsymbol{y}-\hat{\boldsymbol{y}})_i}{\sum_{i=1}^{n}\boldsymbol{y_i}} \times \mathbf{100}\% \tag{6.11}$$

- **Average $\boldsymbol{NMPE}$ ($\overline{\boldsymbol{NMPE}}$):**

$$\overline{\boldsymbol{NMPE}} = \frac{\sum_{i=1}^{M}\boldsymbol{NMPE_i}}{\boldsymbol{M}} \tag{6.12}$$

Where n is the total number of validation (n_V) or prediction (n_P) samples in each MC run, and M is the total number of MC runs during validation (M_V) or prediction (M_P).

Chapter 7. Case Studies, Results & Discussion of Feature based Soft Sensor

7.1 Case studies for feature based soft sensor:

In order to comprehensively compare SPA with PLS full model, SiPLS, Lasso and K-PLS four datasets from different areas are used in this work.

- **Corn dataset:** This dataset consists of 80 samples of NIR absorbance spectra from three spectrometers and corresponding property values of moisture, oil, protein and starch. Wavelength range is 1100nm to 2498nm at 2nm interval. In this paper moisture property and NIR spectra from mp6spec was used for study and comparison, any other property can also be used. More detailed discussion of the dataset can be found in [108].
- **Gasoline dataset:** This dataset consists of 60 samples of NIR absorbance spectra and corresponding octane number. Wavelength range is 900nm to 1700nm at 2nm interval. More detailed discussion of the dataset can be found in [109].
- **Pharmaceutical tablets dataset:** This dataset consists of 655 samples of NIR absorbance spectra of pharmaceutical tablets and corresponding values of total weight, hardness and Active pharmaceutical ingredients (API). Wavelength range is 600nm to 1798 nm at 2nm interval. In this paper API of the tablets was used for comparison and study. This dataset has already been divided into calibration, validation and test sets. More detailed discussion of the dataset can be found in [110]–[112].
- **Co-culture dataset:** This dataset consists of 47 samples of UV/Vis absorbance spectra of *E.coli* and *S. cerevisiae* co-culture with known individual cell mass concentration.

- In this data spectra were clearly separated into 6 groups. Wavelength range is 300nm to 900nm at 1nm interval. Detailed description of the dataset and the experimental design can be found in [73].

Spectra of all four datasets used for this study are shown in Figure 7.1.

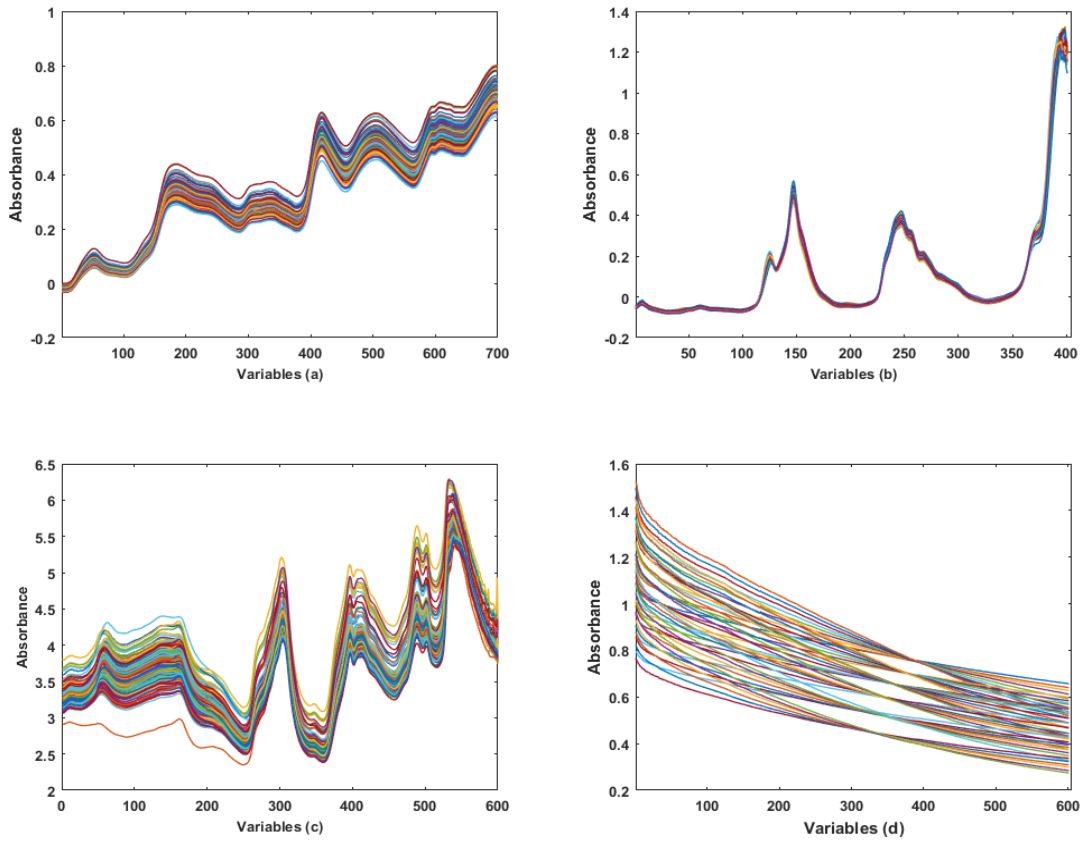


Figure 7.1 Spectra (a) Corn, (b) Gasoline, (c) Pharmaceutical, (d) Co-culture

For consistent and fair comparison across different datasets, the datasets were divided into calibration, validation and test sub-sets in consistent proportions. For small datasets (*i.e.*, corn, gasoline and co-culture datasets), about 20% of all samples were left out as test samples while the remaining ~80% of all samples were used for model training (*i.e.*, calibration and validation). For pharmaceutical dataset that has 655 samples, ~30%

of all samples were left out as test samples while ~70% of all samples were used for model training. In addition, to avoid model overfitting, based on the recommendation suggested by [107], about 45% of all training samples were left out for validation (*i.e.*, data used for the optimization of the model parameters) while the rest of the training samples (about 55% of all training samples) were used for calibration (*i.e.*, data actually used to build the models). Details of the data division for all datasets are given in Table 6.1. Literature [113] and author's experiences suggest that such division of training and validation (*i.e.*, ~55% vs. ~45%) results in models that are generally without overfitting issues. This is confirmed by the results of this work as discussed in details in next section. In addition, for small datasets such as most spectrum based datasets, the performance could be significantly affected by the data division (e.g., how many samples in training and testing respectively, and specific samples included in each group). To address this potential bias, a Monte Carlo validation and testing (MCVT) procedure, discussed in previous chapter, is proposed such that multiple (25 in this work) training and testing sets are randomly selected in each MC run and the average and standard deviation of the performances across different MC runs are used to robustly and fairly evaluate the soft sensor performance.

7.2 Results for feature based soft sensor:

All analyses followed the MCVT procedure discussed in the previous section (with $M_V = 100$ and $M_P = 25$) and MCVT-based performance indices (*i.e.*, $\overline{NRMSE}$, $\sigma_{NRMSE}$and $\overline{NMPE}$) were compared among different methods. For the proposed SPA soft sensor, after optimization during calibration/validation, the statistics and features selected for different datasets are listed in Table 7.1. As can be seen from Table 7.1, not all

statistics and features are selected for all datasets. This is because the features are selected to minimize $\overline{NRMSE_Y}$, which is similar to variable selection, where there is a trade-off between information added by the feature and noise and/or bias added by the feature. The average model sizes in terms of number of variables/features in the final soft sensor model over the 25 outer/prediction MC runs are listed in *Table 7.2*. It can be seen that all models with variable selection are substantially smaller than the full model. SPA has the smallest model size in three cases and the moderate model size in the rest two cases. In addition, it was found that the interval based methods, *i.e.*, SiPLS and SPA, are quite robust to spectrum segmentation (*i.e.*, the number of segments the spectrum is divided into). In the following, the findings from the comparison of different methods on different datasets is discussed.

Table 7.1 Statistics and features selected for different datasets

Dataset	Statistics and features selected
Corn dataset	μ, γ, κ, ASD, SLL, SSL
Gasoline dataset	μ, SLL
Pharmaceutical dataset	μ, γ, κ, AFD, ASD, SLL, SSL
Co-culture dataset	μ, σ

Table 7.2 Average number of variables/features of different soft sensors

Dataset	PLS	SiPLS	LASSO	SPA
Corn	700	152	141	84
Gasoline	401	84	14	30
Pharma	600	136	21	98
Co-cult (*E. coli*)	601	129	102	34
Co-cult (*S. cerevisiae*)	601	138	109	28

Corn data: Figure 7.2 shows comparison of all three indices, for validation and predication, obtained from all five approaches discussed in this paper. First, if we compare performance indices (*i.e.*, $\overline{NRMSE}$, σ_{NRMSE}, and $\overline{NMPE}$) of validation vs. prediction, there is a general trend of performance deterioration from validation to prediction, which is expected as the model parameters were optimized based on the validation data. However, the performance deteriorations are not drastic, indicating that there is no obvious overfitting of models. The second observation is that in general variable selection improves soft sensor performance. Although $\overline{NMPE}_P$'s of Lasso and SiPLS are slightly higher than the full model in this particular case study, the error is insignificant – only less than 0.1% of the measurement mean. The third observation is that for SiPLS, the model prediction performances are noticeably worse than that of the validation, especially σ_{RMSE}, and $\overline{NMPE}$. The likely reason is that wavelengths selected by Lasso or SiPLS to optimize the prediction performance based on the training (*i.e.*, calibration and validation) data may be "tilted" to over-fit or capture the noise or unknown disturbances contained in the calibration data. As a result, when the model is extrapolated or applied to new samples, the performance may deteriorate noticeably. Finally, SPA outperforms Lasso and SiPLS in all performance metrics. We believe the likely reason is that SPA does not discard any wavelength. Instead, SPA extracts features over the whole spectrum, making it more robust against performance degradation from validation to prediction. Overall, SPA-based soft sensor provides the best performance.

Gasoline data: As shown in *Figure 7.3*, for the gasoline data, similar trend of performance deterioration from validation to prediction is observed as expected. But again, the insignificant difference in $\overline{NRMSE}$ suggests no obvious overfitting of the

models. In addition, the performance of SiPLS deteriorate noticeably from validation to prediction, especially σ_{NRMSE} and $\overline{NMPE}$, which increased by more than three folds and eight folds, respectively. In contrast, the performances of Lasso and SPA are more consistent with reasonable increase in σ_{RMSE} and $\overline{NMPE}$. Strictly speaking SPA out performs all the other approaches under consideration.

Pharmaceutical tablet data: As shown in Figure 7.4, for the pharmaceutical data, the performances of Lasso, SiPLS and SPA are similar, which are slightly better than that of the full model in terms of $\overline{NRMSE}$ and $\overline{NMPE}$ but slightly higher σ_{NRMSE}. SPA gives the smallest $\overline{NMPE}$ among all four models.

Co-culture data: For the coculture data, two sets of models were built to predict the concentrations of *E. coli* and *S. cerevisiae* independently. Separate models were used because the concentrations of the two species are properties that are not supposed to be correlated to each other. As shown in Figure 7.5, for *E. coli*, Lasso and SiPLS actually perform worse than the full model in prediction, but SPA performs noticeably better than the full model. Although the $\overline{NMPE}$ of SPA is higher than that of the full model, its value of less than 0.05% of the measurement mean indicates already very accurate predictions. Therefore it can be safely concluded that SPA outperforms other approaches for this dataset as well.

For *S. cerevisiae*, Lasso again performs worse than the full model, while SiPLS and SPA performing slightly better than the full model as shown in Figure 7.6. For $\overline{NMPE}$, again, the values from all models are less than 0.05% of the measurement mean indicates high accuracy of predictions from all models. When models for both species are considered, SPA performs better than Lasso and SiPLS. This further supports the SPA

methodology of not removing any wavelength but rather extracting features from the full spectrum.

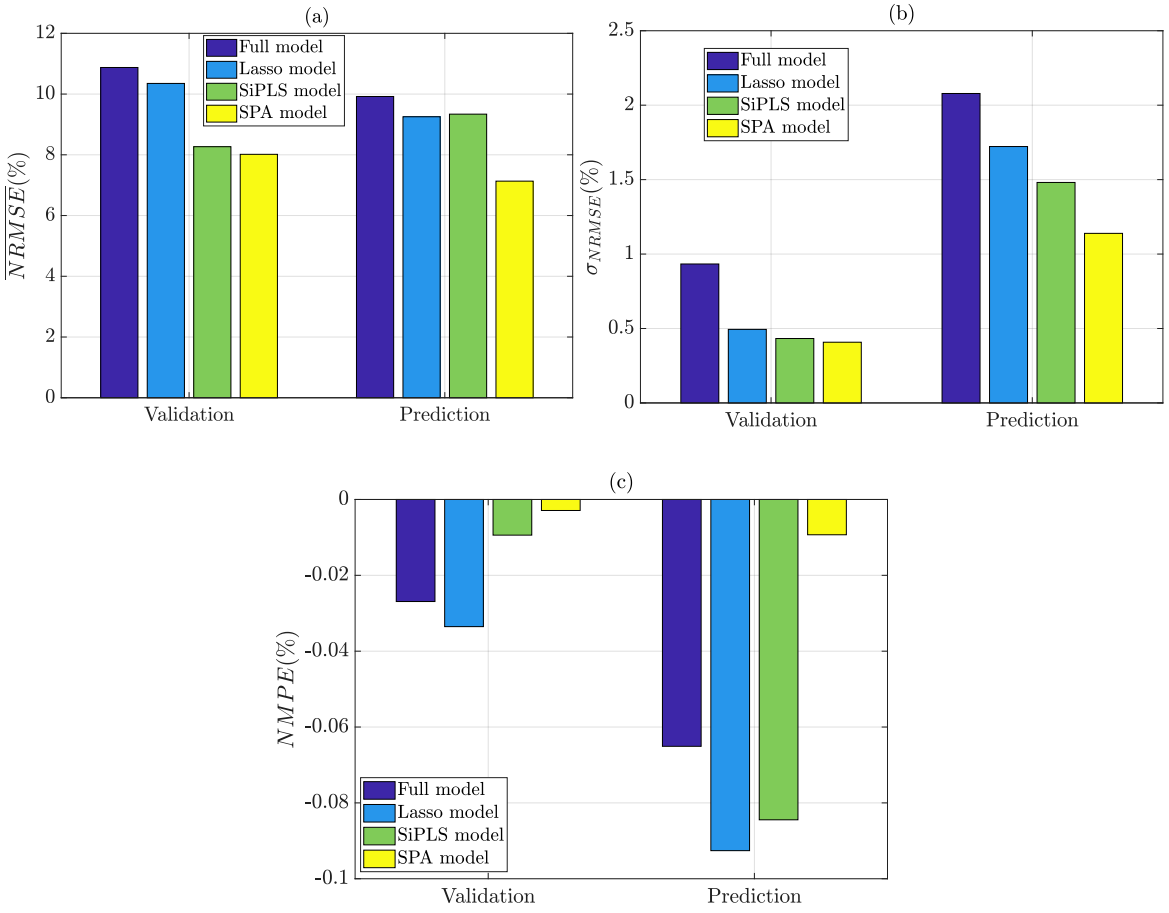


Figure 7.2 Comparison of soft sensors using corn data (moisture): (a) $\overline{NRMSE}$, (b) σ_{NRMSE}, (c) $\overline{NMPE}$

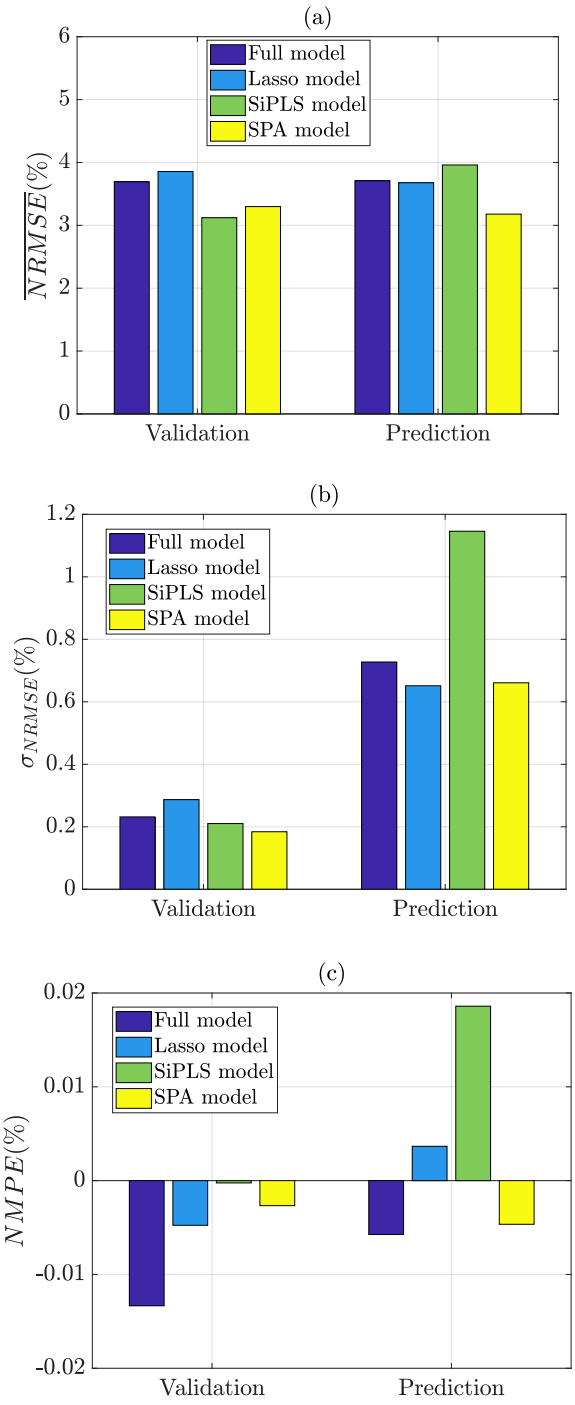


Figure 7.3 Comparison of soft sensors using gasoline data: (a) $\overline{NRMSE}$; (b) σ_{NRMSE}; (c) $\overline{NMPE}$

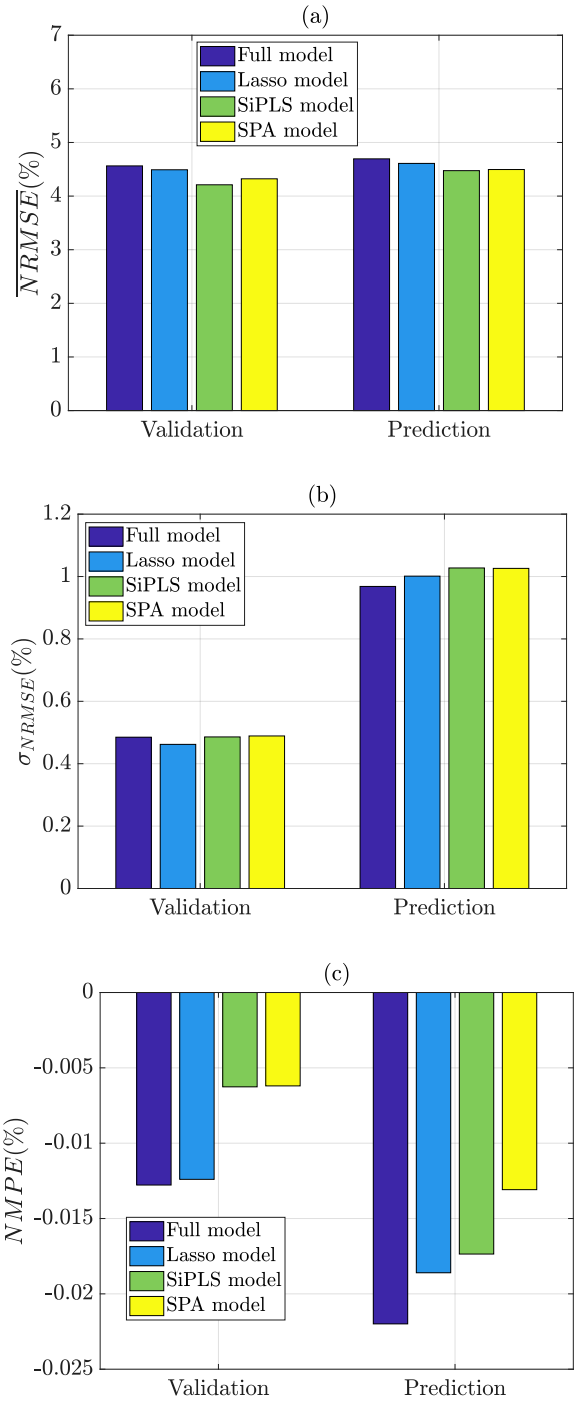


Figure 7.4 Comparison of soft sensors using pharmaceutical data: (a) $\overline{NRMSE}$, (b) σ_{NRMSE}, (c) $\overline{NMPE}$

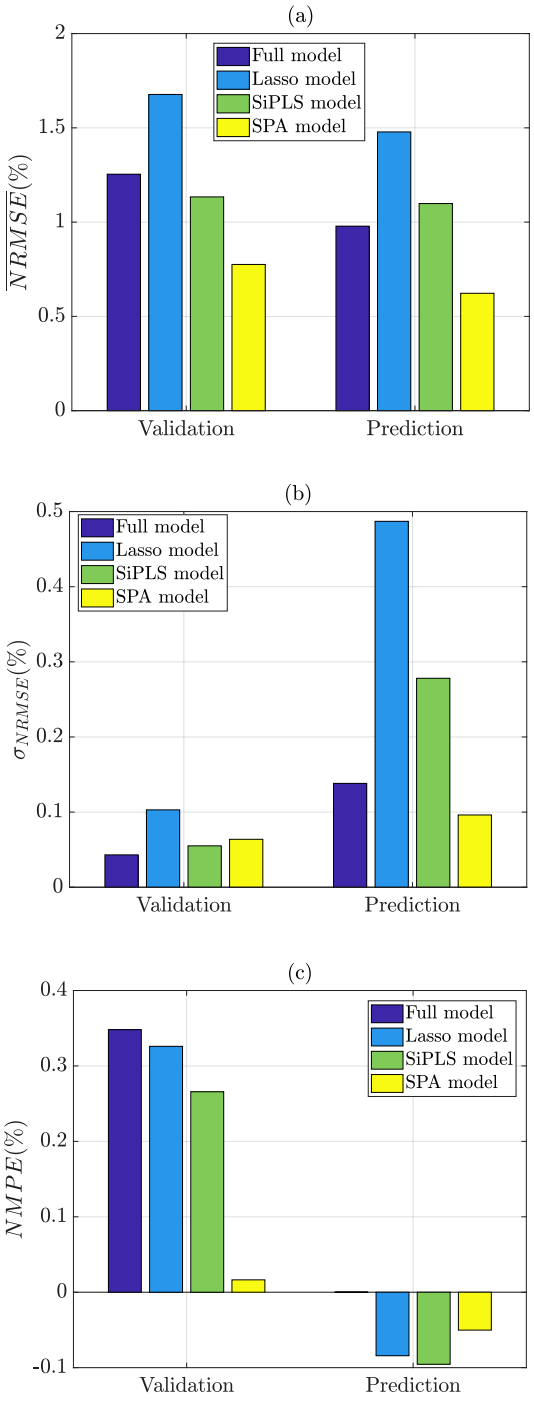


Figure 7.5 Comparison of soft sensors using co-culture (E.coli) data: (a) $\overline{NRMSE}$, (b) σ_{NRMSE}, (c) $\overline{NMPE}$

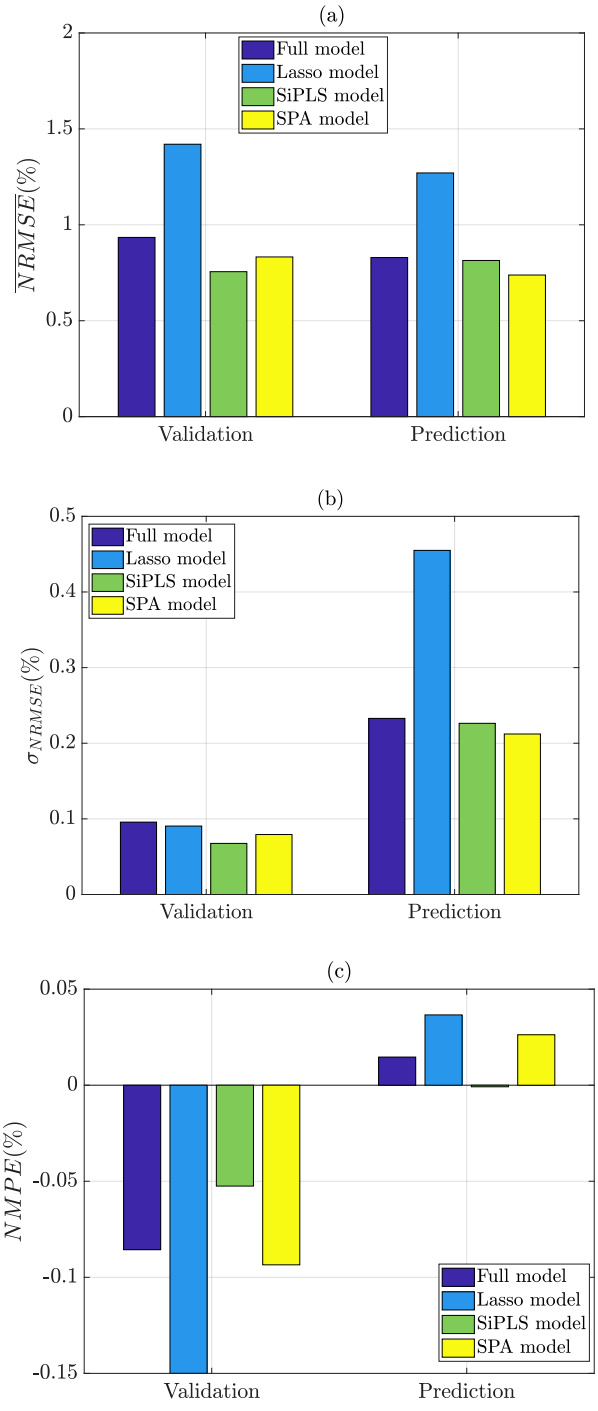


Figure 7.6 Comparison of soft sensors using co-culture (S. cerevisiae) data: (a) $\overline{NRMSE}$, (b) σ_{NRMSE}, (c) $\overline{NMPE}$

7.3 Discussion on feature based soft sensor:

Next, potential reasons for the improved performance from SPA feature-based soft sensor will be discussed. For the corn data, Figure 7.7 (a) shows the predicted vs. measured moisture content, which confirms that SPA performs better than full PLS, Lasso and SiPLS across the whole property region. More importantly, it should be noted that SPA performs especially better at extreme or boundary regions, as highlighted by red ellipses in Figure 7.7 (a), where the number of samples are usually fewer than other regions. For example, when the moisture content is below 9.5, the predictions from the full PLS, Lasso and SiPLS models are widespread (*i.e.*, large σ_{NRMSE}) and with significant bias (*i.e.*, high $\overline{NMPE}$). Similar observations are found for the gasoline (Figure 7.7 (b)). It can be postulated that the wavelengths selected by Lasso or SiPLS to optimize the performance based on the calibration data may be "tilted" to over-fit or capture the noise or unknown disturbances contained in the calibration data, which are dominated by samples from the dense regions. As a result, the prediction performance of SiPLS may deteriorate significantly when the model is extrapolated or applied to samples from the sparse regions. As for the full PLS model, we believe the reason is because its performance is optimized by $NRMSE$, which means the spare regions with fewer samples would weigh less than the dense regions with more samples in determining the model parameters. As for SPA, although it is also optimized by $NRMSE$, it seems that the statistics and features extracted based on different regions of the full spectrum can

alleviate this situation and the resulted model can extrapolate much better to the sparse regions with fewer samples than the models from PLS, Lasso and SiPLS.

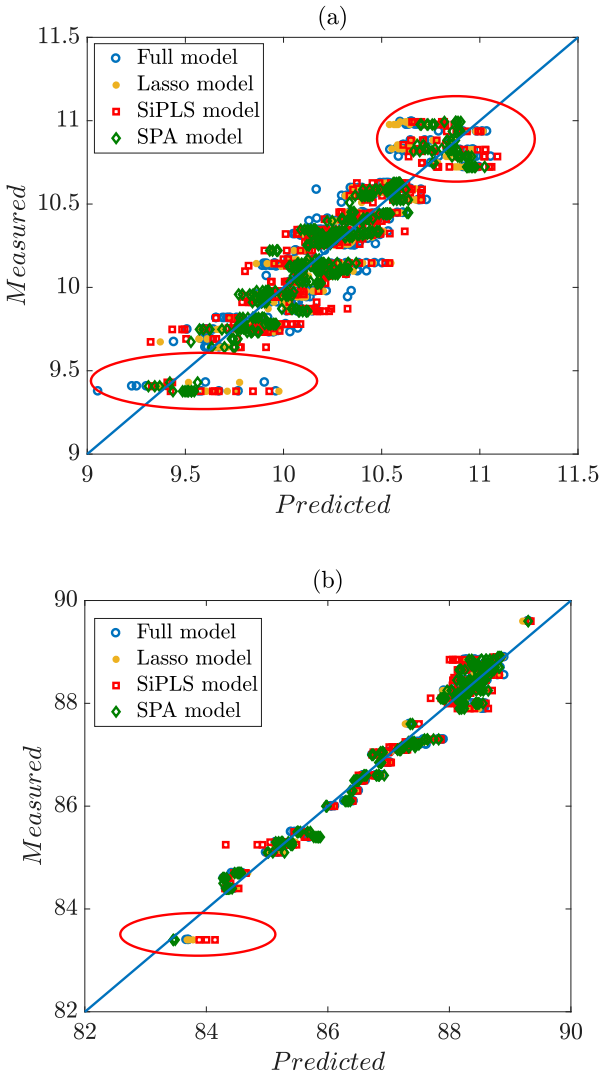


Figure 7.7 Comparison of predicted vs. measured properties from different soft sensors using (a) corn data and (b) gasoline data. Red ellipses highlight the regions where SPA performs significantly better than the full PLS, Lasso and SiPLS models.

Until this point, only linear soft sensors are compared, although, strictly speaking, the mapping from the original spectrum to the SPA features is not linear. In this study the potential of KPLS as a nonlinear soft sensor for spectroscopic data analysis applications has been explored. For KPLS with a Gaussian kernel, the number of variables in the

feature/kernel space equals the number of training samples, which is usually significantly larger than the number of variables under normal circumstances. However, this is not true for many spectroscopic data analysis applications as discussed previously. In those cases, KPLS actually shrinks the variable dimension in the kernel space. It is worth noting that this work is not intended to invalidate the merits of KPLS, but rather a case study to see if nonlinearity in the spectroscopic data can be captured for improving soft sensor performance given the severe constraint of the limited number of samples. In this work two scenarios are studied: the first scenario is to apply KPLS on the original full spectra; the second scenario is to apply KPLS on SPA features. The same MCVT procedure is followed to tune KPLS parameters, including the Gaussian kernel parameter σ and number of PC's. For the first scenario where KPLS is applied on the full spectra of each dataset, the performance of KPLS is poor for all four datasets. Table 7.4 compares the $\overline{NRMSE}$ and $\overline{NMPE}$ of 25 MC prediction runs for the pharmaceutical dataset, which is the largest dataset studied in this work with 263 training samples. Table 7.4 shows that KPLS performs even worse than full PLS. This is not surprising due to the severe constraint of limited number of samples. Therefore, the poor performance of KPLS does not indicate that there is no nonlinearity exists in the data, instead, it can only be said that KPLS cannot overcome the deficiency of variable shrinkage (instead of variable expansion in a regular KPLS application) due to the smaller number of samples (263) than that of variables (600). In the second scenario, we apply KPLS on top of SPA and term it SPA-KPLS. In other words, KPLS is replacing PLS and applied to SPA features. Again, the same MCVT procedure is followed to tune the KPLS parameters. The performances of SPA-KPLS on the four datasets are compared to PLS based SPA, and

the results are listed in Table 7.3. Table 7.3 shows that KPLS does not improve the performance of SPA-based soft sensor in three out of four datasets (*i.e.*, corn, gasoline and pharmaceutical datasets), which we believe can be attributed to the small number of training samples. However, KPLS does help in predicting *E. coli* and *S. cerevisiae* concentrations in the co-culture dataset with only 21 samples for each strain. Although author do not have a definitive answer to explain this, it is believed that this is due to the significant similarity between the absorbance of the two strains, which makes the nonlinear interactions between the absorbance of the two strains an important factor in predicting their individual concentrations. In other words, the nonlinearity captured by KPLS outweighs the deficiency of dimension shrinkage due to limited samples.

Table 7.4 Prediction Performance of KPLS on the Pharmaceutical dataset compared to PLS & SPA

	PLS	SPA	KPLS
$\overline{NRMSE}$	4.69%	4.49%	5.03%
$\overline{NMPE}$	-0.022%	-0.013%	-0.032%

Table 7.3 Prediction Performance of SPA-KPLS compared to SPA

		Corn	Gasoline	Pharma	*E.coli*	*S. cerevisiae*
$\overline{NRMSE}$	SPA	7.31%	3.18%	4.50%	0.62%	0.74%
	SPA-KPLS	8.04%	3.40%	4.73%	0.48%	0.44%
$\overline{NMPE}$	SPA	-0.0093%	-0.0047%	-0.013%	-0.050%	0.026%
	SPA-KPLS	-0.055%	-0.011%	-0.0034%	-0.015%	0.033%

Note: Another variation of the proposed feature based soft sensor developed by author is given in appendix C.

Chapter 8. Overall framework, Conclusions & Future work

8.1 Overall framework:

In the previous segments entire recipe for building a scalable industrial IoT enabled smart manufacturing testbed was presented. In the section below all the components of the testbed will be combined to present and overall framework that can be applied to build SM testbeds.

Firstly, an important unit operations for any manufacturing process needs to be identified, for this study multi-stage centrifugal pump, next different humanizing easily measured properties specific to unit operation is identified. Different properties can be combined, vibrations corresponding to human touch & video camera as human sight. IIoT sensors were used to measure these properties quantifiably. Design data collection, transfer & storage architecture, mimicking human nervous system. Study data characteristics and noise characteristics to validate quality of data as well as to obtain imperceptible insights about system under consideration. Next design a data driven models to capture process behavior *i.e*. flowrate & RPM prediction using vibrations & video data. Next the setup a sensor fault detection approach & design redundancy for during the same, proposed reconstruction approach during long missing data using adjacent sensors & reconstructed signal can be further used to predict properties of interests. Finally if different linear- non-linear models do not capture process information to sufficient accuracy then a new set of features shall be engineered to provide more robust & accurate models, Feature-based soft sensors. Schematic of the overall

framework for establishing IIoT enabled smart manufacturing is given in Figure 8.1. In green rectangle envelopes all the steps needed to build the testbed this signifies that for successful SM system knowledge is of immense importance as it helps optimize the process and significantly reduce amount of resources required for such implementation, as was demonstrated in the study. Blue rectangle signifies big data analytics, efficient big data technologies will significantly contribute for data characterization & process understanding. Big data technologies can be improved with established process knowledge & process knowledge in turn be improved by enabling big data technologies this synergy will further between analytics technologies & process knowledge accelerates learning & is imperative for industrial scale smart manufacturing. This understanding is signified in schematic by overlapping lower ends of blue & green rectangles.

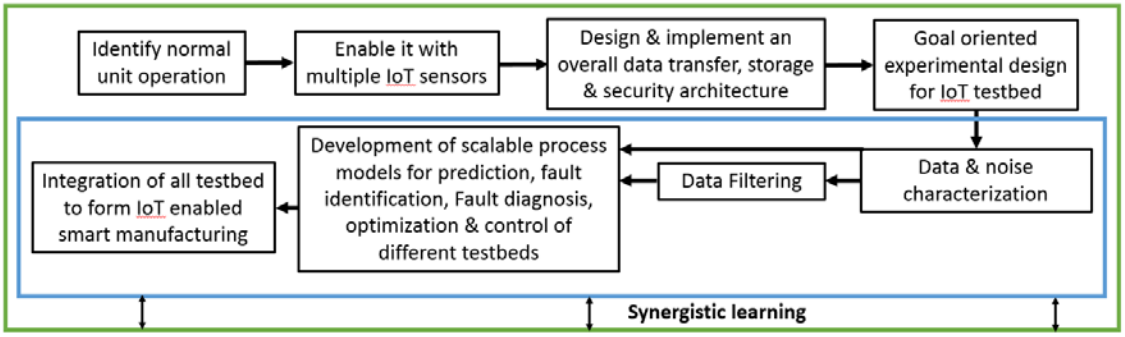


Figure 8.1 Overall Framework for IIoT enabled Smart Manufacturing; Green rectangle=process knowledge; Blue rectangle= Big data analytics

8.2 Conclusions:

This work presents systematic study for setting up an internet of things enabled smart manufacturing testbed for industrial applications while incorporating big data analytics. Study first characterizes big data & shows how IIoT SM testbed for industrial applications is build & combined with big data analytics for real-time process monitoring

& fault detection. IIoT SM testbed was build using multi-stage centrifugal pumping system & non-invasive IIoT vibration sensors. IIoT sensor data characteristics were systematically studied and several ways to overcome such challenges were proposed with specific example. Internet network properties were also studied to uncover several aspects of veracity in the datasets. It was shown how observed data characteristics relates to 4v's of big data & how data involved with IIoT enabled testbed qualify as big data. Therefore techniques and tools developed in this study are contribution to the big data analytics. Next, by applying raw vibration data on large long short term memory (LSTM) NN models it was shown how rote application of machine learning/ deep learning algorithms on complex data may force modeler to use highly complex non-intuitive models which requires extremely large amount of computation resources. Moreover in some cases (as presented here) such models fails to provide accurate predictions and may lead to wrong conclusion. Next, after extensive study of multiple approaches it was suggested that lomb's algorithm (used in astronomy community) shall be used for obtaining frequency response of any un-equally spaced vibration signals in order to overcome unequally sampled data IIoT sensor data characteristic. Next by applying spectrum resulting from optimized Lomb's algorithm (optimized using primary system knowledge (frequency feature extraction)) on deep neural network models it was shown that incorporation of primary system knowledge and incorporating some data characteristic knowledge (unequally spaced data)) can significantly reduce model complexity and improves model prediction performance as deep neural network resulted in a fairly accurate soft sensors for flowrate & RPM prediction. However integration of basic system knowledge may not provide necessary insights about scientific

fundamentals associated with underlying process that is being modelled. Moreover there are many limitations associated with highly parametrized models like NN training & optimization with big data, model interpretation, and model consistency *etc*. that can be mitigated by developing a modelling structure which has less parameterized more robust models. More detailed advanced data analytics carried out by using principal component analysis for data mining & data visualization and was combined with system & process knowledge for identifying scientific fundamental relationship underlying the developed process models. It was shown how presence of imperceptible relationships in high density noisy data can be uncovered and be targeted for modelling. Using fundamental and operation understanding of process a hierarchical modelling approach for RPM & flowrate prediction was proposed. In this approach fundamental understanding of process was used for accurate modelling of RPM using binary matrix approach. Next for flowrate model fundamental knowledge was combined with statistics for identification of important variables by introducing use of a new statistic inverse of coefficient of variation. Flowrate model was further improved by studying and addressing noise characteristics with the help of moving average. Resulting models were more accurate, highly consistent, robust & easy to train & optimize. Thus overall development process of predictive RPM & flowrate models showed that incorporation of system and process knowledge in combination of advanced data analytics is extremely important for development of robust, accurate & simple models that can be used for real-time industrial processes. Next, process monitoring framework for IIoT enabled SM testbed was reinforced with signal reconstruction to addressing missing signal data characteristic challenge of IIoT sensors by proposing a new combined PCA & PLS approach. This

approach be easily extended for fault identification & have high potential for fault diagnosis. Finally for the cases were no direct relations can be modelled to sufficient prediction performance a new feature based soft sensor modelling approach for spectrum data by engineering relevant high & low order statistics. It was proven that models developed by combinations of these statistics are more accurate, more robust & less biased than any of the other approaches used currently using raw dependent variables. All the approaches & models were developed for non –invasive IIoT vibration sensor enabled multi-stage centrifugal pumps however author believes all the approaches like hierarchical modelling, important vibration frequency identification using ICV, combined PCA & PLS for reconstruction & moving average for dealing Gaussian noise can easily be applied for other similar applications as well.

For feature based soft sensor, although variable selection in general could significantly improve soft sensor performance and reduce model complexity, there are potential pitfalls. As demonstrated by multiple case studies in this work, variable selection methods can be sensitive to the choice of training data and their performance could deteriorate noticeably when applied to test samples. Author believes the possible reason is that the wavelengths selected (or wavelengths removed for that matter) to optimize the performance based on the calibration (*i.e.*, training and validation) data may be “tilted” to over-fit or capture the noise or unknown disturbances contained in the calibration data. As a result, the model prediction performance may deteriorate significantly when the model is extrapolated or applied to new samples. To address this limitation, we propose a feature-based soft sensor approach by adapting the idea of SPA-based process monitoring framework we developed previously. Instead of selecting

certain wavelengths or wavelength segments, the SPA feature-based soft sensor considers the whole spectrum, which is divided into segments, and extracts different features over each spectrum segment to build the soft sensor. In this way, there is no removal or exclusion of any wavelength or spectrum segment. As demonstrated in multiple case studies in this work, the proposed SPA feature-based soft sensor in general outperforms the original absorbance based soft sensor (*i.e.*, the full PLS soft sensor) as well as the variable selection method Lasso or SiPLS based soft sensor in terms of $\overline{NRMSE}$, $\sigma NRMSE$, and $\overline{NMPE}$. The SPA feature-based soft sensor is more robust than Lasso and SiPLS based soft sensor as evidenced by the smaller performance dip from validation to prediction. In addition, the SPA feature-based soft sensor can extrapolate much better to the sparse regions with fewer samples than the soft sensors based on full PLS, Lasso or SiPLS. Author believes the main reasons for the good performance and robustness of the SPA based soft sensor are due to the following two factors: (1) features of spectrum segments correlate better to the property of interest (in a linear fashion through PLS) than the original spectrum or selected wavelengths; (2) inclusion of all information from the whole spectrum without removal or exclusion of any wavelength or wavelength segment enhances the robustness of the soft sensor. Finally, for small datasets such as most spectrum based datasets, the soft sensor performance could be significantly affected by the data division (e.g., how many samples in training and testing respectively, and specific samples included in each group). To address this potential bias, Monte Carlo validation and testing (MCVT) procedure was proposed such that multiple (25 in this work) training and testing sets are randomly selected in each MC run and the average and standard deviation of the performances across different MC runs are used to robustly and

fairly evaluate the soft sensor performance across different datasets, which are generally applicable for model comparison in other applications. Although linear soft sensor methods are much preferred in most applications for their simplicity and interpretability, potential of nonlinear KPLS was tested by applying it to both original spectra and SPA features. The results indicate that when the number of samples are severely limited, applying KPLS to full spectra is not a good solution compared to PLS. However, when KPLS is applied to SPA features, although still suffering the deficiency of small sample size, the results are much improved. The results of the co-culture dataset justify the advantage of KPLS over PLS when there are potentially strong nonlinear interactions.

9.3 Future work:

Development of new IIoT enabled testbeds for smart manufacturing is still at its incipient and requires much more research. It will be extremely useful if pumping system in consideration can be connected some upstream & downstream unit operations, more specifically a reactor at pumps downstream & a storage tank on pump's upstream. It will be interesting to capture process information like reactor conversion rate, dead zones or non-reacting zones, mixing patterns *etc*. using different types of IIoT sensors like surface temperature sensors, surface acoustics sensors, pressure sensors, IR cameras *etc*. Several different types of modelling frameworks can be designed based on process requirements. Author believes that the impact of system knowledge & operating conditions will guide development framework further. After development of upstream & downstream testbeds, rules and techniques for integration & sharing of information across testbeds also needs to be established and shall be studied.

For multistage centrifugal pump testbed although model for reconstruction of spectrum has been developed for missing signals. Detailed study to develop complete process & sensor fault identification & diagnosis approaches needs to be carried out. For more realistic & complete fault diagnosis & testing, experimental data collection by inducing known faults on the developed testbed needs to be carried out. Moreover models developed in the study perform extremely well and are efficient, proposed feature based approach can be incorporated to further reduce number of variables. As discussed before proposed modelling approach is applicable on most of the rotating mechanical systems, however more complex rotating machines relationships between process information and vibrations can be extremely complex and needs to be identified for such cases hierarchical modelling with ensemble of non-linear models like KPLS, ANNs, random forest *etc*. can be used to cover entire range of operations.

More detailed study regarding the noise behavior of the data collected over different types of connection protocols, networks (wired or wireless, public or private *etc*.) and with different type of micro-controllers should be carried out.

Application of proposed feature based soft sensor using statistics pattern analysis is has huge potential in almost all fields of science so that more complex relationships can be established using simpler, consistent & easily interpretable models. More detailed study on how to select the features or how to engineer the features more systematically with less computation is required & will add immense value. One potential way is to first identify several bands of wavelengths most influencing property of interest & then using those properties identify set of features which in combination results into improved model. More study is also desirable to establish fundamental relations or theories for

rational selection of features. Further study is required for integrating non-linear regression methods such as ANN & kernel-based approaches into features-based soft sensor, especially for cases where large number of samples are available and/or potentially strong nonlinear interactions exists. This study will help further understand extensions & limitations of feature based approaches in non-linear modelling.

Bibliography

[1] J. Davis, T. Edgar, R. Graybill, P. Korambath, B. Schott, D. Swink, J. Wang, and J. Wetzel, “Smart Manufacturing,” *Annu. Rev. Chem. Biomol. Eng.*, vol. 6, no. 1, pp. 141–160, Jul. 2015.

[2] J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli, “Smart manufacturing, manufacturing intelligence and demand-dynamic performance,” *Comput. Chem. Eng.*, vol. 47, pp. 145–156, Dec. 2012.

[3] D. Shah, J. Wang, and Q. P. He, “An Internet-of-things Enabled Smart Manufacturing Testbed,” 2019.

[4] P. Zheng, H. wang, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarok, S. Yu, and X. Xu, “Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives,” *Front. Mech. Eng.*, vol. 13, no. 2, pp. 137–150, Jun. 2018.

[5] Y. Zhang, G. Zhang, J. Wang, S. Sun, S. Si, and T. Yang, “Real-time information capturing and integration framework of the internet of manufacturing things,” *Int. J. Comput. Integr. Manuf.*, vol. 28, no. 8, pp. 811–822, Aug. 2015.

[6] F. Tao and Q. Qi, “New IT Driven Service-Oriented Smart Manufacturing: Framework and Characteristics,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 1, pp. 81–91, Jan. 2019.

[7] P. Kadlec, B. Gabrys, S. S.-C. & chemical engineering, and undefined 2009, "Data-driven soft sensors in the process industry," *Elsevier*.

[8] H. J. Galicia, Q. P. He, and J. Wang, "A reduced order soft sensor approach and its application to a continuous digester," *J. Process Control*, vol. 21, no. 4, pp. 489–500, 2011.

[9] H. J. Galicia, Q. P. He, and J. Wang, "A reduced order soft sensor approach and its application to a continuous digester," *J. Process Control*, vol. 21, no. 4, pp. 489–500, Apr. 2011.

[10] J. R.-J. of M. L. Research and undefined 2003, "Overfitting in making comparisons between variable selection methods," *jmlr.org*.

[11] C. Andersen, R. B.-J. of Chemometrics, and undefined 2010, "Variable selection in regression—a tutorial," *Wiley Online Libr.*

[12] Z. X. Wang, Q. P. He, and J. Wang, "Comparison of variable selection methods for PLS-based soft sensor modeling," *J. Process Control*, vol. 26, pp. 56–72, Feb. 2015.

[13] M. Adams, *Rotating Machinery Vibration*. CRC Press, 2009.

[14] E. P. Carden and P. Fanning, "Vibration Based Condition Monitoring: A Review," *Struct. Heal. Monit. An Int. J.*, vol. 3, no. 4, pp. 355–377, Dec. 2004.

[15] D. Lyon, "JOURNAL OF OBJECT TECHNOLOGY The Discrete Fourier Transform, Part 4: Spectral Leakage," *J. Object Technol.*, vol. 8, no. 7, pp. 23–34.

[16] N. Tandon and A. Choudhury, "A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings," *Tribol. Int.*, vol. 32, no. 8, pp. 469–480, Aug. 1999.

[17] A. Devices, “ADXL345 (Rev. 0).”

[18] J. Mankar, C. Darode, K. Trivedi, M. Kanoje, and P. Shahare, “REVIEW OF I2C PROTOCOL,” *Int. J. Res. Advent Technol.*, vol. 2, no. 1, 2014.

[19] F. Leens, “An introduction to I ^{2} C and SPI protocols,” *IEEE Instrum. Meas. Mag.*, vol. 12, no. 1, pp. 8–13, Feb. 2009.

[20] A. K. Oudjida, M. L. Berrandjia, R. Tiar, A. Liacha, and K. Tahraoui, “FPGA implementation of I^2C & SPI protocols: A comparative study,” in *2009 16th IEEE International Conference on Electronics, Circuits and Systems - (ICECS 2009)*, 2009, pp. 507–510.

[21] M. Maksimović, V. Vujović, N. Davidović, … V. M., and undefined 2014, “Raspberry Pi as Internet of things hardware: performances and constraints,” *researchgate.net*.

[22] “Arduino vs Raspberry Pi: A Detailed Comparison.” [Online]. Available: https://beebom.com/arduino-vs-raspberry-pi/. [Accessed: 19-Mar-2018].

[23] “Raspberry Pi: 11 reasons why it’s the perfect small server | ZDNet.” [Online]. Available: http://www.zdnet.com/article/raspberry-pi-11-reasons-why-its-the-perfect-small-server/. [Accessed: 19-Mar-2018].

[24] “Raspberry Pi: What are its limitations? | ITProPortal.” [Online]. Available: https://www.itproportal.com/2013/04/25/raspberry-pi-what-are-its-limitations/. [Accessed: 19-Mar-2018].

[25] C. Anglano, “Forensic analysis of WhatsApp Messenger on Android smartphones,” *Digit. Investig.*, vol. 11, no. 3, pp. 201–213, Sep. 2014.

[26] “Epoch Converter - Unix Timestamp Converter.” [Online]. Available:

https://www.epochconverter.com/. [Accessed: 19-Mar-2018].

[27] "Unix Time Converter." [Online]. Available: https://www.unixtimeconverter.io/. [Accessed: 19-Mar-2018].

[28] J. O'Brien, *Frequency-domain control design for high-performance systems*. The Institution of Engineering and Technology, 2012.

[29] W. T. Cochran, J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, D. E. Nelson, C. M. Rader, and P. D. Welch, "What is the fast Fourier transform?," *Proc. IEEE*, vol. 55, no. 10, pp. 1664–1674, 1967.

[30] D. M. Pirouz, "An Overview of Partial Least Squares," *SSRN Electron. J.*, Oct. 2006.

[31] P. A. Chemistry, "Partial least-squares regression: a tutorial," vol. 186, 1986.

[32] I. T. Jolliffe, "Principal Component Analysis and Factor Analysis," Springer, New York, NY, 1986, pp. 115–128.

[33] L. Eldén, "Partial least-squares vs. Lanczos bidiagonalization-I: Analysis of a projection method for multiple regression," *Comput. Stat. Data Anal.*, vol. 46, no. 1, pp. 11–31, 2004.

[34] J. Park and I. W. Sandberg, "Universal Approximation Using Radial-Basis-Function Networks," *Neural Comput.*, vol. 3, no. 2, pp. 246–257, Jun. 1991.

[35] B. Kosko, "Fuzzy systems as universal approximators," *IEEE Trans. Comput.*, vol. 43, no. 11, pp. 1329–1333, 1994.

[36] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[37] M. A. Nielsen, "Neural Networks and Deep Learning." Determination Press, 2015.

[38] H. Jaeger and H. Jaeger, “A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach,” 2002.

[39] S. W. Doebling, C. R. Farrar, M. B. Prime, and D. W. Shevitz, “Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review,” Los Alamos, NM, May 1996.

[40] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[41] G. Chen, “A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation,” Oct. 2016.

[42] H. Wold, “Partial Least Squares,” in *Encyclopedia of Statistical Sciences*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2006.

[43] H. Wold, “Soft Modelling by Latent Variables: The Non-Linear Iterative Partial Least Squares (NIPALS) Approach,” *J. Appl. Probab.*, vol. 12, no. S1, pp. 117–142, 1975.

[44] S. de Jong, “SIMPLS: An alternative approach to partial least squares regression,” *Chemom. Intell. Lab. Syst.*, vol. 18, no. 3, pp. 251–263, Mar. 1993.

[45] “PLS regression.” .

[46] F. Lindgren, P. Geladi, and S. Wold, “The Kernel algorithm for PLS,” *J. Chemom.*, vol. 7, no. April 1992, pp. 45–59, 1993.

[47] X. Yang, “Artificial Neural Networks,” in *Handbook of Research on Geoinformatics*, IGI Global, 1AD, pp. 122–128.

[48] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines.”

[49] V. W. Porto and D. B. Fogel, “Alternative neural network training methods [active sonar processing],” *IEEE Expert*, vol. 10, no. 3, pp. 16–22, Jun. 1995.

[50] C. J. C. Burges, N. Hamilton, C. Burges, T. Shaked, E. Renshaw, and G. Hullender, “Learning to Rank using Gradient Descent Light-Ion-Induced Multifragmentation View project Learning to Rank using Gradient Descent.”

[51] L. Bottou, “Stochastic Gradient Learning in Neural Networks.”

[52] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for Hyper-Parameter Optimization.” pp. 2546–2554, 2011.

[53] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, “Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets,” May 2016.

[54] J. Musial, M. V.-… chemistry and physics, and undefined 2011, “Comparing the effectiveness of recent algorithms to fill and smooth incomplete and noisy time series,” *atmos-chem-phys.net*.

[55] S. Vaseghi, “Interpolation,” in *Advanced Digital Signal Processing and Noise Reduction*, Chichester, UK: John Wiley & Sons, Ltd, 2001, pp. 297–332.

[56] R. Patro and C. Kingsford, “Data-dependent bucketing improves reference-free compression of sequencing reads.,” *Bioinformatics*, vol. 31, no. 17, pp. 2770–7, Sep. 2015.

[57] L. Margueritte, P. Markov, L. Chiron, J.-P. Starck, C. Vonthron-Sénécheau, M. Bourjot, and M.-A. Delsuc, “Automatic differential analysis of NMR experiments

in complex samples," 2017.

[58] N. R. Lomb, "Least-squares frequency analysis of unequally spaced data," *Astrophys. Space Sci.*, vol. 39, no. 2, pp. 447–462, Feb. 1976.

[59] J. S.-T. A. Journal and undefined 1982, "Studies in astronomical time series analysis. II-Statistical aspects of spectral analysis of unevenly spaced data," *articles.adsabs.harvard.edu*.

[60] W. Press, G. R.-T. A. Journal, and undefined 1989, "Fast algorithm for spectral analysis of unevenly sampled data," *articles.adsabs.harvard.edu*.

[61] J. T. VanderPlas, "Understanding the Lomb-Scargle Periodogram," Mar. 2017.

[62] G. L. Bretthorst, "Generalizing the Lomb-Scargle periodogram—the nonsinusoidal case," in *AIP Conference Proceedings*, 2001, vol. 568, no. 1, pp. 246–251.

[63] G. L. Bretthorst, "Generalizing the Lomb-Scargle periodogram," in *AIP Conference Proceedings*, 2001, vol. 568, no. 1, pp. 241–245.

[64] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[65] A. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Trans. Acoust.*, vol. 29, no. 1, pp. 84–91, Feb. 1981.

[66] R. B. Randall, *Frequency analysis*. Brüel & Kjaer, 1987.

[67] M. Cerna and A. F. Harvey, "Application Note 041 The Fundamentals of FFT-Based Signal Analysis and Measurement," 2000.

[68] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemom. Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, Aug. 1987.

[69] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec.

2014.

[70] C. C. Felício, L. P. Brás, J. A. Lopes, L. Cabrita, and J. C. Menezes, “Comparison of PLS algorithms in gasoline and gas oil parameter monitoring with MIR and NIR,” *Chemom. Intell. Lab. Syst.*, vol. 78, no. 1–2, pp. 74–80, Jul. 2005.

[71] A. M. Mouazen, J. De Baerdemaeker, and H. Ramon, “Towards development of on-line soil moisture content sensor using a fibre-type NIR spectrophotometer,” *Soil Tillage Res.*, vol. 80, no. 1–2, pp. 171–183, Jan. 2005.

[72] Q. Chen, J. Zhao, M. Liu, J. Cai, and J. Liu, “Determination of total polyphenols content in green tea using FT-NIR spectroscopy and different PLS algorithms,” *J. Pharm. Biomed. Anal.*, vol. 46, no. 3, pp. 568–573, Feb. 2008.

[73] K. A. Stone, D. Shah, M. H. Kim, N. R. M. Roberts, Q. P. He, and J. Wang, “A novel soft sensor approach for estimating individual biomass in mixed cultures,” *Biotechnol. Prog.*, vol. 33, no. 2, pp. 347–354, Mar. 2017.

[74] P. Geladi and B. R. Kowalski, “Partial least-squares regression: a tutorial,” *Anal. Chim. Acta*, vol. 185, pp. 1–17, Jan. 1986.

[75] T.-H. Pan, B.-Q. Sheng, D. S.-H. Wong, and S.-S. Jang, “A virtual metrology model based on recursive canonical variate analysis with applications to sputtering process,” *J. Process Control*, vol. 21, no. 6, pp. 830–839, Jul. 2011.

[76] D. Zhou, G. Li, and S. J. Qin, “Total projection to latent structures for process monitoring,” *AIChE J.*, vol. 56, no. 1, p. NA-NA, Jan. 2009.

[77] F. Souza, R. Araújo, J. M.-C. and I. Laboratory, and undefined 2016, “Review of soft sensor methods for regression applications,” *Elsevier*.

[78] K. Bennett, M. E.-N. S. S. sub series III, and undefined 2003, “An optimization

perspective on kernel partial least squares regression," *researchgate.net*.

[79] R. Rosipal, L. T.-J. of machine learning research, and undefined 2001, "Kernel partial least squares regression in reproducing kernel hilbert space," *jmlr.org*.

[80] V. Vapnik, *The nature of statistical learning theory*. 2013.

[81] M. Momma and K. P. Bennett, "Sparse Kernel Partial Least Squares Regression," 2003, pp. 216–230.

[82] Z. Wang, Q. He, J. W.-J. of P. Control, and undefined 2015, "Comparison of variable selection methods for PLS-based soft sensor modeling," *Elsevier*.

[83] R. M. Balabin and S. V. Smirnov, "Variable selection in near-infrared spectroscopy: Benchmarking of feature selection methods on biodiesel data," *Anal. Chim. Acta*, vol. 692, no. 1–2, pp. 63–72, Apr. 2011.

[84] Z. Xiaobo, Z. Jiewen, M. J. W. Povey, M. Holmes, and M. Hanpin, "Variables selection methods in near-infrared spectroscopy," *Anal. Chim. Acta*, vol. 667, no. 1–2, pp. 14–32, May 2010.

[85] F. Lindgren, P. Geladi, S. Rännar, and S. Wold, "Interactive variable selection (IVS) for PLS. Part 1: Theory and algorithms," *J. Chemom.*, vol. 8, no. 5, pp. 349–363, Sep. 1994.

[86] I. Chong, C. J.-C. and intelligent laboratory systems, and undefined 2005, "Performance of some variable selection methods when multicollinearity is present," *Elsevier*.

[87] E. Zamprogna, M. Barolo, D. S.-J. of process control, and undefined 2005, "Optimal selection of soft sensor inputs for batch distillation columns using principal component analysis," *Elsevier*.

[88] J. L.-J. of P. Control and undefined 2014, “Developing a soft sensor based on sparse partial least squares with variable selection,” *Elsevier*.

[89] L. Cappellin, E. Aprea, P. Granitto, R. Wehrens, C. Soukoulis, R. Viola, T. D. Märk, F. Gasperi, and F. Biasioli, “Linking GC-MS and PTR-TOF-MS fingerprints of food samples,” *Chemom. Intell. Lab. Syst.*, vol. 118, pp. 301–307, Aug. 2012.

[90] C. Pan, J. Bai, G. Yang, D. Wong, S. J.-J. of P. Control, and undefined 2012, “An inferential modeling method using enumerative PLS based nonnegative garrote regression,” *Elsevier*.

[91] Q. P. He and J. Wang, “Statistics pattern analysis: A new process monitoring framework and its application to semiconductor batch processes,” *AIChE J.*, vol. 57, no. 1, pp. 107–121, Jan. 2011.

[92] J. Wang and Q. P. He, “Multivariate Statistical Process Monitoring Based on Statistics Pattern Analysis,” *Ind. Eng. Chem. Res.*, vol. 49, no. 17, pp. 7858–7869, Sep. 2010.

[93] R. Tibshirani, “Regression Shrinkage and Selection Via the Lasso,” *J. R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[94] L. Norgaard, A. Saudland, … J. W.-A., and undefined 2000, “Interval partial least-squares regression (iPLS): a comparative chemometric study with an example from near-infrared spectroscopy,” *osapublishing.org*.

[95] D. da Silva, H. W.-V. Spectroscopy, and undefined 2017, “Using PLS, iPLS and siPLS linear regressions to determine the composition of LDPE/HDPE blends: A comparison between confocal Raman and ATR-FTIR,” *Elsevier*.

[96] J. W. M.H. Kim, Q.P. He, “Quantifying the effects of oxygen utilization rate on ethanol 5 production by *S. stipitis* under controlled chemostat,” *AICHE Annu. Conf.*, 2015.

[97] L. Nørgaard, “iToolbox Manual,” 2013. [Online]. Available: http://www.models.kvl.dk/sites/default/files/iToolbox_Manual.pdf. [Accessed: 01-Apr-2005].

[98] G. Baffi, E. Martin, A. M.-C. & C. Engineering, and undefined 1999, “Non-linear projection to latent structures revisited: the quadratic PLS algorithm,” *Elsevier*.

[99] A. Berglund and S. Wold, “INLR, implicit non-linear latent variable regression,” *J. Chemom.*, vol. 11, no. 2, pp. 141–156, Mar. 1997.

[100] … R. R. advanced machine learning perspectives: complex and undefined 2011, “Nonlinear partial least squares an overview,” *igi-global.com*.

[101] S. Wold, N. Kettaneh-Wold, B. S.-C. and intelligent, and undefined 1989, “Nonlinear PLS modeling,” *Elsevier*.

[102] B. Schölkopf, A. Smola, K. M.-N. computation, and undefined 1998, “Nonlinear component analysis as a kernel eigenvalue problem,” *MIT Press*.

[103] J. Lee, C. Yoo, S. Choi, … P. V.-C. engineering, and undefined 2004, “Nonlinear process monitoring using kernel principal component analysis,” *Elsevier*.

[104] H. Susi and D. Michael Byler, “Protein structure by Fourier transform infrared spectroscopy: Second derivative spectra,” *Biochem. Biophys. Res. Commun.*, vol. 115, no. 1, pp. 391–397, Aug. 1983.

[105] Y. De Micalizzi, N. Pappano, N. D.- Talanta, and undefined 1998, “First and second order derivative spectrophotometric determination of benzyl alcohol and

diclofenac in pharmaceutical forms," *Elsevier*.

[106] N. Zhao, Z. Wu, Q. Zhang, X. Shi, Q. Ma, and Y. Qiao, "Optimization of Parameter Selection for Partial Least Squares Model Development," *Sci. Rep.*, vol. 5, no. 1, p. 11647, Dec. 2015.

[107] Q.-S. Xu and Y.-Z. Liang, "Monte Carlo cross validation," *Chemom. Intell. Lab. Syst.*, vol. 56, no. 1, pp. 1–11, Apr. 2001.

[108] "NIR of Corn Samples." [Online]. Available: http://www.eigenvector.com/data/Corn/index.html. [Accessed: 19-Mar-2018].

[109] "Gasoline dataset." [Online]. Available: https://www.mathworks.com/help/stats/sample-data-sets.html.

[110] "Pharmaceutical dataset." [Online]. Available: http://www.idrc-chambersburg.org/shootout_2002.html.

[111] "NIR Spectra of Pharmaceutical Tablets." [Online]. Available: http://www.eigenvector.com/data/tablets/index.html. [Accessed: 19-Mar-2018].

[112] D. W. Hopkins, "Shoot-out 2002: Transfer of Calibration for Content of Active in a Pharmaceutical Tablet," *NIR news*, vol. 14, no. 5, pp. 10–13, Oct. 2003.

[113] Q.-S. Xu, Y.-Z. Liang, and Y.-P. Du, "Monte Carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration," *J. Chemom.*, vol. 18, no. 2, pp. 112–120, Feb. 2004.

[114] W. Sabra, D. Dietz, D. Tjahjasari, and A.-P. Zeng, "Biosystems analysis and engineering of microbial consortia for industrial biotechnology," *Eng. Life Sci.*, vol. 10, no. 5, pp. 407–421, Oct. 2010.

[115] R. Kleerebezem, M. van L.-C. opinion in biotechnology, and undefined 2007,

"Mixed culture biotechnology for bioenergy production," *Elsevier*.

[116] J. Bader, E. Mast-Gerlach, M. K. Popović, R. Bajpai, and U. Stahl, "Relevance of microbial coculture fermentations in biotechnology," *J. Appl. Microbiol.*, vol. 109, no. 2, pp. 371–387, Aug. 2010.

[117] K. Brenner, L. You, F. A.-T. in biotechnology, and undefined 2008, "Engineering microbial consortia: a new frontier in synthetic biology," *Elsevier*.

[118] J. Shong, M. Diaz, C. C.-C. O. in Biotechnology, and undefined 2012, "Towards synthetic microbial consortia for bioprocessing," *Elsevier*.

[119] T. J. Hanly, M. Urello, and M. A. Henson, "Dynamic flux balance modeling of *S. cerevisiae* and *E. coli* co-cultures for efficient consumption of glucose/xylose mixtures," *Appl. Microbiol. Biotechnol.*, vol. 93, no. 6, pp. 2529–2541, Mar. 2012.

[120] M. B. Biggs, G. L. Medlock, G. L. Kolling, and J. A. Papin, "Metabolic network modeling of microbial communities," *Wiley Interdiscip. Rev. Syst. Biol. Med.*, vol. 7, no. 5, pp. 317–334, Sep. 2015.

[121] R. Dutta, *Fundamentals of biochemical engineering*. Ane Books India, 2008.

[122] T. H. Kim and Y. Y. Lee, "Pretreatment of Corn Stover by Soaking in Aqueous Ammonia at Moderate Temperatures," in *Applied Biochemistry and Biotecnology*, Totowa, NJ: Humana Press, 2007, pp. 81–92.

[123] A. Puri, S. Owen, F. Chu, … T. C.-A. E., and undefined 2015, "Genetic tools for the industrially promising methanotroph Methylomicrobium buryatense," *Am Soc Microbiol*.

[124] M. Liang, M. Kim, Q. He, J. W.-J. of bioscience and, and undefined 2013, "Impact of pseudo-continuous fermentation on the ethanol tolerance of

Scheffersomyces stipitis," *Elsevier*.

[125] M. Haenlein and A. M. Kaplan, "A Beginner's Guide to Partial Least Squares Analysis," *Underst. Stat.*, vol. 3, no. 4, pp. 283–297, Nov. 2004.

[126] H. Bothe, K. M. Jensen, M. A., L. J., J. C., H. Bothe, and J. L., "Heterotrophic bacteria growing in association with Methylococcus capsulatus (Bath) in a single cell protein production process," *Appl. Microbiol. Biotechnol.*, vol. 59, no. 1, pp. 33–39, Jun. 2002.

[127] N. Okuda, K. Ninomiya, Y. Katakura, S. S.-J. of bioscience and, and undefined 2008, "Strategies for reducing supplemental medium cost in bioethanol production from waste house wood hydrolysate by ethanologenic Escherichia coli: inoculum," *Elsevier*.

[128] M. Qian, S. Tian, X. Li, J. Zhang, Y. Pan, and X. Yang, "Ethanol Production From Dilute-Acid Softwood Hydrolysate by Co-Culture," *Appl. Biochem. Biotechnol.*, vol. 134, no. 3, pp. 273–284, 2006.

[129] B. H. Davison and G. Stephanopoulos, "Coexistence of*S. cerevisiae* and*E. coli* in chemostat under substrate competition and product inhibition," *Biotechnol. Bioeng.*, vol. 28, no. 11, pp. 1742–1752, Nov. 1986.

[130] R. S. Senger and H. Nazem-Bokaee, "Resolving Cell Composition Through Simple Measurements, Genome-Scale Modeling, and a Genetic Algorithm," 2013, pp. 85–101.

Appendices

Appendix A.1:

Table 0.1 Experimental conditions for which vibration data was collected

Sr. No.	Conditions	Approx. RPM	Approx. flow (gpm)
1	3	1500	5, 7, 9
2	3	1600	5, 7, 9
3	4	1700	5, 7, 9, 11
4	4	1750	5, 7, 9, 11
5	4	1800	5, 7, 9, 11
6	4	1850	5, 7, 9, 11
7	4	1900	6, 8, 10,12
8	4	1950	6, 8, 10,12
9	5	2000	5, 7, 9, 11, 13
10	5	2050	5, 7, 9, 11, 13
11	5	2100	6, 8, 10, 12, 14
12	5	2150	6, 8, 10, 12, 14
13	5	2200	6, 8, 10, 12, 14
14	5	2250	6, 8, 10, 12, 14
15	5	2300	7, 9, 11, 13, 15
16	5	2350	7, 9, 11, 13, 15
17	5	2400	7, 9, 11, 13, 15
18	5	2450	7, 9, 11, 13, 15
19	5	2500	8, 10, 12, 14, max ~15.9
Sum	85		

RPM value drifts around a value RPM label in Table 0.1 indicates values of RPM near that label. Similarly for flowrate, label indicate values collected near that value. For model training and testing actual measured values were used.

Appendix A.2:

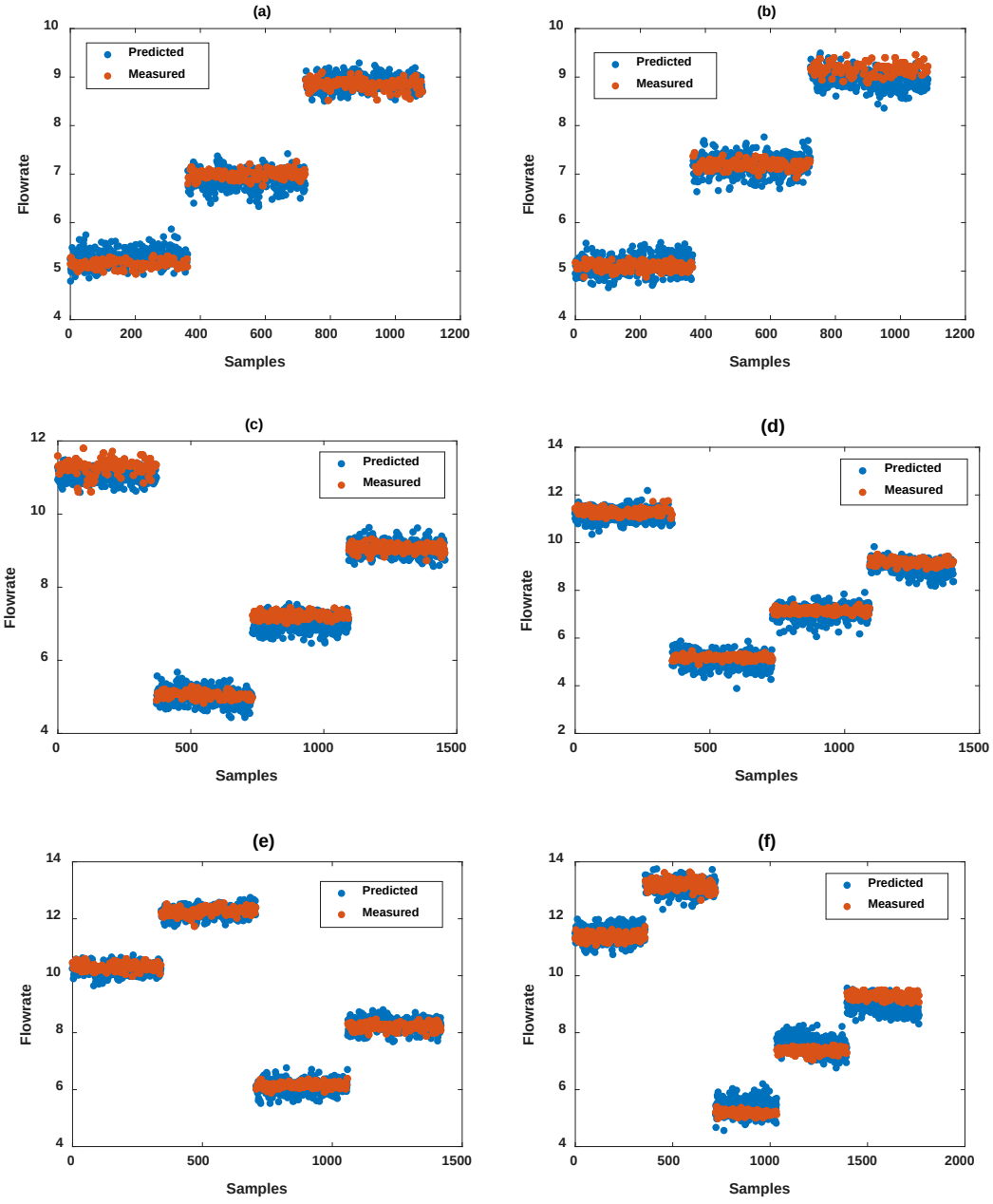
Table 0.2 Samples distribution data for flowrate hierarchical model

RPM model	Variables Selected	Calibration set size	Validation set size	Test set size
1500	1784	2167	2167	1085
1600	4701	2166	2167	1086
1700	1799	2909	2910	1457
1800	1550	2894	2895	1405
1900	2527	2836	2836	1420
2000	2163	3529	3529	1766
2100	2035	3435	3435	1720
2200	2264	3606	3607	1728
2300	2247	3600	3601	1647
2400	1126	3599	3600	1579
2500	1924	3594	3595	1595

Table 0.3 Samples distribution data for flowrate Improved hierarchical model

RPM model	Variables Selected	Calibration set size	Validation set size	Test set size
1500	1784	2161	2161	1082
1600	4701	2160	2161	1083
1700	1799	2901	2901	1453
1800	1550	2886	2887	1347
1900	2527	2264	2264	1094
2000	2163	3519	3519	1761
2100	2035	3425	3425	1715
2200	2264	3597	3598	1731
2300	2247	3600	3601	1697
2400	1126	3599	3600	1579
2500	1924	3594	3595	1595

Appendix A.3


(a)
Predicted
Measured
Flowrate
Samples
(b)
Predicted
Measured
Flowrate
Samples
(c)
Predicted
Measured
Flowrate
Samples
(d)
Predicted
Measured
Flowrate
Samples
(e)
Predicted
Measured
Flowrate
Samples
(f)
Predicted
Measured
Flowrate
Samples

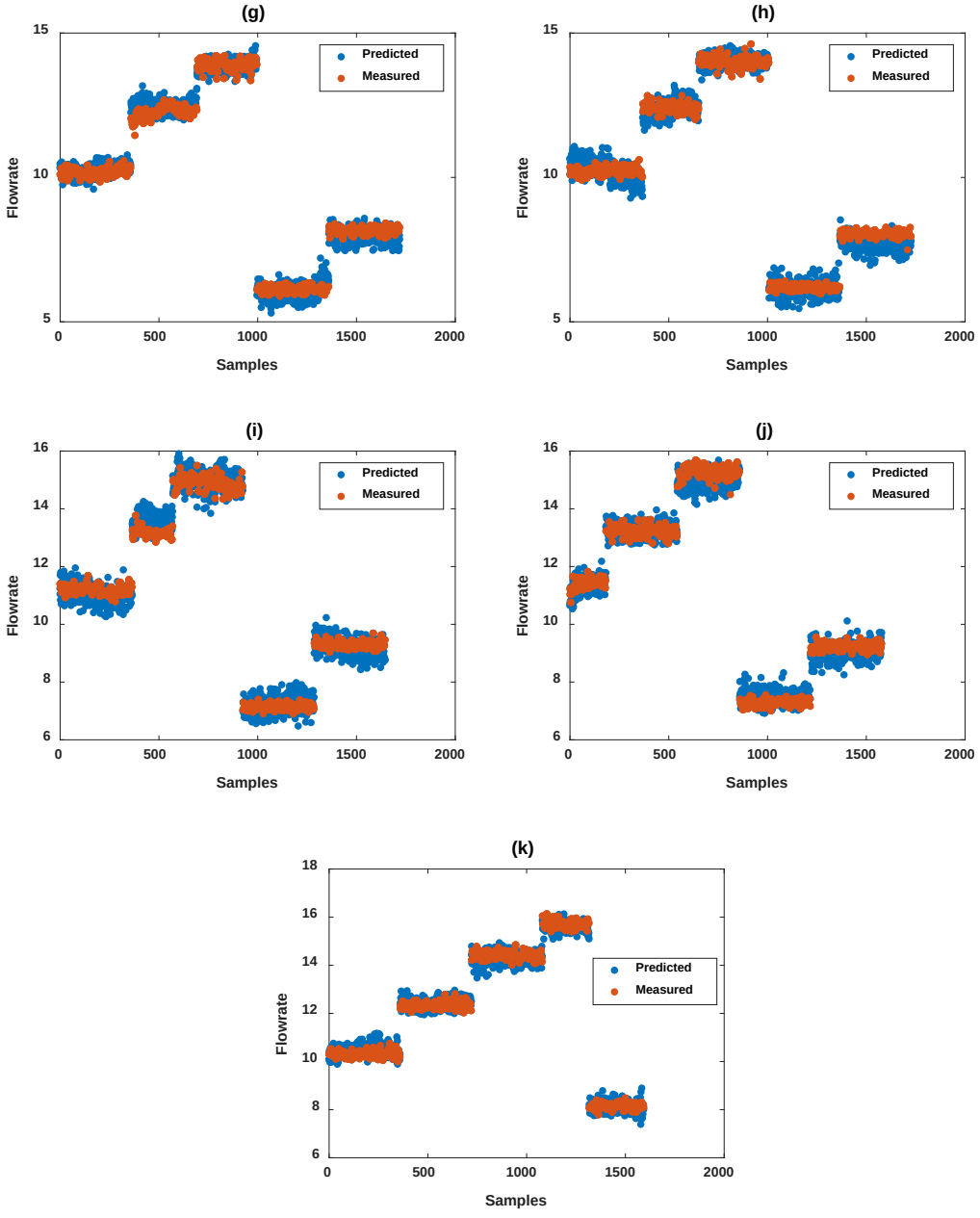
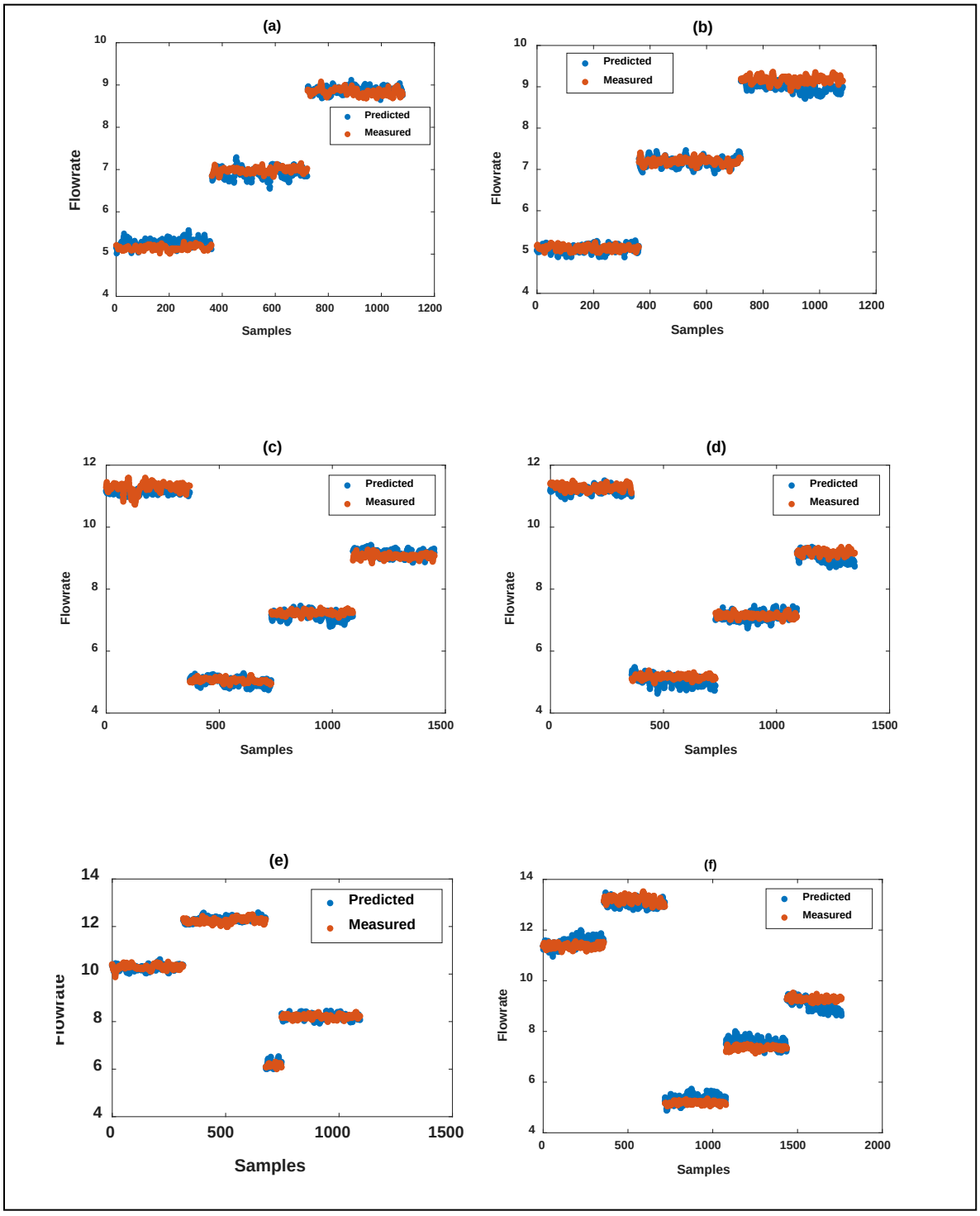


Figure 0.1 Hierarchical Model's Prediction performance for (a) 1500, (b) 1600, (c) 1700, (d) 1800, (e) 1900, (f) 2000, (g) 2100, (h) 2200, (i) 2300, (j) 2400, (k) 2500 conditions.


(a)
Predicted
Measured
Flowrate
Samples
(b)
Predicted
Measured
Flowrate
Samples
(c)
Predicted
Measured
Flowrate
Samples
(d)
Predicted
Measured
Flowrate
Samples
(e)
Predicted
Measured
Flowrate
Samples
(f)
Predicted
Measured
Flowrate
Samples

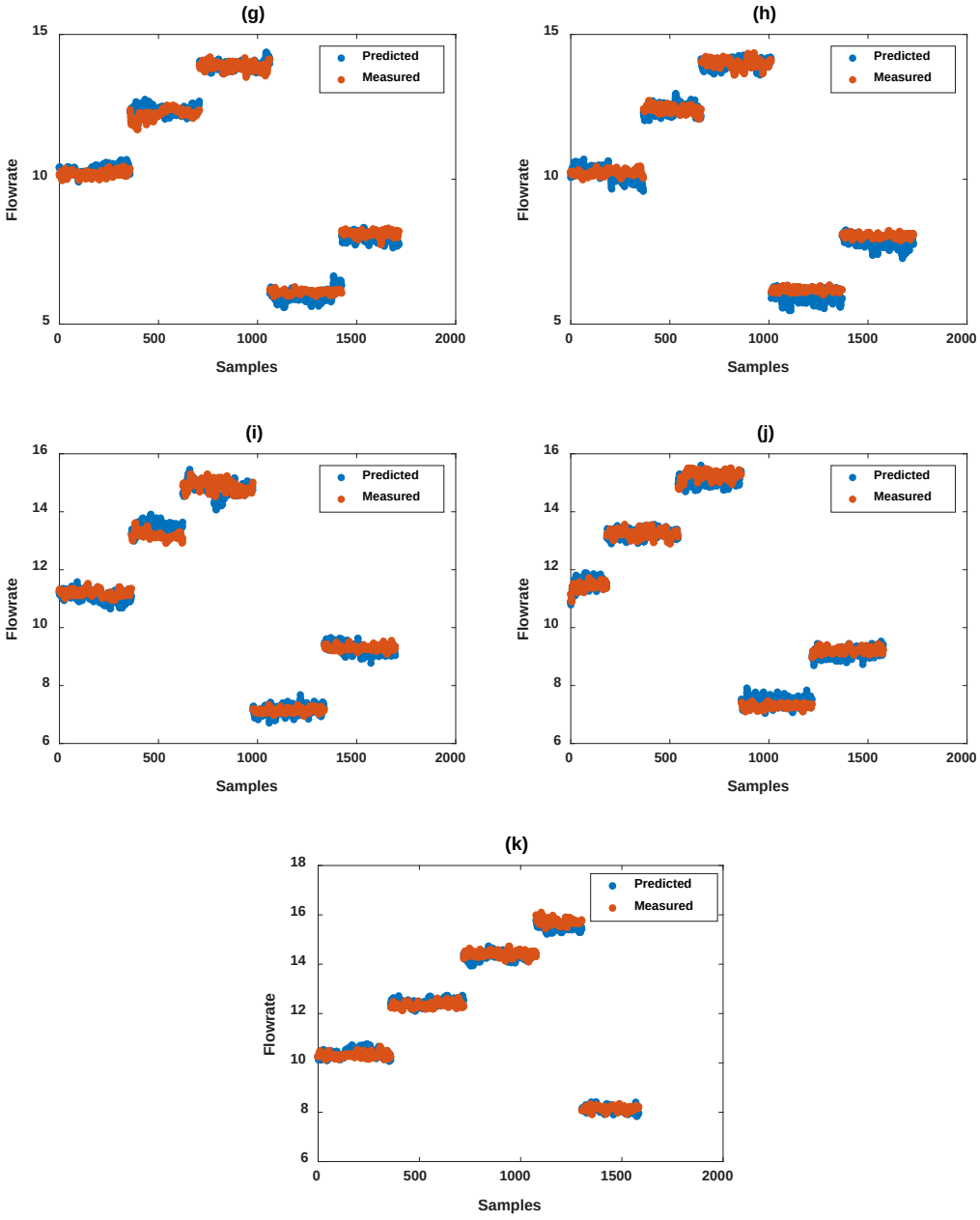


Figure 0.2 Improved Hierarchical Model's Prediction performance for (a) 1500, (b) 1600, (c) 1700, (d) 1800, (e) 1900, (f) 2000, (g) 2100, (h) 2200, (i) 2300, (j) 2400, (k) 2500 conditions.

Appendix B:

A Novel Soft Sensor Approach for Estimating Individual Biomass in Mixed Cultures

Abstract:

Due to many advantages associated with mixed cultures, their application in biotechnology has expanded rapidly in recent years. At the same time, many challenges remain for effective mixed culture applications. One obstacle is how to efficiently and accurately monitor the individual cell populations. Current approaches on individual cell mass quantification are suitable for off-line, infrequent characterization. In this study, we propose a fast and accurate "soft sensor" approach for estimating individual cell concentrations in mixed cultures. The proposed approach utilizes optical density scanning spectrum of a mixed culture sample measured by a spectrophotometer over a range of wavelengths. A multivariate linear regression method, partial least squares or PLS, is applied to correlate individual cell concentrations to the spectrum. Three experimental case studies are used to examine the performance of the proposed soft sensor approach.

Redrafted from:

K.A. Stone, **D. Shah**, M.H. Kim, N.R.M. Roberts, Q.P. He, J. Wang, A Novel Soft Sensor Approach for Estimating Individual Biomass in Mixed Cultures, Biotechnology Progress, 33, 2017, 347-354.

Introduction:

Mixed cultures are biological systems with more than one type of organism sharing or competing for available nutrients. Traditional applications of mixed culture include food and beverage processes, waste water treatment, soil remediation, biogas production, *etc*.[114]–[116]. The application of mixed cultures in biotechnology has expanded rapidly in the last few years with newer applications for producing solvents, acids, plastics, hydrogen, antibiotics, and other valued commodities. There are many benefits associated with mixed cultures. For example, when organisms in mixed cultures work symbiotically, the cells' metabolic pathways complement each other and can result in efficient utilization of substrates and increased product yield. Other advantages include the use of mixed or cheaper substrates, reduction in sterilization, and increased robustness to environmental changes[117].

At the same time, many challenges remain for effective mixed culture applications, such as cultivation, microbial interaction, and culture characterization. To address these challenges, many recent publications have focused on microbial communication[118], and metabolic modeling and analysis[119], [120]. However, one prerequisite to the above mentioned research has not been fully addressed, *i.e.*, how to efficiently and accurately monitor the individual cell populations in a mixed culture.

The current approaches on individual cell mass quantification in mixed cultures can be classified into three groups: molecular biological, biochemical, and microbiological methods. Molecular biological methods are based on the analysis and differentiation of microbial DNA to separately identify and quantify the individual strains in a mixed culture. Biochemical methods focus on certain biomolecules (*e.g.*, lipids) as an indicator of cell characterization and quantification of mixed cultures. Microbiological methods

rely on traditional tools (such as cell counting, selective growth, cell sorting, and microscopic examination) for the same analysis.

Although some of the existing methods can provide more information besides individual cell concentration, there are some drawbacks. For example, some of them are expensive, such as flow cytometry, community genome sequencing, DNA microarray and metaproteomics; some involve challenging techniques such as extraction and isolation of RNA, proteins and metabolites; others involve manual and time-consuming procedures such as direct/indirect cell counting and morphological counting. Therefore, they are suitable for off-line, infrequent characterization of the mixed culture system.

In order to develop a simple, fast, and accurate measurement of individual cell concentrations in mixed cultures, we extended the use of a spectrophotometer to relate absorbance readings with cell concentrations. Specifically, we propose a 'soft sensor' approach by utilizing the samples' absorption spectrum over a range of wavelengths, instead of at a single wavelength. It should be noted that the measurement of cell mass in pure cultures using spectrophotometers at a single wavelength is well established [11,12]. By measuring the absorbance of a sample (often gauged by optical density, OD) at a specified wavelength, such as 600 or 650 nm, and utilizing a known calibration curve that correlates cell mass concentrations with OD, one can easily obtain cell mass concentrations of a pure culture[121]. However, this method cannot be directly applied to measure individual cell mass concentrations in mixed cultures because the absorbance readings at a single wavelength does not correspond to a unique concentrations profile for mixed cultures. For example, a mixed culture that contains two different organisms could have an infinite number of concentration pairs that result in the same total OD reading.

Therefore, one cannot determine the individual cell concentration of the mixed culture based on the OD reading at a single wavelength. However, the basic idea behind the measurement of cell mass of pure cultures can be extended to measure individual cell mass of mixed cultures.

Materials and Methods

Microorganisms and Growth Medium

Both *Escherichia coli* KO11 and *Saccharomyces cerevisiae* D5A were provided by Dr. Y.Y. Lee from Auburn University. *Methylomicrobium buryatense* 5GB1 was provided by Dr. Mary Lidstrom from the University of Washington. *Scheffersomyces stipitis* CBS 5773 was purchased from ATCC. *Escherichia coli* MG1655 utilized for the co-culture dynamic Case 4 was provided by Dr. Sang-Jin Suh from Auburn University.

E. coli KO11 was grown in LB medium as described by Kim and Lee [122] and *M. buryatense* in modified nitrate mineral slats (NMS2) medium as described by Puri *et. al.*[123]. Both *S. cerevisiae* and *S. stipitis* were grown in the same pre-cultured medium as described by Liang *et. al.*[124] with glucose used for *S. cerevisiae* and xylose for *S. stipitis*. Co-cultures of *E. coli* and *S. cerevisiae* were grown in the same pre-culture YPD medium as the pure yeast cultures with glucose serving as the main carbon source.

Experimental Design

Two groups of experiments were designed to examine the robustness and performance of the proposed soft sensor approach. Cases 1 and 2 are static experiments where little or no active cell growth occurred, while Case 3 is a dynamic experiment where both strains in the mixed culture grew together till stationary phase. In Cases 1 and 2, the individual strains (*M. buryatense* and *S. stipitis* in Case 1) and (*E. coli* KO11 and *S.*

cerevisiae in Case 2) were grown independently until late exponential phase or early stationary phase to minimize biomass changes during experiments. The cells were centrifuged into dense pellets utilizing an Eppendorf centrifuge 5702 R with a speed of

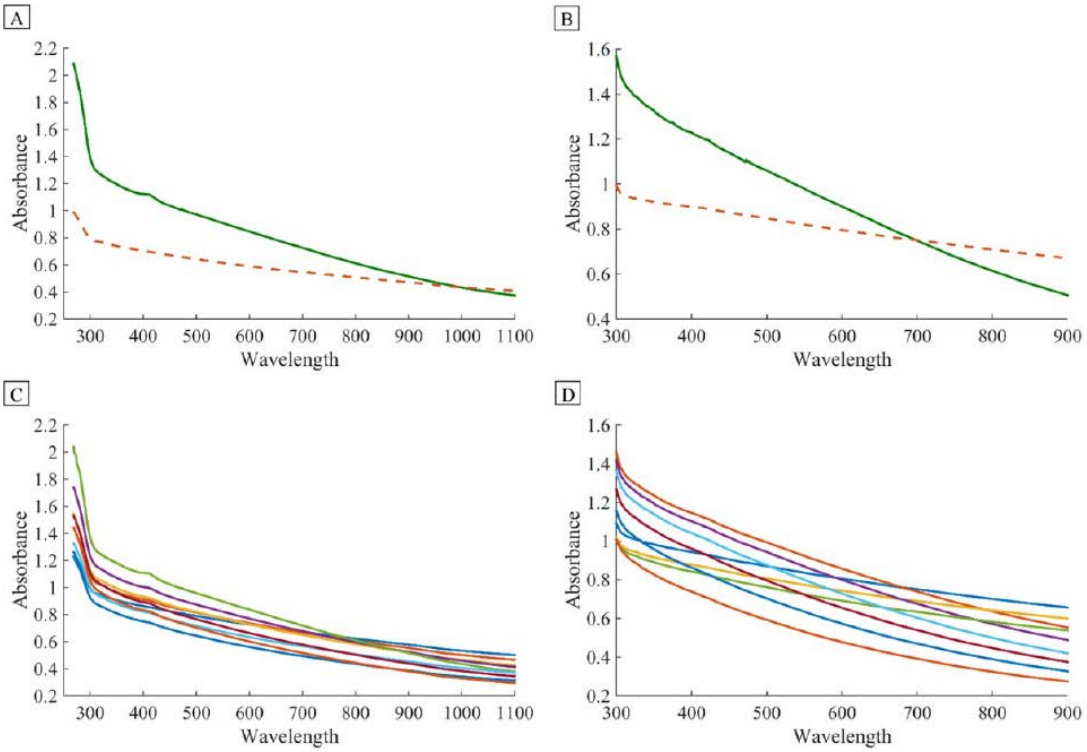


Figure B.0.3 (a) OD spectra of pure M. buryatense (solid line) and S. stipitis (dashed line) over the wavelength of 269–1100 nm. (b) OD spectra of pure E. coli (solid line) and S. cerevisiae (dashed line) over the wavelength of 300–900 nm. (c) OD spectra of nine different mixed cultures of M. buryatense and S. stipitis over the wavelength of 269–1100 nm.

From top to bottom at 1100 nm are samples 1, 5, 8, 12, 19, 22, 24, 27, and 29 as listed in Table B.0.4. (d) OD spectra of nine different mixed cultures of *E. coli* and *S. cerevisiae* over the wavelength of 300–900 nm. From top to bottom at 900 nm are samples 2, 1, 10, 7, 16, 26, 33, 41, and 47 as listed in Table B.3

4400 RPM, then cells were washed, subsequently resuspended with sterile DI water, and collectively placed in a bottle that served as a stock culture with a relatively constant OD600. The OD readings were taken with the Beckman Coulter DUVR 730 spectrophotometer. The OD600 of the stock solution was related to dry cell weight per liter (g DCW/L) through calibration curves. The experimental design for Cases 1 and 2 are shown in the first five columns of Table B.0.4 and Table B.0.5. All mixture samples were 2 mL in total volume. They were prepared by adding specific amounts of stock solution of each strain (shown in the third and fourth columns of Table B.0.4 and Table B.0.5) into a test tube, then adding sterile DI water till the total volume is 2 mL. Triplicate samples were taken from each mixture for OD scanning. In Case 3, S. cerevisiae and *E. coli* KO11 were cocultured together. Because *E. coli* grows much faster than *S. cerevisiae*, the inoculum ratio of *S. cerevisiae* to *E. coli* was set to be 4:1 in terms of the OD600 reading. Throughout the coculture growth, samples were taken at roughly 1 h intervals until cells reach stationary phase. Each sample was centrifuged and resuspended in DI water, before taking OD scanning and cell counting. Again, triplicate samples were taken at each time point for individual cell concentration measurements. Cell counting was done by using a Bright LineTM Hemacytometer (Cambridge Instrument Inc) under an AMG EVOSVR FL Auto Imaging System (Life Technology Corporation). The microscopic images for cell counting were processed using ImageJ (Image processing and analysis in Java, https://imagej.nih.gov/ij/) as well as manual counting.

Table B.0.4 Experimental Design and Soft Sensor Performance for Case 1 with M. buryatense and S. stipitis

		Stock Solution Volume (mL) Concentration[1] (g DCW/L)			Concentration[1] (g DCW/L)		Predicted Concentration ± STD (g DCW/L)	
Sub-group	Sample No	*M. buryatense*	*S. stipitis*	Total[2]	*M. buryatense*	*S. stipitis*	*M. buryatense*	*S. stipitis*
1	1	0.1	1.9	2	0.016	1.319	0.025±0.001	1.330±0.029
1	2	0.2	1.8	2	0.032	1.250	0.034±0.001	1.286±0.013
1	3	0.3	1.7	2	0.048	1.180	0.050±0.001	1.131±0.008
1	4	0.4	1.6	2	0.064	1.111	0.068±0.001	1.073±0.011
1	5	0.5	1.5	2	0.080	1.041	0.081±0.001	1.040±0.007
1	6	0.6	1.4	2	0.096	0.972	0.095±0.001	0.947±0.007
1	7	0.7	1.3	2	0.111	0.903	0.110±0.001	0.933±0.007
1	8	0.8	1.2	2	0.127	0.833	0.124±0.001	0.782±0.004
1	9	0.9	1.1	2	0.143	0.764	0.140±0.001	0.755±0.006
1	10	1.0	1.0	2	0.159	0.694	0.157±0.001	0.710±0.005
1	11	1.1	0.9	2	0.175	0.625	0.170±0.001	0.644±0.006
1	12	1.2	0.8	2	0.191	0.555	0.187±0.001	0.548±0.006
1	13	1.3	0.7	2	0.207	0.486	0.206±0.001	0.463±0.004
1	14	1.4	0.6	2	0.223	0.417	0.218±0.001	0.438±0.006
1	15	1.5	0.5	2	0.239	0.347	0.238±0.001	0.358±0.007
1	16	1.6	0.4	2	0.255	0.278	0.255±0.001	0.273±0.005
1	17	1.7	0.3	2	0.271	0.208	0.273±0.001	0.209±0.006
1	18	1.8	0.2	2	0.287	0.139	0.288±0.002	0.127±0.006
1	19	1.9	0.1	2	0.303	0.069	0.305±0.002	0.067±0.009
2	20	0.2	1.4	1.6	0.032	0.972	0.033±0.002	0.987±0.013
2	21	0.4	1.2	1.6	0.064	0.833	0.067±0.001	0.882±0.004
2	22	0.6	1.0	1.6	0.096	0.694	0.093±0.001	0.712±0.005
2	23	0.8	0.8	1.6	0.127	0.555	0.129±0.001	0.561±0.003
2	24	1.0	0.6	1.6	0.159	0.417	0.162±0.001	0.439±0.006
2	25	1.2	0.4	1.6	0.191	0.278	Outlier	Outlier
2	26	1.4	0.2	1.6	0.223	0.139	Outlier	Outlier
3	27	0.2	1.2	1.4	0.032	0.833	0.025±0.002	0.822±0.011
3	28	0.4	1.0	1.4	0.064	0.694	0.057±0.001	0.675±0.009
3	29	0.6	0.8	1.4	0.096	0.555	0.092±0.001	0.526±0.004
3	30	0.8	0.6	1.4	0.127	0.417	0.126±0.001	0.409±0.005
3	31	1.0	0.4	1.4	0.159	0.278	0.163±0.002	0.278±0.009
3	32	1.2	0.2	1.4	0.191	0.139	0.203±0.001	0.146±0.024

[1]Calculated concentrations with gram dry cell weight (g DCW)/L were based on the stock solution absorbance (OD600) and the calibrated concentration curves of *E. coli* and *S. cerevisiae*.

[2]Total stock solution volume before dilution. If the total stock solution volume is less than 2 mL, sterile DI water will be added till the total volume is 2 mL for each sample before characterization.

Table B.0.5 Experimental Design and Soft Sensor Performance for Case 2 with E. coli and S. cerevisiae

		Stock Solution Volume (mL) Concentration[1] (g DCW/L)			Concentration[1] (g DCW/L)		Predicted Concentration ± STD (g DCW/L)	
Sub-group	Sample No	*E. coli*	*S. cerevisiae*	Total[2]	*E. coli*	*S. cerevisiae*	*E. coli*	*S. cerevisiae*
1	1	0.2	1.8	2	0.046	0.386	0.041±0.001	0.382±0.001
1	2	0.4	1.6	2	0.092	0.343	0.09±0.001	0.339±0.001
1	3	0.6	1.4	2	0.138	0.301	0.131±0.001	0.304±0.001
1	4	0.8	1.2	2	0.184	0.258	0.176±0.001	0.264±0.001
1	5	1	1	2	0.23	0.215	0.223±0.001	0.22±0.001
1	6	1.2	0.8	2	0.276	0.172	0.268±0.001	0.176±0.001
1	7	1.4	0.6	2	0.322	0.129	0.315±0.001	0.131±0.001

1	8	1.6	0.4	2	0.368	0.086	0.365±0.001	0.086±0.001
1	9	1.8	0.2	2	0.414	0.043	0.418±0.001	0.038±0.001
2	10	0.2	1.6	1.8	0.046	0.347	0.045±0.001	0.342±0.001
2	11	0.4	1.4	1.8	0.092	0.303	0.095±0	0.299±0.001
2	12	0.6	1.2	1.8	0.137	0.26	0.141±0	0.259±0
2	13	0.8	1	1.8	0.183	0.217	0.183±0.001	0.22±0.001
2	14	1	0.8	1.8	0.229	0.173	0.227±0	0.175±0
2	15	1.2	0.6	1.8	0.275	0.13	0.275±0	0.131±0
2	16	1.4	0.4	1.8	0.321	0.087	0.324±0.001	0.085±0
2	17	1.6	0.2	1.8	0.366	0.043	0.368±0.001	0.041±0.001
3	18	0.2	1.4	1.6	0.046	0.307	0.047±0.001	0.308±0.001
3	19	0.3	1.3	1.6	0.069	0.285	0.074±0.001	0.284±0.001
3	20	0.4	1.2	1.6	0.092	0.264	0.095±0	0.263±0.001
3	21	0.6	1	1.6	0.137	0.22	0.143±0.001	0.218±0
3	22	0.8	0.8	1.6	0.183	0.176	0.19±0	0.174±0
3	23	1	0.6	1.6	0.229	0.132	0.229±0	0.134±0
3	24	1.2	0.4	1.6	0.275	0.088	0.28±0	0.086±0
3	25	1.3	0.3	1.6	0.297	0.066	0.299±0.001	0.065±0
3	26	1.4	0.2	1.6	0.32	0.044	0.321±0.001	0.042±0
4	27	0.2	1.2	1.4	0.046	0.266	0.045±0.001	0.27±0.002
4	28	0.4	1	1.4	0.091	0.221	0.102±0.001	0.219±0.001
4	29	0.6	0.8	1.4	0.137	0.177	0.152±0.001	0.17±0
4	30	0.7	0.7	1.4	0.16	0.155	0.164±0	0.154±0
4	31	0.8	0.6	1.4	0.183	0.133	0.186±0	0.133±0
4	32	1	0.4	1.4	0.228	0.089	0.229±0.001	0.091±0
4	33	1.2	0.2	1.4	0.274	0.044	0.274±0.001	0.045±0
5	34	0.2	1	1.2	0.045	0.223	0.04±0.001	0.234±0.001
5	35	0.3	0.9	1.2	0.068	0.201	0.07±0.001	0.2±0.001
5	36	0.4	0.8	1.2	0.091	0.179	0.093±0.001	0.177±0.001
5	37	0.5	0.7	1.2	0.114	0.156	0.119±0.001	0.153±0
5	38	0.6	0.6	1.2	0.136	0.134	0.14±0	0.133±0
5	39	0.7	0.5	1.2	0.159	0.112	0.159±0	0.112±0
5	40	0.8	0.4	1.2	0.182	0.089	0.18±0	0.091±0
5	41	1	0.2	1.2	0.227	0.045	0.227±0.001	0.046±0.001
6	42	0.3	0.7	1	0.068	0.158	0.058±0.001	0.163±0.002
6	43	0.4	0.6	1	0.091	0.135	0.084±0.001	0.137±0.001
6	44	0.5	0.5	1	0.113	0.113	0.111±0.001	0.111±0
6	45	0.6	0.4	1	0.136	0.09	0.134±0.001	0.089±0
6	46	0.7	0.3	1	0.159	0.068	0.153±0	0.067±0
6	47	0.8	0.2	1	0.182	0.045	0.177±0	0.045±0

[1]Calculated concentrations with gram dry cell weight (g DCW)/L were based on the stock solution absorbance (OD600) and the calibrated concentration curves of *E. coli* and *S. cerevisiae*.

[2]Total stock solution volume before dilution. If the total stock solution volume is less than 2 mL, sterile DI water will be added till the total volume is 2 mL for each sample before characterization.

Results and Discussion

Soft Sensor Development

As shown in *Figure B.0.3* a and *Figure B.0.3* b, different organisms have different absorption spectra over a range of wavelengths because of their different cellular compositions. Upon mixing these organisms at different concentrations, unique

absorbance spectra over a range of wavelengths would be obtained as shown in *Figure B.0.3* c and *Figure B.0.3* d, which would allow for the complete determination of the individual cell concentrations in the mixed culture.

The proposed method is based on multivariate linear regression, which correlates individual biomass concentrations in a mixed culture to the overall absorption spectrum of the mixed culture. Such an approach is called ‘soft sensing’ and is commonly used in the process industry where easily obtained secondary measurements (e.g., absorption spectra in this study) are used to estimate the primary measure that is difficult to obtain in real-time (e.g., individual cell concentration in this study)[7]. In this study, partial least squares (PLS) is used to build the soft sensor model that correlates the mixture sample’s OD scanning spectrum to individual cell concentration.

The soft sensor development consists of two steps: model building and testing. In the model building step (step I), different samples of mixed culture with known individual cell concentrations (established through sample preparation) are measured using a spectrophotometer to obtain their absorption spectra. Then a multivariate model (partial least square, or PLS, in this work) is developed by using the absorbance at different wavelengths as independent variables (x), and the corresponding individual cell concentrations as dependent variables (y). In the testing step (step II), the absorption spectrum of a new mixed culture sample is fed to the model to estimate the individual biomass concentrations. The estimated concentrations are then compared with the actual concentrations (established through sample preparation) to evaluate the model performance.

Partial least squares (PLS) used in this work is a multivariate linear regression method that utilizes dimension reduction and can handle highly correlated inputs[42]. Generally speaking, proper variable selection would improve the soft sensor performance[74]. For more information regarding linear regression and principal component reduction utilized in PLS, readers are directed to comprehensive works elsewhere[125], [126]. The soft sensor development consists of the following two steps: model building and testing. In the model building step (step I), different samples of mixed culture with known individual cell concentrations (established through sample preparation) are measured using a spectrophotometer to obtain their absorption spectra. Then, a multivariate model (i.e., PLS model in this work) is developed by using the OD readings at different wavelengths as independent variables (X) and the corresponding individual cell concentrations as dependent variables (Y). In the testing step (step II), the OD scanning spectrum of a new mixed culture sample is fed to the model to estimate the individual biomass concentrations. The estimated concentrations are then compared with the actual concentrations (established through sample preparation) to evaluate the model performance. In this study, the soft sensor models were built using static data only, that is, data collected in Cases 1 and 2. Then, the soft sensor performance was tested with both static and dynamic case studies (Cases 1, 2, and 3).

Static Case Studies and Outlier Removal

Two case studies (Cases 1 and 2) were conducted to examine the performance of the proposed soft sensor approach, where both mixed cultures consist of a bacterium and a yeast strain. Case 1 uses *M. buryatense*, a methanotrophic bacterium, and *S. stipitis*, a yeast. Methanotrophs have been known to grow well in mixed consortiums and are able

to produce potential excreted carbon products (*i.e.*, lactate and polysaccharides) that are potentially useful to heterotrophic organisms. 22,23 The choice of *M. buryatense* and *S. stipitis* has no real application significance other than to evaluate the performance of the soft sensor approach. Case 2 uses *E. coli* KO11, a bacterium, and *S. cerevisiae*, a yeast. Cocultures of *E. coli* and *S. cerevisiae* have been studied since 1986 and have been found useful for lignocellulosic ethanol production, especially for efficient consumption of mixed lignocellulosic sugars[119], [127]–[129]. For both case studies, the experiments were designed in the way that all mixture samples can be divided into different subgroups. Within each subgroup, the cell concentration of each individual strain changes in the opposite direction linearly, that is, the linearly increasing concentration of strain A is paired with linearly decreasing concentrations of strain B, while the total volume of the stock solutions (before dilution to 2 mL eventually) is maintained at a fixed level. More details about sample preparation can be found in the materials and methods section. Case 1 includes 32 samples from three subgroups (*i.e.*, three total stock solution volume levels as shown in the fifth column of Table B.0.4), while Case 2 includes 47 samples from six subgroups (*i.e.*, six total stock solution volume levels as shown in the fifth column of Table B.0.5). Detailed experimental design for both cases is given in Table B.0.4 and Table B.0.5, along with the predicted concentrations using the proposed soft sensor approach.

Outlier Removal: Before utilizing the measured OD of the mixed cultures samples for the building and testing of the soft sensor, preliminary outlier analysis was conducted through principal component analysis (PCA). PCA projects the high-dimensional input variables X (*i.e.*, OD readings over hundreds of wavelengths) onto a low dimensional

principal component subspace defined by few underlying independent variables. Figure B.0.4 a,b shows the projection of the sample spectra in both case studies onto a 2-dimensionl principal component subspace, and it can be seen that the samples from different subgroups are clearly separated as samples of each subgroup cluster together roughly to form a line. For Case 1, Figure B.0.4 a shows that samples 25 and 26, which belong to subgroup 2 according to the experimental design, are obvious outliers, as they fall out of place. This suggests that the two samples may contain errors during sample preparation (e.g., not putting the specified amount of single cell broth into the mixed culture), and therefore, these two samples were excluded from further analysis. It is worth noting that the outliers have no impact on the OD readings of the other samples and therefore do not affect the subsequent analyses.

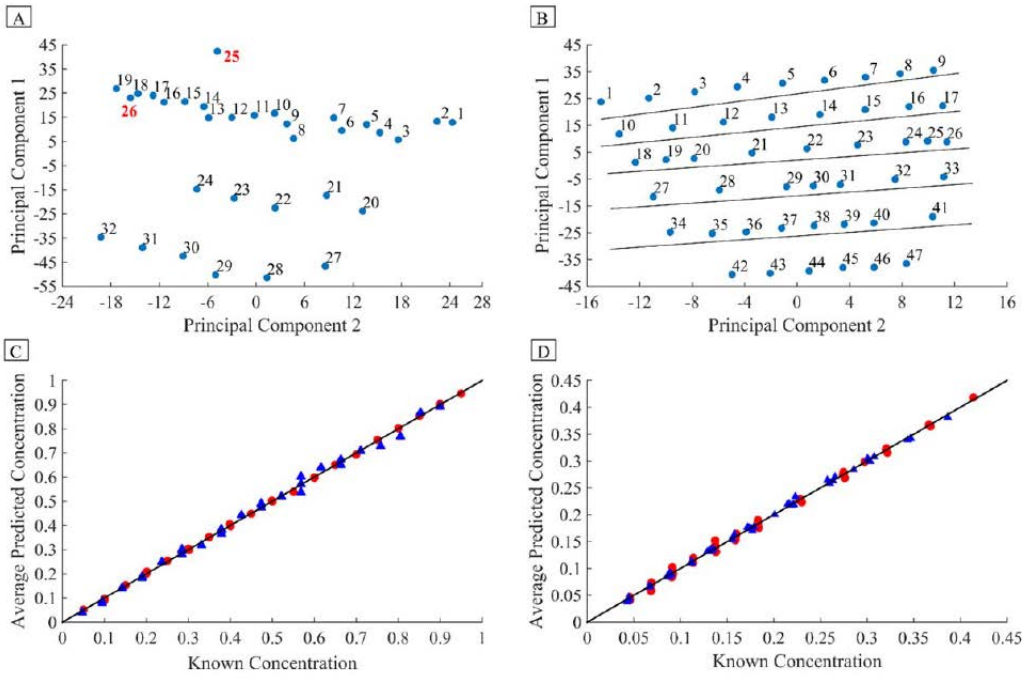


Figure B.0.4 (a) PC plot of all the samples for Case 1 containing *M. buryatense* and *S. stipitis*. Subgroup 1 consists of samples 1–19, subgroup 2 consists of 20–26, and

subgroup 3 consists of 27–32. Points 25 and 26 are outliers detected by PCA and consequently not utilized in the soft sensor development. (b) PC plot of the different samples for Case 2 containing *E. coli* and *S. cerevisiae*. The solid lines divide all samples into six subgroups, which is consistent with the experimental design. Subgroup 1 consists of samples 1–9, subgroup 2 consists of 10–17, subgroup 3 consists of 18–26, subgroup 4 consists of 27–33, subgroup 5 consists of 34–41, and subgroup 6 consists of 42–47. (c) Comparison of soft sensor predictions and known concentrations for Case 1. The average predicted concentrations are those from the 100 random MC runs. The diagonal line represents the case where predicted and known concentrations are the same. Due to the low cell concentration of *M. buryatense* stock solution, the predication and known values were scaled for this plot. The actual values are provided in Table B.0.4. The red filled dots represent *M. buryatense* and the blue filled triangles represent *S. stipitis*. (d) Comparison of soft sensor predictions and known concentrations for Case 2. The average predicted concentrations are those from the 100 random MC runs. The diagonal line represents the case where predicted and known concentrations are the same. The red filled dots represent E.coli and the blue filled triangles represent *S. cerevisiae*.

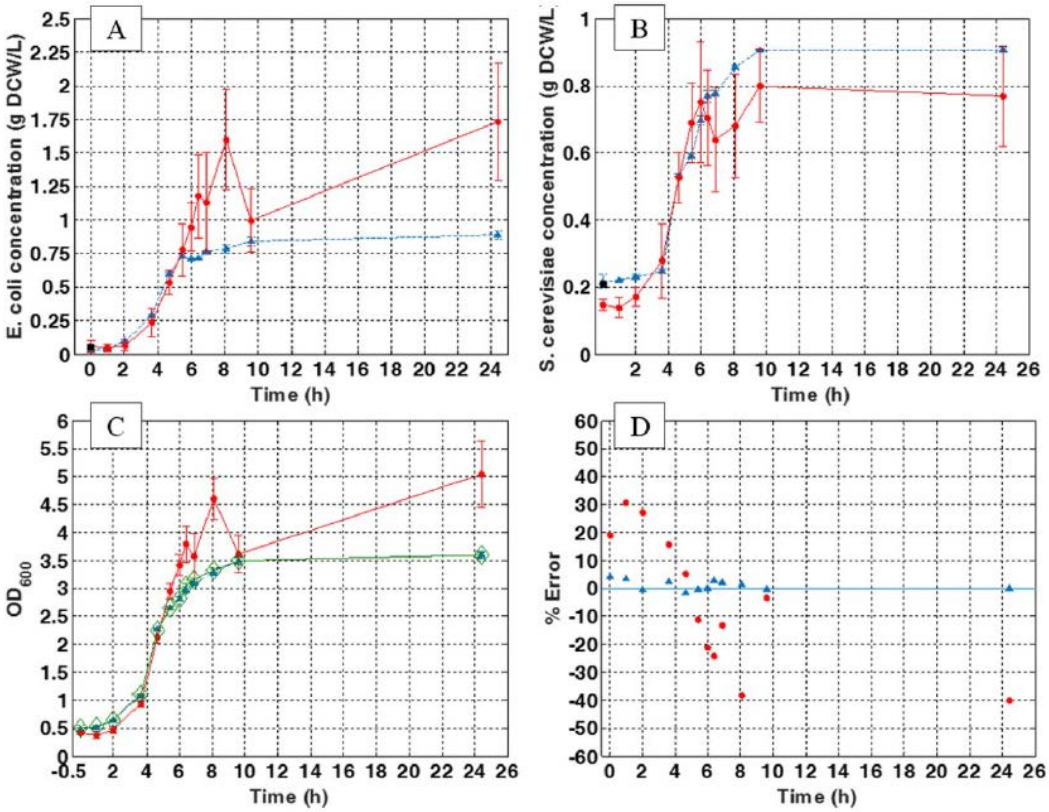


Figure B.0.5 (a and b) *E. coli* and *S. cerevisiae* cell concentrations in gram dry cell weight per liter (g DCW/L) over time, respectively. The red filled circles with solid lines represent the cell concentration estimated via cell counting and the blue filled triangles with dashed lines represent the cell concentration estimated via the soft sensor. The known individual cell concentration in the initial inoculum is marked as a black filled square at zero hour. (c) Total OD600 from the results of the soft sensor and cell counting compared to the measured total OD600. The green open diamond with solid lines represent measured OD600; the red filled circles with solid lines represent the total OD600 calculated by linear superposition of the reproduced individual strain OD600 based on the cell counting method. The blue filled triangles with dashed lines represent the total OD600 calculated by linear superposition of the reproduced individual strain

OD600 based on the soft sensor approach. (d) Percentage error of the total OD600 of the cell counting method (represented by the red filled circles) and the soft sensor method (represented by the blue filled triangles).

Evaluating Soft Sensor Performance

After outlier removal, for each case study all samples were randomly divided into two sets: a training set and a testing set, with training set containing at least 2 samples from each subgroup. For Case 1, the training set contains 20 samples, while the testing set contains 10 samples. For Case 2, the training set contains 35 samples and the testing set contains 12 samples.

In order to evaluate how robust the soft sensor performance is, 100 Monte Carlo (MC) simulations were performed. Within each simulation, a different training set was used, which results in a different testing set as well. For each training set (*i.e.*, the samples' absorbance spectra and corresponding cell concentrations for both strains) a PLS soft sensor model is built to correlate the absorbance spectra with each individual cell concentration (*i.e.*, two models were built for Case 1 and Case 2 respectively). The number of principal components were selected based on cross validation. For Case 1, 4 PCs were chosen for predicting *M. buryatense* concentration, while 3 PCs were chosen for predicting *S. stipitis*. For Case 2, 4 PCs were chosen for predicting both *E. coli* and *S. cerevisiae*. For Case 1, the average percentage errors for the 100 testing sets were 1.69% for *M. buryatense* and 3.97% for *S. stipitis*. The prediction performance is tabulated in Table B.0.4 and visualized in Figure B.0.4 c where model predictions are compared to measurements. For Case 2, the average percentage error for the 100 testing sets were

3.23% and 1.92% for *E. coli* and *S. cerevisiae*, respectively. The detailed results are provided in Table B.0.5 and Figure B.0.4 d.

Dynamic Case Study: Case 3

The static case studies clearly demonstrate the feasibility of the soft sensor for non-growth binary mixtures. However, it has been well recognized that the cell compositions change dynamically under different culture conditions. To test out whether the proposed soft sensor approach works well for co-cultures that grow together, we conducted Case 3 where *E. coli* KO11and *S. cerevisiae* grew together with glucose as the carbon source till stationary phase. In Case 3, the individual cell mass concentrations were estimated by both the soft sensor approach and a cell counting approach. Because the ground truth of individual cell mass concentrations were not available, the performance of different methods were evaluated by comparing the measured OD600 of the mixed culture with the OD600 reproduced from the estimated individual cell concentrations by linear superposition.

Evaluating Soft Sensor Performance for the Dynamic Case Study 3

The individual cell concentration estimated by the soft sensor and cell counting approaches are given in Figure B.0.5 a,b for *E. coli* and *S. cerevisiae*, respectively. Figure B.0.5 suggests that the estimated individual cell concentrations from the soft sensor approach follow an expected trajectory, *i.e.*, exponential growth phase followed by stationary phase. In addition, the standard deviation among the triplicate samples is significantly small than that from the cell counting approach. Figure B.0.5 c,d shows the comparison between the measured and reproduced OD600 of the mixture samples from

both methods, as well as the percentage error of both methods. The percentage error is calculated as the following

$$\boldsymbol{Error}\% = \frac{\boldsymbol{measured\ OD_{600} - reproduced\ OD_{600}}}{\boldsymbol{measured\ OD_{600}}}$$

Figure B.0.5 shows that the soft sensor approach can almost exactly reproduce the OD600 reading of the mixture samples, with percentage errors less than 5% (when cell density is low) and 1% (when cell density is high), while the cell counting approach shows significant errors throughout. In addition, for the initial sample that was taken right after the inoculation, the individual cell concentration is known (through inoculum volume). The actual individual cell concentrations of *E. coli* and *S. cerevisiae* are marked as solid squares in Figure B.0.5 a,b, which further confirms the accuracy of the soft sensor approach.

Conclusion

In summary, a soft sensor approach was developed to estimate the individual cell concentration in a mixed culture. By using the OD scanning spectrum of the mixture sample as the model input, the soft sensor is able to predict the individual cell concentration accurately, as demonstrated in the two static case studies and the dynamic case study. The dynamic case study suggests that although cell compositions do change due to culturing with other strains, such changes do not deteriorate the soft sensor estimation performance noticeably. The following two possible reasons might explain this: first, although certain cell composition may change (significantly) due to changing culture condition, their resulted changes in overall optical density spectrum may not be significant; second, even if the optical density at a few wavelengths may change

noticeably due to the cell composition change, because the soft sensor utilizes the whole optical density spectrum to estimate the individual cell mass concentration, such local changes do not affect the overall model noticeably. Because we have no ways to separate the two strains and obtain the optical density spectrum of each strain, we have no way to determine for sure which of these two cases would be the reason for the robustness of the developed soft sensor approach. It is very likely that both of them contribute to the robustness of the soft sensor approach. In addition, it is worth noting that the changing cell composition due to culture condition is not limited to mixed cultures; it is true for pure culture as well. It has been reported that cell composition is different between exponential growth and stationary phase[130].However, almost all existing research on single cultures uses OD at a single wavelength to quantity cell concentration without considering such composition changes. This also suggests that the errors introduced by potential cell composition changes are not significant enough to be separately considered. In conclusion, these case studies suggest the developed soft sensor approach offers a fast, simple, and accurate approach to monitor individual cell growth in a mixed culture.

It is worth noting that, in our experiments, cells were centrifuged, washed and resuspended in DI water. To speed up and simplify the process, it is possible to use samples directly from a bioreactor. In this case, cell-free broth (from the same bioreactor that the sample was taken from) should be used as the “blank” for the spectrophotometer OD scan during both soft sensor development and testing. It is also worth noting that the soft sensor approach is not limited to binary mixtures. It can be extended for mixed cultures with more than two components, as long as each component has a unique OD spectrum. However, because training samples with known individual cell concentrations

are required to build the model, the proposed approach only applies to defined mixed cultures. It does not apply to mixed cultures with unknown organisms. Also, the proposed approach does not characterize other properties of the mixed culture such as cell viability.

Appendix C:

Another application of feature based model building by engineering features from raw data and its advantages have been published in one of author's publication. In this application features were extracted in time dimension. Abstract of this publication is given in appendix C. Interested readers can refer to the full paper.

Feature-based Virtual Metrology for Semiconductor Manufacturing

Abstract:

In semiconductor manufacturing, virtual metrology (VM), a.k.a. soft sensor, is the prediction of wafer properties using process variables and other information available for the process and/or the product (*i.e.*, machine data) without physically conducting property measurement. VM has been utilized in semiconductor manufacturing for process monitoring and control. In this work, we discuss the shortcomings of some of the commonly used VM methods and propose a feature-based VM framework. An industrial case study is used to demonstrate the effectiveness of the proposed method.

Refer full paper for details:

K. Suthar, **D. Shah**, J. Wang, Q. Peter He, Feature-based Virtual Metrology for Semiconductor Manufacturing, Computer Aided Chemical Engineering, 44,2018, 2083-2088.