Data-Driven Systems Engineering Techniques for Advanced Manufacturing

by

Kerul Suthar

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
August 7, 2021

Keywords: Virtual Metrology, Soft Sensor, Process Monitoring, Statistic Pattern Analysis, Data-Driven Modeling, Systems Engineering

Approved by:

Q. Peter He (Chair), Associate Professor, Chemical Engineering
Jin Wang (Co-chair), Walt and Virginia Woltosz Endowed Professor, Chemical Engineering
Zhihua Jiang, Auburn Pulp and Paper Foundation Assistant Professor, Chemical Engineering
Selen Cremaschi, Redd Associate Professor, Chemical Engineering
Nedret Billor, Professor, Mathematics and Statistics
Bart Prorok, Professor, Materials Engineering

Abstract

This dissertation presents research performed in the area of data-driven systems engineering to address some challenges of existing sensing and modeling technologies when applied in the emerging field of advanced manufacturing. One major contribution of this work is the development of data-driven machine learning techniques utilizing novel industrial internet of things (IIoT) sensors and applying them to industrial manufacturing processes. The research covers the following three areas: an IIoT Wi-Fi based woodchip moisture estimation system for the pulp and paper industry, a novel feature engineering enhanced virtual metrology methodology for the semiconductor manufacturing industry, and a process modeling and monitoring framework utilizing IIoT vibration sensors for the process industries.

In recent years, IIoT has transformed industry by changing the way industries operate from day to day. Specifically, the advent of tiny low-cost IIoT sensors and high bandwidth wireless systems means even the smallest devices can be connected, monitored, and easily communicate and share data with other devices. Thanks to these advancements, industries have gained access to vast amounts of high-frequency data, i.e., so-called big data. The use of computationally efficient data-driven machine learning techniques to extract valuable information from the big data has led to a significant advancement in the manufacturing industry, such as a more accurate view of the operations, enhancement in scalability and performance, and bridging the gap between production floors and control systems, all leading to more efficient data-informed decision-making.

Machine learning (ML) and artificial intelligence (AI) are at the core of data-driven decision-making for advancement in smart manufacturing. The use of collected data through sensors, when processed with robust machine learning algorithms, has changed the spectrum of

real-time decision-making in industries. However, more often, the rote application of ML algorithms without considering domain knowledge leads to inadequate modeling of the process, such as underfitting or overfitting. These models are undesirable in production environments due to poor performance and/or robustness. Through this work, the author addresses these challenges in various industrial settings and demonstrates that the key to robust and high-performance data-driven modeling is the synergistic integration of domain knowledge and machine learning.

In the first part of this work (Chapter 2), the author proposes a non-destructive, economic, and robust woodchip moisture content (MC) sensing approach utilizing channel state information (CSI) from IIoT based Wi-Fi to address the limitations of the existing technologies in the pulp and paper industry. An experimental design and an algorithmic technique were proposed to handle the confounding factors. To address the challenge that the raw CSI data is very noisy and sensitive to woodchip packing, a feature-based classification system based on statistics pattern analysis (SPA) was proposed in this work, which shows the advantages of domain knowledge combined with machine learning instead of the rote application of ML algorithms on the raw data collected. Specifically, the CSI data collected through IIoT Wi-Fi-based sensors is processed through SPA to extract not only robust and predictive but also physically meaningful features that enable accurate estimation of the woodchip MC with the help of robust ML algorithms.

In the second part of this work (Chapter 3), the author proposes a feature-based virtual metrology (FVM) framework to address the limitations of the existing virtual metrology (VM) methods. In semiconductor manufacturing, VM, also known as soft sensors, predicts wafer properties using process variables. The author explores how batch features can better capture process characteristics and dynamic behaviors than the original process variables. This work also demonstrates how FVM can inherently handle and avoid some of the tedious and time-consuming data preprocessing steps and leads to better predictive models by extracting relevant features. In addition, the author shows how non-linearity and non-Gaussianity can be handled with FVM. The FVM based approach is compared with existing VM approaches to demonstrate its superior predictive ability through a simulated industrial case study and an actual industrial case study.

In the third part of this work (Chapter 4), the author demonstrates how IIoT sensors have great potential in advancing manufacturing process modeling through yet another type of IIoT sensor - accelerometer. This is an extension to a previous work where data from a centrifugal pump IIoT testbed was used to predict the motor speed and water flow rate inside a pipe using machine-learning techniques. In this work, the author compares different levels or extent of feature engineering and examines their impact on model performance. While the modeling of motor speed is relatively less challenging after appropriate feature engineering, efficiently predicting water flow rate requires a fusion of time-domain and frequency-domain features and relatively complex machine learning techniques as the relationship is not linear. This is the main contribution of this work. Through appropriate domain knowledge and feature engineering, superior models are proposed to predict the motor speed and water flow rate in comparison to the application of machine learning techniques on the raw data. The author demonstrates the performance of the predictive models for motor speed and water flow rate and shows that approaches that integrate feature engineering with human learning through exploration achieved superior performance.

The contribution of this work and potential future directions are summarized in Chapter 5.

Acknowledgments

I would like to express my deepest gratitude to my supervisor Dr. Peter He for his motivation and mentorship throughout my Ph.D. project and in preparation of this dissertation. He believed in me throughout the journey, during my failures as well, and without him, this dissertation would not have been possible. I would also like to express immense gratitude towards my co-advisor, Dr. Jin Wang. She has motivated me throughout this journey to keep moving forward, and I have learned so much from her. Her valuable guidance has helped me become an independent researcher. There were times when I faced failures during my journey, but Dr. He and Dr. Wang have always motivated me to look forward and taught me how to move past challenges and learn from previous mistakes. Their unparalleled knowledge helped me improve and thrive as a researcher.

I would like to express my sincere gratitude to my committee members as well. Dr. Zhihua Jiang's advice was crucial in shaping my research, and his experience with the pulp and paper industry surely helped me enhance the quality of my research. Dr. Selen Cremaschi's insightful suggestions and comments helped me grow and develop as a researcher. Dr. Nedret Billor has played a crucial role in my journey, helped me strengthen my knowledge of statistics, and motivated me to consider a Statistics minor. She believed in me, motivated me to explore the field through her courses, and that turned out crucial in shaping my research work. Dr. Bart Prorok's valuable suggestions during my prelims helped me frame my future work and enhance my research. I would also like to thank Dr. Shiwen Mao for agreeing to serve as the University reader on my dissertation committee.

I would like to thank current and past members of the He lab (DE-PSE group) that include Jangwon, Farshad, Arrslan, Farnaz, Jisung, and Alex. Jangwon has been like a brother to me throughout this Ph.D., and without his constant support and motivation, this journey would not have been possible. I will certainly cherish those times in Auburn with you forever. I would also like to thank our Head of the Department, Dr. Mario Eden, for his support during this 5-year journey. I would also like to acknowledge the help of Elaine Manning, Karen Cochran, Naomi Gehling, Emma Goodlett, and Brian Schwieker from the Chemical Engineering department for their help in various matters.

I would like to thank my parents and extended family for providing constant support throughout this journey. My deepest gratitude to my brother and friend, Dr. Raj Thakur, who has been with me in my ups and downs, made this journey easier, and constantly motivated me to keep moving forward. I would also like to thank my friend Varun. His jokes and humor made this journey memorable for sure. I would also like to thank my second family, Moksha, and Aaditya for standing with me through thick and thin. Lastly, sincere thanks to countless other friends in Auburn who made my time at Auburn memorable.

Most importantly, I dedicate this to my lovely and beautiful wife Kruthika. I owe everything that I am today to her. Kruthika, you have been my absolute pillar of strength throughout these years. You helped me keep things in perspective and supported the family throughout my graduate studies. Thank you for being my muse, having unconditional trust in me, and most importantly, earning this degree right along with me. You are the best thing that happened to me in Auburn.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

IoT	Internet of things
IIoT	Industrial internet of things
ML	Machine learning
AI	Artificial intelligence
MC	Moisture content
CSI	Channel state information
VM	Virtual metrology
FVM	Feature based virtual metrology
RPM	Rotations per minute
GPM	Gallons per minute
M2M	Machine-to-machine
SPA	Statistics pattern analysis
PCA	Principal component analysis
PCR	Principal component regression
KF	Kalman filter
TSA	Time series analysis
CMP	Chemical mechanical planarization
OES	Optical emission spectroscopy
PLS	Partial least squares
DOE	Department of energy
CWT	Cell wall thickness
MPC	Model predictive control
ASTM	American society for testing and materials
RF	Radiofrequency
NIR	Near infrared

NIC	Network interface card
OFDM	Orthogonal frequency domain multiplexing
MIMO	Multi-input multi-output
m_D	Oven dry weight
m_T	Total mass
m_W	Mass of water
LOS	Line of sight
LSTM	Long short-term memory
RNN	Recurrent neural network
RBF-NN	Radial basis function – Neural Network
SVM	Support vector machine
LDA	Linear discriminant analysis
HOS	Higher order statistics
CLT	Central limit theorem
μ	Mean
Med	Median
Max	Maximum
Min	Minimum
IQR	Interquartile range
D_{mean}	Mean absolute deviation
D_{med}	Median absolute deviation
C_V	Coefficient of variation
γ	Skewness
κ	Kurtosis
R_{xy}	Cross-correlation coefficient
$MDCS_{xy}$	Mean difference of consecutive subcarriers
PCS	Principal component subspace
MCVT	Monte Carlo validation and testing
ANN	Artificial neural network
XGBoost	Extreme gradient boosting
NN	Neural network

AFM	Atomic force microscope
W2W	Wafer-to-wafer
IM	Integrated metrology
MLR	Multiple linear regression
ARMA	Autoregressive moving average
ARIMA	Autoregressive integrated moving average
SVR	Support vector regression
RPLS	Recursive partial least squares
FIFO	First-in-First-out
DTW	Dynamic time warping
DDTW	Derivative dynamic time warping
RMSE	Root mean square error
RFVM	Recursive feature based virtual metrology
MAPE	Mean absolute percentage error
R^2	Coefficient of determination
MRR	Material removal rate
WWNU	Within wafer non uniformity
MC	Monte Carlo
PC	Principal components
EWMA	Exponentially weighted moving average
Hz	Hertz
GHz	Giga Hertz
UET	Unix epoch time
NIPALS	Non iterative partial least squares
LOF	Local outlier factor
FFT	Fast Fourier transform
DFT	Discrete Fourier transform
PSD	Power spectral density
SNR	Signal-to-noise ratio
THD	Total harmonic distortion
SINAD	Signal to noise and distortion ratio

DL	Deep learning
DNN	Deep neural networks
TPE	Tree parzen estimator
EI	Expected improvement

1. Chapter 1. Introduction and dissertation Outline

The major contribution of this work is the development of data-driven machine learning techniques utilizing novel industrial IIoT sensors and applying them to industrial manufacturing processes. IIoT refers to billions of devices around the world that are connected via internet collecting and sharing data. Kevin Ashton[1] first coined the term in 1999 although it took a long time for the technology to catch up with the vision. Due to the significant advancement in IIoT technologies and the ubiquity of wireless networks, it is possible to replace or augment the traditional sensors with IIoT sensors. Adding IIoT sensors and connecting them wirelessly introduce a level of data intelligence that enables them to communicate in real-time without any human intervention[2].

IIoT brings together critical assets, advanced predictive and prescriptive analytics, and modern industrial workers[3]. Smart machines and real-time analytics have been proposed to take advantage of the massive data produced by machines or systems in a variety of industrial settings, which help drive business decisions faster and more accurately. This network of a multitude of smart devices connected by communication technologies results in a system that can monitor, collect, exchange, analyze, and deliver valuable insights as never before. As a result, IIoT has led to a significant change in the way industries operate from day to day[4]–[8]. Through the combination of machine-to-machine (M2M) communication and industrial data analytics, IIoT is driving unprecedented efficiency, productivity, and performance levels. In addition, the small size and low cost of IIoT devices make it possible to equip manufacturing systems in large numbers providing detailed spatial and temporal information that otherwise would not have been possible.

Due to recent advancements in IIoT it has led to a significant increase in their use in the manufacturing industry[7], [9]. In an industrial setting, IIoT and ML are the two important aspects that pave the road towards smart manufacturing (also known as advanced manufacturing, industrial 4.0, or intelligent manufacturing). Smart manufacturing is a broad concept and a combination of various technologies and solutions, which, when implemented in a manufacturing ecosystem, can help in optimizing the entire manufacturing process and thus increase overall profits, reduce costs, improve efficiency, and avoid downtime[10]–[13]. Smart manufacturing is about harnessing the data in the most efficient way possible for smart data-driven decision-making telling users "What to do" and "When to do it". Thanks to IIoT, the collection of so-called big data is easier than ever before. This data can be analyzed at every step in a production process leading to efficient data-driven decision making.

The goal of this study is to explore the utilization of novel IIoT sensors for industrial applications and address the challenges in ML modeling when the traditional pure data-driven ML techniques are applied to model the data collected from IIoT sensors. Specifically, collecting a vast amount of high-frequency data via IIoT sensors is one aspect but extracting and processing the information within this data with the right tools is more critical to the data-driven decision-making process to enhance and optimize operations in an industrial setting. For example, big data has its own challenges, and its characteristics can be summarized by 4 V's: Variety (different types of data), volume (systems needs to be able to handle the massive amount of data in real-time), velocity (the speed at which data is generated) and veracity (trustworthiness of data in terms of accuracy). This study aims to provide insights into the use of IIoT devices and data-driven ML techniques for smart manufacturing. Specifically, this work demonstrates that processing big data directly through data-driven ML techniques can lead to incomplete or misleading information and insights. For example, when considering predictive modeling in manufacturing, this study shows that the use of ML techniques directly on collected data often leads to poor performance of ML algorithms, and the raw data needs to be processed more efficiently through effective feature engineering. For example, the data collected from IIoT devices are often noisy, show no clear trend or relationship, and needs extensive preprocessing. The goal of this study is to demonstrate the synergistic integration of domain knowledge and machine learning. The author addresses existing challenges in various industrial business settings through IIoT and ML techniques combined with domain knowledge to enhance efficiency, reduce costs and downtime, increase the overall profitability of industrial systems, and take a step towards smart manufacturing.

In chapter 2, the author proposes a multiclass woodchip moisture classification approach using IIoT Wi-Fi and ML techniques. This chapter explains the existing problem in the pulp and paper industry related to the moisture estimation of woodchips and a review of existing solutions and their limitations. Then the author proposes IIoT based Wi-Fi sensors as a solution due to their favorable characteristics. The feasibility of using IIoT Wi-Fi sensors for moisture estimation is investigated in detail using preliminary experiments. After that, the experimental data collection procedure, equipment, and data characteristics are discussed, followed by the predictive modeling using raw data. The author discusses the drawback of using noisy raw data and the disadvantages when used in a process industry. As a solution to these limitations, the author proposes a SPA based feature engineering approach to extract so-called smart data from the raw big data. When combined with robust machine learning techniques, this domain knowledge-based smart data provide effective moisture estimation in the pulp and paper industry. SPA is discussed in detail, along with a description of the features and an insight into the data using dimension reduction techniques such as principal component analysis (PCA). Further, different linear and non-linear ML approaches are considered, and their classification performance is compared. The author also provides a brief description of each of these ML approaches and their critical hyperparameters.

The work demonstrates that classification accuracy alone is not a good performance metric for industrial applications, and the practical implications of misclassification must also be considered. Lastly, the conclusions and future work are discussed.

In chapter 3, a novel next-generation feature-based virtual metrology (FVM) framework, is proposed to address the existing challenges in the semiconductor industry. This work focuses on the prediction of wafer properties using process variables and other information available for process monitoring without physically conducting property measurement. The author describes the need for an efficient approach for predicting wafer properties without actual physical measurement to reduce costs and downtime, and to increase the overall efficiency of the process. A brief review of existing VM approaches is provided, followed by the proposed FVM. A detailed explanation of the approach and properties is provided. The author describes how the feature-based approach can eliminate some of the data pre-processing steps directly, which are common issues in a batch process such as a typical semiconductor process. These data pre-processing steps include data mismatch, trajectory shift and alignment. Then, the author discusses various performance metrics for different VM approaches. The author also investigates how FVM addresses process non-linearity inherently to achieve superior performance than other existing VM approaches. To evaluate the proposed approach, two different case studies are demonstrated, and their performance is compared with other existing approaches, including principal component regression (PCR), partial least squares (PLS), kalman filter (KF), time series analysis (TSA), etc. The first case study is a simulated dataset based on a chemical mechanical planarization (CMP) process. The second case study describes a real industrial dataset based on plasma etch, where optical emission spectroscopy (OES) signals are used to predict the wafer properties. Lastly, conclusions and future directions are described in the chapter.

In chapter 4, the study focuses on the use of non-invasive IIoT sensors such as vibration sensors for predictive modeling in the process industry. The process of data collection along with the centrifugal testbed and the sensors are described in this work. This is followed by the performance of approaches such as PLS on raw vibration data collected with accelerometer sensors. PLS is used to predict the motor speed, and the water flow inside the pipe of the system. This work shows how rote application of ML can lead to misleading results and fails to capture the relationship between the explanatory and response variables. The author describes further feature engineering to extract features in the frequency domain along with some data characteristics. Compared to modeling using raw data, the performance is significantly improved by integrating domain knowledge with ML algorithms.

Further, the author describes the main contribution of this work, where a combination of features in the time domain (i.e., the features that describe signal behavior) and features relevant to peak frequency in the frequency domain are used to further improve the predictive performance. The author describes how the relationship between the data and RPM is easily modeled using approaches such as PLS while, for flowrate modeling, a relatively complex model such as k-neighbors regression is proposed, and the performance is further improved. Here, the author again demonstrates how rote application of ML can lead to poor modeling performance. Lastly, the author discusses some limitations and some insights into future directions.

In chapter 5, the author summarizes the contributions of this work and sheds some light on potential future directions in this area of research.

2. Chapter 2. Multiclass moisture classification in woodchips using IIoT Wi-Fi and machine learning techniques

For the pulping process in a pulp & paper plant that uses woodchips as raw material, the MC of the woodchips is a major process disturbance that affects product quality and consumption of energy, water, and chemicals. Existing woodchip MC sensing technologies have not been widely adopted by the industry due to unreliable performance and/or high maintenance requirements that can hardly be met in a manufacturing environment. To address these limitations, we propose a non-destructive, economic, and robust woodchip MC sensing approach utilizing CSI from industrial IIoT based Wi-Fi. While these IIoT devices are small, low-cost, and rugged to stand for harsh environment, they do have their limitations such as the raw CSI data are often very noisy and sensitive to woodchip packing. To address this, SPA is utilized to extract physically and/or statistically meaningful features from the raw CSI data, which are sensitive to woodchip MC but not to packing. The SPA features are then used for developing multiclass classification models using various linear and nonlinear ML techniques to provide potential solutions to woodchip MC estimation for the pulp and paper industry. This work also demonstrates that classification accuracy alone is not a good performance metric for industrial applications, and the practical implications of misclassification must also be considered.

2.1. Introduction

The US is one of the largest producers of pulp products and one of the largest producers of paper and paperboard products. The US pulp and paper industry ranks third in energy consumption among US industries and spends over $7 billion annually on purchased fuels and electricity [14]. The pulping process, which converts woodchips into pulp by displacing lignin from cellulose fibers, is one of the most energy-intensive processes and has been identified by the ENERGY STAR® and the Department of Energy (DOE) reports as a significant opportunity to improve energy productivity and efficiency of the industry [14]–[16]. Currently, chemical pulping processes produce the vast majority of the US pulp, and most of them utilize continuous Kamyr digesters. A Kamyr digester is a complex vertical plug flow reactor where the woodchips react with an aqueous solution of sodium hydroxide and sodium sulfide, also known as white liquor, at elevated temperatures to remove lignin. For Kamyr digesters, the incoming woodchip moisture content (MC) is a significant source of disturbance that affects the cooking performance, as it dilutes the white liquor concentration, therefore, reducing the delignification reaction rate. In this work, wet basis MC is used, which is defined as the following:

$$\boldsymbol{MC} = \frac{\boldsymbol{m_W}}{\boldsymbol{m_T}} \times \mathbf{100\%} = \frac{\boldsymbol{m_W}}{\boldsymbol{m_W + m_D}} \times \mathbf{100\%} \qquad (2.1)$$

where $\boldsymbol{m_W}$, $\boldsymbol{m_D}$, and $\boldsymbol{m_T}$ represent the mass of water, dry wood, and total mass, respectively.

Currently, the woodchip MC is not measured in real-time due to the lack of affordable, reliable, and easy-to-maintain sensors [17]. Instead, woodchip MC is commonly measured only four times per year corresponding to the four seasons and used to determine the operation parameters such as chemical usage and cooking temperature. Because this significant process disturbance is unmeasured, the performance of existing control solutions is often unsatisfactory and process engineers often overcook the woodchips to ensure pulp quality, which results in significant loss of pulp yield, overuse of heat/energy, and chemicals. Chemical overuse also adds burdens to the downstream processes, such as washing and evaporation, and results in increased energy and chemical usages for downstream processes as well. It is worth noting that there have been significant efforts and advancements in the modeling and control of chemical pulping over the past decade [17]. In particular, progress has been made on multiscale modeling of Kraft pulping processes to capture the evolution of fiber morphology such as fiber length, porosity, and cell wall thickness (CWT) of cooked pulp [18], [19]. A recent study integrates macroscopic and microscopic models of the Kraft pulping process to develop an inferential model predictive control (MPC) to handle pulp grade transitions better[20]. These efforts have not explicitly considered the woodchip MC variability in a production environment, and this information, if made available, can be directly incorporated into these models for improved model accuracy in practical applications.

The oven-drying method is a direct and precise method based on the weight loss after a drying process, with a standard defined by the American Society for Testing and Materials (ASTM) [21], [22]. However, it is an offline test that takes 24 hours, and is mainly used for validating other indirect methods. A variety of indirect sensing methods have been examined for measuring woodchip MC online, including technologies that are based on microwave [23], radio-frequency (RF) [24], capacitance [25]–[27], Resonant half-wave antenna [28], near-infrared (NIR) [29], [30] and X-ray [31], [32]. However, these methods have not been widely adopted by the industry due to poor performance and/or high maintenance requirements that can hardly be met in a manufacturing environment.

To address the robustness and performance limitations of the existing methods, we propose a non-destructive, economic, and robust approach based on 5 GHz IIoT short-range Wi-Fi and use

CSI to predict MC in woodchips. CSI data have been used for moisture and mildew detections in wheat[33]–[35]. However, woodchip MC classification is a much more challenging task due to the much bigger size and significantly more heterogeneous in both size and shape of the woodchips than those of wheat. Because of that, the woodchip packing or arrangement in the container is expected to have a significant impact on the CSI data, *i.e.*, woodchip packing is a strong confounding factor to MC level. There are generally three ways to address confounding variables: elimination, measuring, and randomization. Since woodchip packing cannot be eliminated nor measured, randomization is the approach taken in this work to address it. In addition, our recent studies have shown that IIoT sensors have their own shortcomings, including significant noise, missing values, and/or irregular sampling intervals, which result in messy big data and lead to low performing models when directly fed to machine learning algorithms [36], [37]. Because of these challenges, the normalized or PCA transformed raw CSI data, which were used for wheat MC classification, are no longer sufficient for woodchip MC classification. To address it, the SPA framework that we developed previously [38]–[41] is used to extract robust and predictive features from the raw, noisy CSI data. These features are shown to be sensitive to the MC levels but insensitive to the packing of the woodchips. It is worth noting that SPA features are physically and/or statistically meaningful while other algorithmically generated features (*e.g.*, square, square root, exponential, etc.) or kernel-type features are often unintuitive. SPA also eliminates the data preprocessing steps (*e.g.*, outlier detection and handling, environmental noise removal) that were required in previous studies[33]–[35]. These two strategies utilized for addressing a confounding variable and for extracting predictive and meaningful features from raw CSI data are two of the main contributions of this work. Another contribution of this work is the systematic study of different state-of-the-art linear and nonlinear classification techniques, as well as individual vs.

ensemble classification, for woodchip MC classification using CSI data. Finally, classification accuracy has been commonly used in previous studies for evaluating classifier performance. We show that the classification accuracy alone is not a good performance metric, and the practical implications (*e.g.*, cost) of misclassification must also be considered.

The remainder of this work is organized as follows. A brief background on CSI and feasibility study for using CSI in woodchip MC detection are presented in Section 2.2. Section 2.3 outlines the experimental setup and data collection procedure. Section 2.4 discusses the challenge of classification using raw data and the need for feature engineering, followed by the proposed approach based on SPA for feature extraction. The classification approaches studies in this work are introduced in Section 2.5, and the hyperparameter optimization approach used in this work. In Section 2.6, the results from different classification techniques are discussed in terms of both classification accuracy and robustness. The practical implications of these results are also discussed. Finally, the conclusion and future work are discussed in Section 2.7.

2.2. Channel State Information and feasibility for woodchip MC classification

2.2.1. Channel State Information (CSI)

Using Wi-Fi cards such as the Intel Wi-Fi link 5300 network interface card (IWL5300 NIC), it is convenient to collect CSI measurements that record the channel variation during the propagation of wireless signals. In this work, CSI is extracted by modifying the open-source device drivers for IWL5300 based on CSITool [42]. Similar tools are available based on Atheros chipsets as well [43]. CSI amplitude and phase data are collected in this work using IWL5300 NIC by configuring the transmitter and receiver in injection and monitor modes, respectively. We use Lenovo ThinkPad systems equipped with Linux-based OS 14.02 and kernel version 4.2 due to the version-specific CSI tool. Both systems are equipped with IWL5300 NIC with a modified driver and firmware for data collection. Orthogonal frequency-division multiplexing (OFDM) is often utilized to deal with impairments in wireless propagation such as frequency selective fading. In OFDM signal modulation, a single data stream is split into multiple orthogonal subcarriers at different frequencies to avoid interference and crosstalk. The IWL5300 NIC used in this work implements an OFDM system with 56 subcarriers, out of which 30 subcarriers can be read using the CSItool, which is built on IWL5300 NIC using custom modified firmware and open-source Linux wireless drivers [42]. Each channel matrix entry is a complex number, with signed 8-bit resolution each for the real and imaginary parts. It specifies the gain and phase of the signal path between a single transmit-receive antenna pair. For example, the channel response of the i^{th} subcarrier can be represented as:

$$CSI_i = |CSI_i| \exp\{\angle CSI_i\} \qquad (1.2)$$

where $|CSI_i|$ is the amplitude response of the i^{th} subcarrier and $\angle CSI_i$ is the phase response.

CSI can reflect indoor channel characteristics such as multipath effect, shadowing, fading, and delay [44]. Our hypothesis is that the water content in the woodchips has a detectable impact on the strength and/or the phase of the signals that are received on the receiver side. In other words, woodchips at different MC levels would lead to different characteristics of CSI signals in terms of amplitude and/or phase responses. Therefore, ML algorithms can be utilized to correlate these characteristics to woodchip MC levels.

In this work, two laptops equipped with IWL5300 NIC and modified drivers with specific Linux kernels are used to collect CSI data. One is set in injection mode while the other is set in monitor mode to collect 5 GHz CSI amplitude and phase data. One antenna is used on the transmitter side, while three antennas are used on the receiver side to take advantage of the multiple-input multiple-output (MIMO) systems for improving the diversity of signals [44], [45]. This diversity is exploited in this work to improve the multiclass classification performance. In addition, it is desirable to focus the RF energy in one direction as the woodchips are placed in an airtight box between the transmitter and receiver. Therefore, unidirectional antennas are selected over omnidirectional antennas. As the gain of the directional antennas increase, the coverage distance also increases in that direction. Also, directional antennas for point-to-point connection reduce interferences from other sources. In this work, panel antennas ALFA (ALFA Network, Taiwan) with 66° horizontal beam-width and 16° vertical beam-width are used.

2.2.2. Feasibility test

To test the technical feasibility of CSI to classify woodchips based on MC levels, we collect CSI for woodchips at three distinctively different MC levels (*i.e.*, 52.34%, 20.40%, and 11.93%). Figure 2.1 and Figure 2.2 show the CSI amplitude and phase difference for the 15th subcarrier respectively. As shown in Figure 2.1 and Figure 2.2, there are distinctive differences in amplitude and phase differences of different MC levels from all three antennas. This preliminary feasibility test indicates that it is possible to develop a data-driven model to correlate CSI data with the woodchip MC level. Note that the confounding factor of woodchip packing is not considered here.

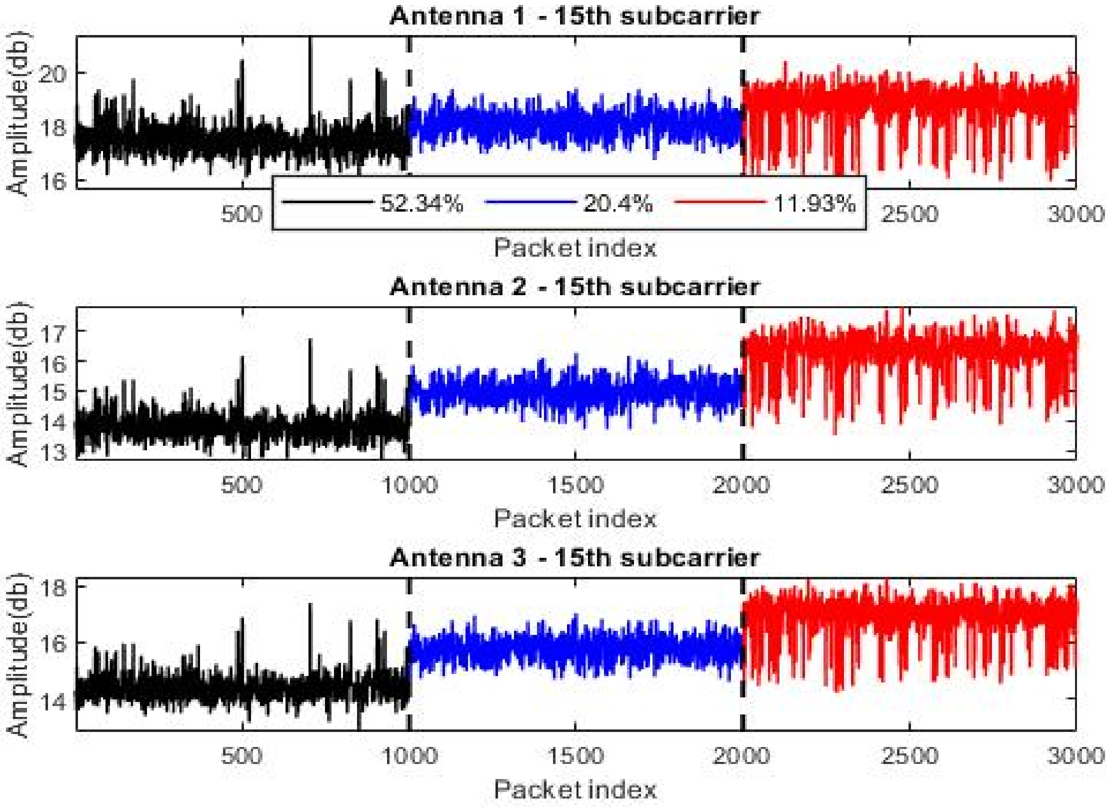


Figure 2.1 CSI signals collected on the three receiving antennas at three different MC levels for amplitude

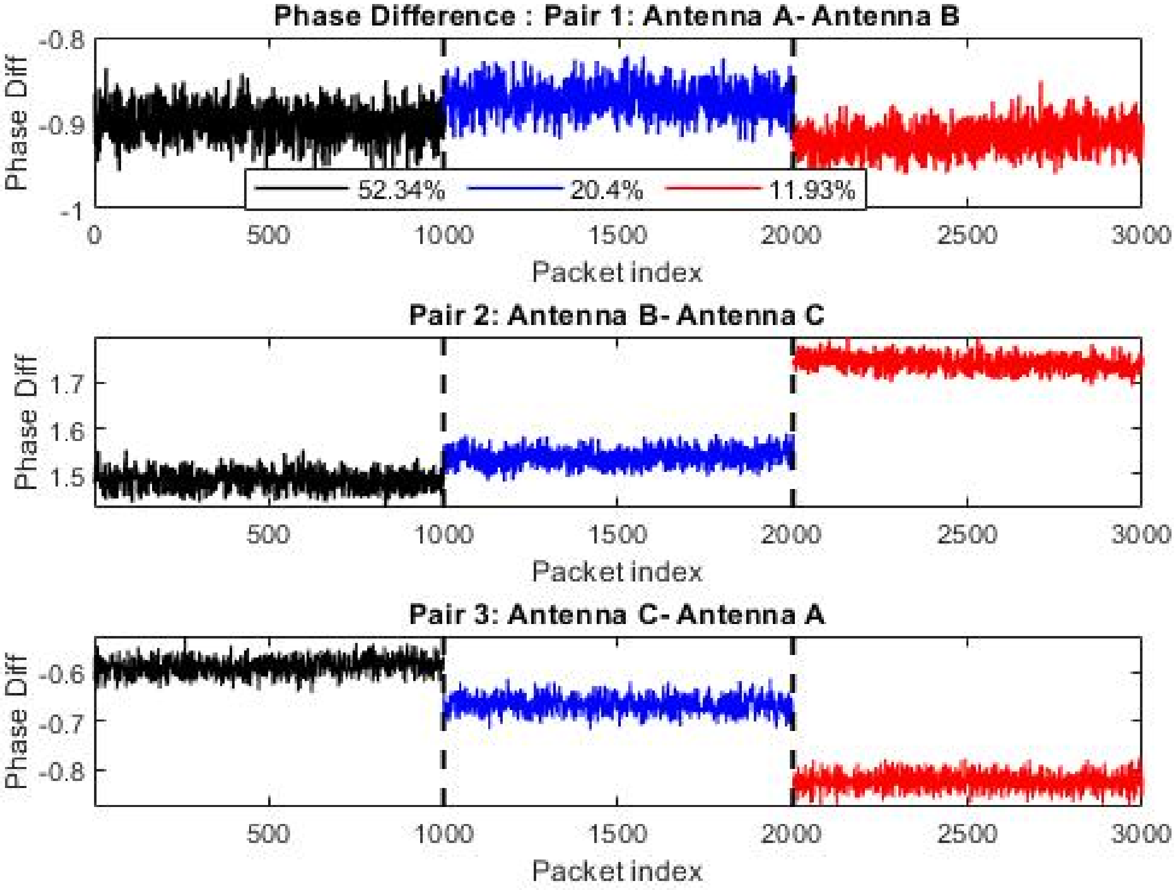


Figure 2.2 CSI signals collected on the three receiving antennas at three different MC levels for phase difference

2.3. Experimental setup and data collection

2.3.1. Experimental setup

With the results from the feasibility test in Section II, we design an experimental setup with antenna positions fixed on an acrylic sheet. The experimental setup is shown in Figure 2.3, where two Lenovo T400s systems equipped with IWL5300 NIC are set 3 m apart. The woodchips at different MC levels are placed at the center (*i.e.,* 1.5 m from transmitting and receiving antennas) in an acrylic container with an air-tight lid to avoid any changes in MC while the data are being collected.

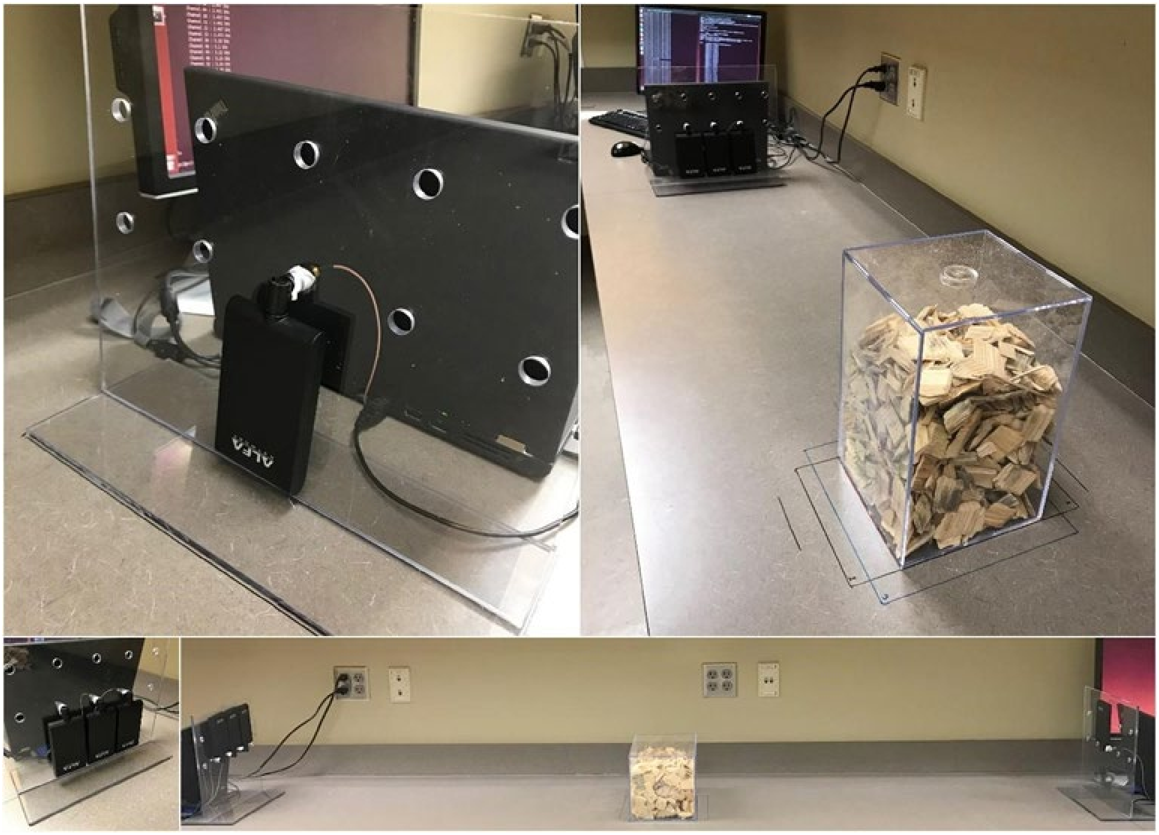


Figure 2.3 Experimental setup for CSI data collection

2.3.2. Data collection

In previous studies, a maximum of 5 MC levels have been studied with a minimum difference of 0.7% in MC [34]. However, this is not nearly sufficient for woodchip MC levels because woodchips are usually stored outdoors, which introduces significant MC variations due to daily weather conditions, and seasonal temperature and humidity changes. In this work, data are collected for 20 different MC classes or levels ranging from 53.39 % to 11.81% on the wet basis (see Eqn (2.1)). Total mass (m_T) is measured during each experiment and oven drying method [21], [22] was performed after all experiments were conducted to determine the oven-dry weight

(m_D). m_T and m_D are then used to determine the mass of water (m_W) and MC according to Eqn. (1.1). The 20 different MC levels are plotted in Figure 2.4. There are two gaps in the tested MC levels, one around 45% and the other around 25%. This is due to the overnight exposure of the woodchips to air in the lab, which should be avoided if a model to be developed for accurate estimation of any MC level in the entire range. Nevertheless, this does not affect our methodology development, nor the conclusions drawn based on the results obtained. This is because there are three regions where MC levels are reasonably separated as shown in Figure 2.4. In addition, MC levels are narrowly separated at the high MC region and even more so at the low MC region. The minimum difference between MC levels is 0.05%. If a model can correctly classify MC levels with such narrow difference, we expect it to work if more MC levels were included in the middle range with wider difference such as 1%, which is sufficient for pulping process optimization and control.

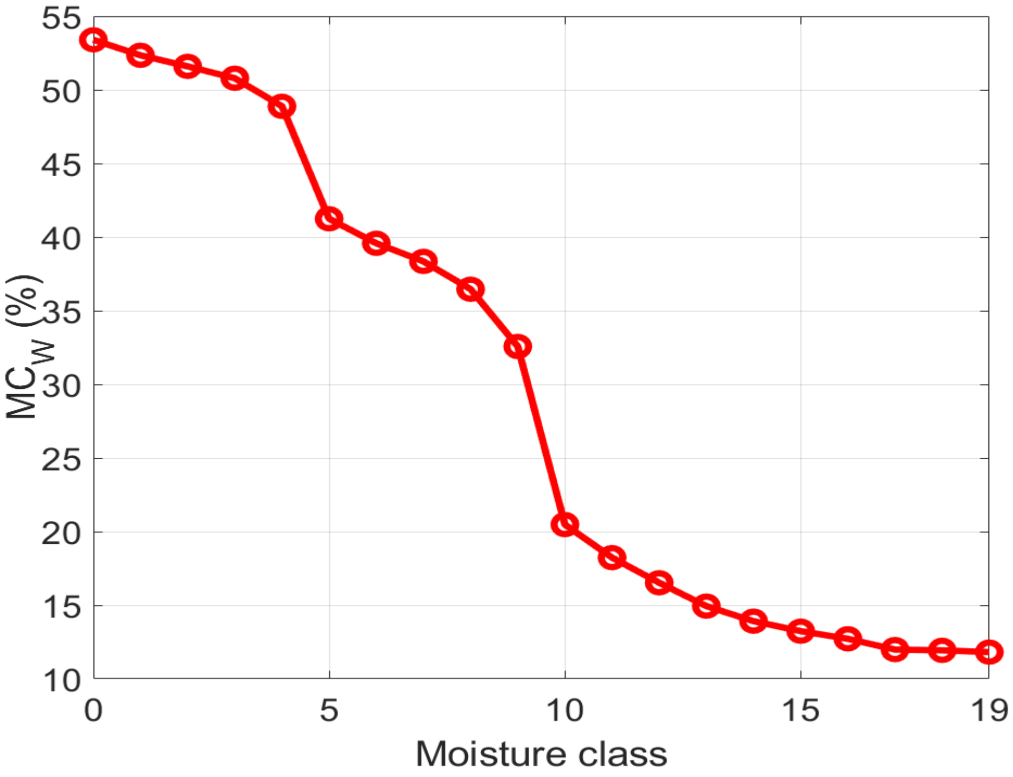


Figure 2.4 20 different MC classes/levels for experimental data collected

As discussed previously, the woodchip packing or arrangement in the container is expected to have a significant impact on CSI data. Based on the principle of randomization for addressing this confounding factor, the woodchips within the air-tight box are shuffled 10 times for each MC level. In other words, for each MC level, 10 datasets (i.e., samples) are collected corresponding to 10 shuffles. Therefore, the experiments generate totally 200 samples for all 20 MC levels. For each sample, 10,000 packets were sent from the transmitter (setup in injection mode) to the three receiver antennas (setup in monitoring mode). Data are collected only for the line of sight (LOS) scenario, i.e., the woodchip container is placed in the middle of the center line between the transmitter and the receivers.

2.4. Feature engineering and selection

For wheat MC classification, normalized raw data were used in long short-term memory (LSTM) recurrent neural network (RNN) [34] and Radial basis function-neural network (RFB-NN) [46], while principal component scores from normalized raw data were used in support vector machines (SVM) [35]. In the next section, we show that raw data are poor features for woodchip MC classification due to the challenges discussed previously in Sec. 2.1. In addition, the Wi-Fi packets are independent from each other (*i.e.*, serially uncorrelated) as evidenced by the close-to-zero autocorrelation coefficients beyond lag 0. Therefore, there is no reason to use an RNN such as LSTM to account for the serial dependency or dynamics of packets.

2.4.1. The challenges of using raw CSI data as features

As discussed in the previous section, for each MC level we shuffle the woodchips 10 times and collect CSI data for each shuffle to address the confounding factor of woodchip arrangement or packing. Figure 2.6 and Figure 2.5 show the raw CSI data of amplitude and phase difference for woodchips at five distinctively different MC levels with 10 shuffles at each MC level. The five MC levels are 53.29%, 41.24 %, 32.57 %, 20.47 % and 11.81 %, in that order where they are plotted in Figure 2.5 and Figure 2.6. For the sake of better visualization and easier interpretation, only 100 packets from the 10th subcarrier for each shuffle are plotted. From *Figure 2.5* and *Figure 2.6*, the impact of shuffling can be seen in both amplitude and phase difference, although it is more

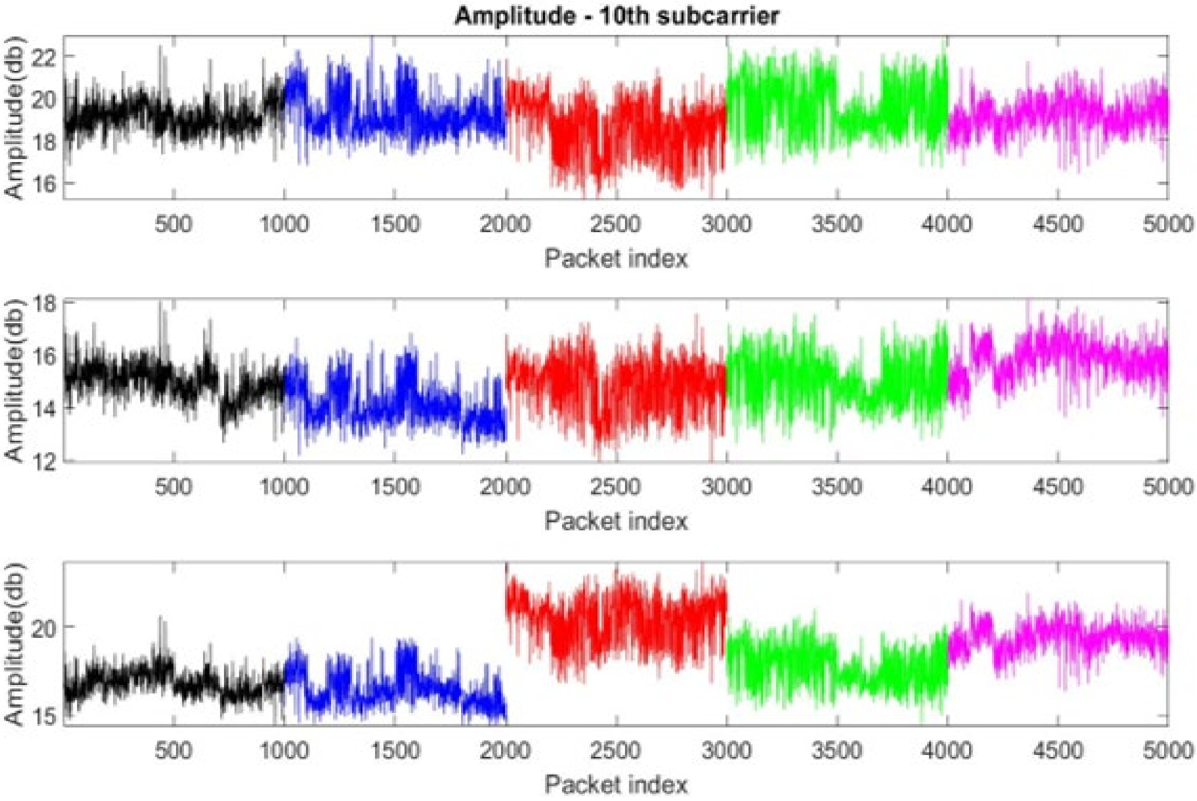


Figure 2.5 Raw CSI data for 5 different MC levels showing 10 shuffles for each MC level for amplitude

obvious in the phase difference. The observation confirms our earlier suspicion that packing or woodchip arrangement is a significant confounding factor to MC level. In addition to packing, another challenge is the significant noises presented in both amplitude and phase difference. Finally, Figure 2.5 and Figure 2.6 show no clear trend or pattern in amplitude or phase difference that correlates with MC levels. All these factors present significant challenges to model MC level with raw CSI data. As an illustrative example, we use linear discriminant analysis (LDA) to

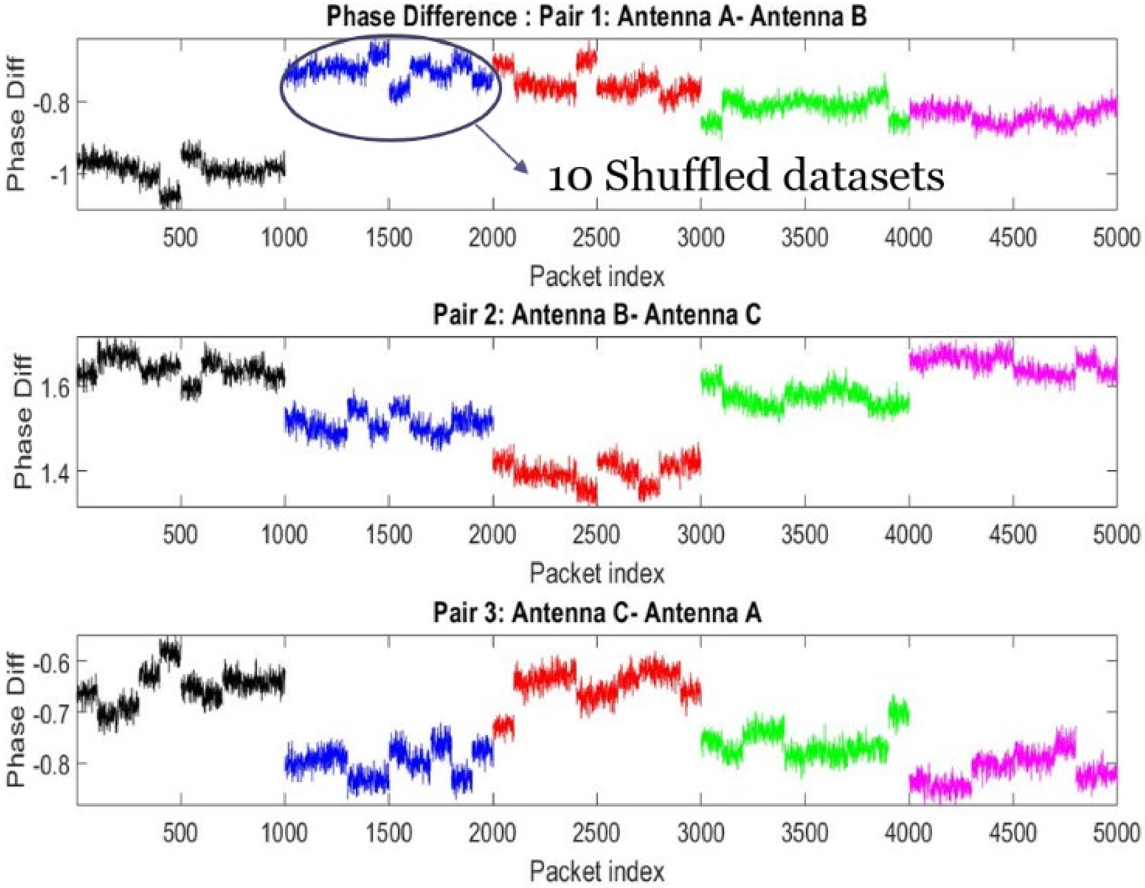


Figure 2.6 Raw CSI data for 5 different MC levels showing 10 shuffles for each MC level for phase difference

perform classification using the raw CSI data, with either amplitude, or phase difference, or both. For training, 9 samples are randomly selected from 10 shuffled samples at each of the 20 MC levels, which results in 180 training samples. The remaining one shuffled sample from each MC level is used for testing after the classification model is trained. This process is repeated 100 times, resulting in 100 Monte Carlo runs and the classification results are shown in Figure 2.7.

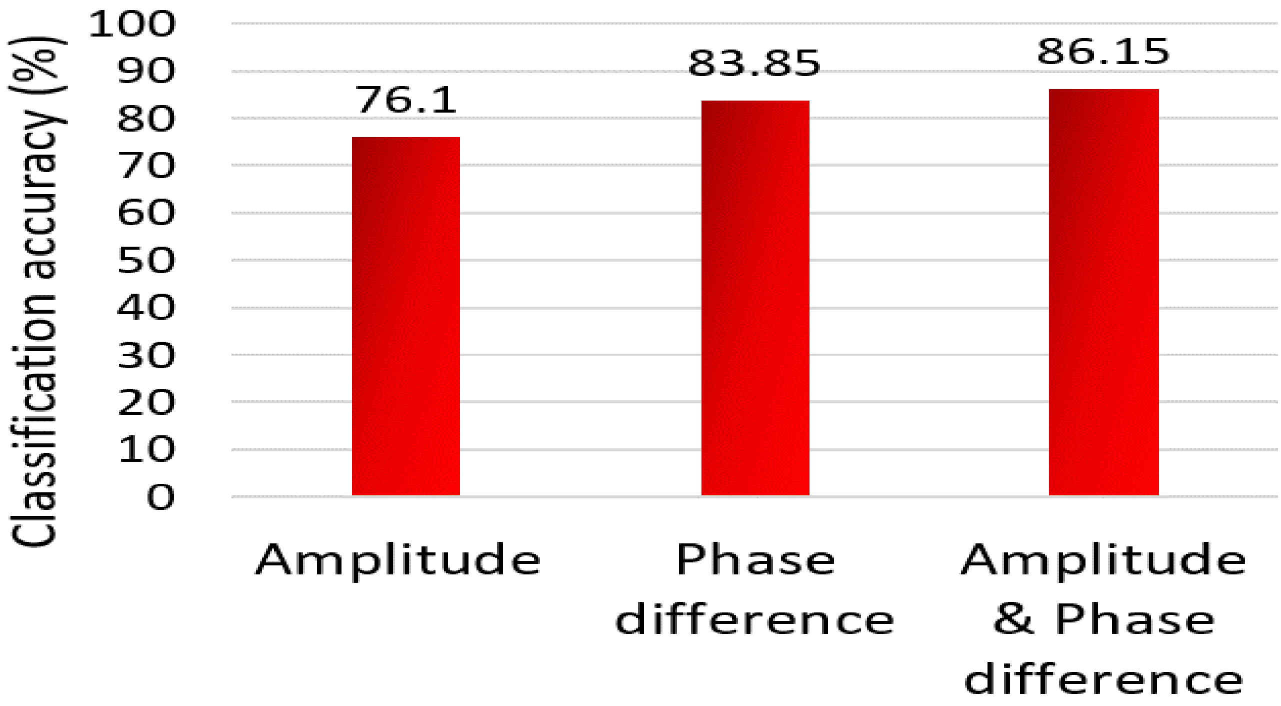
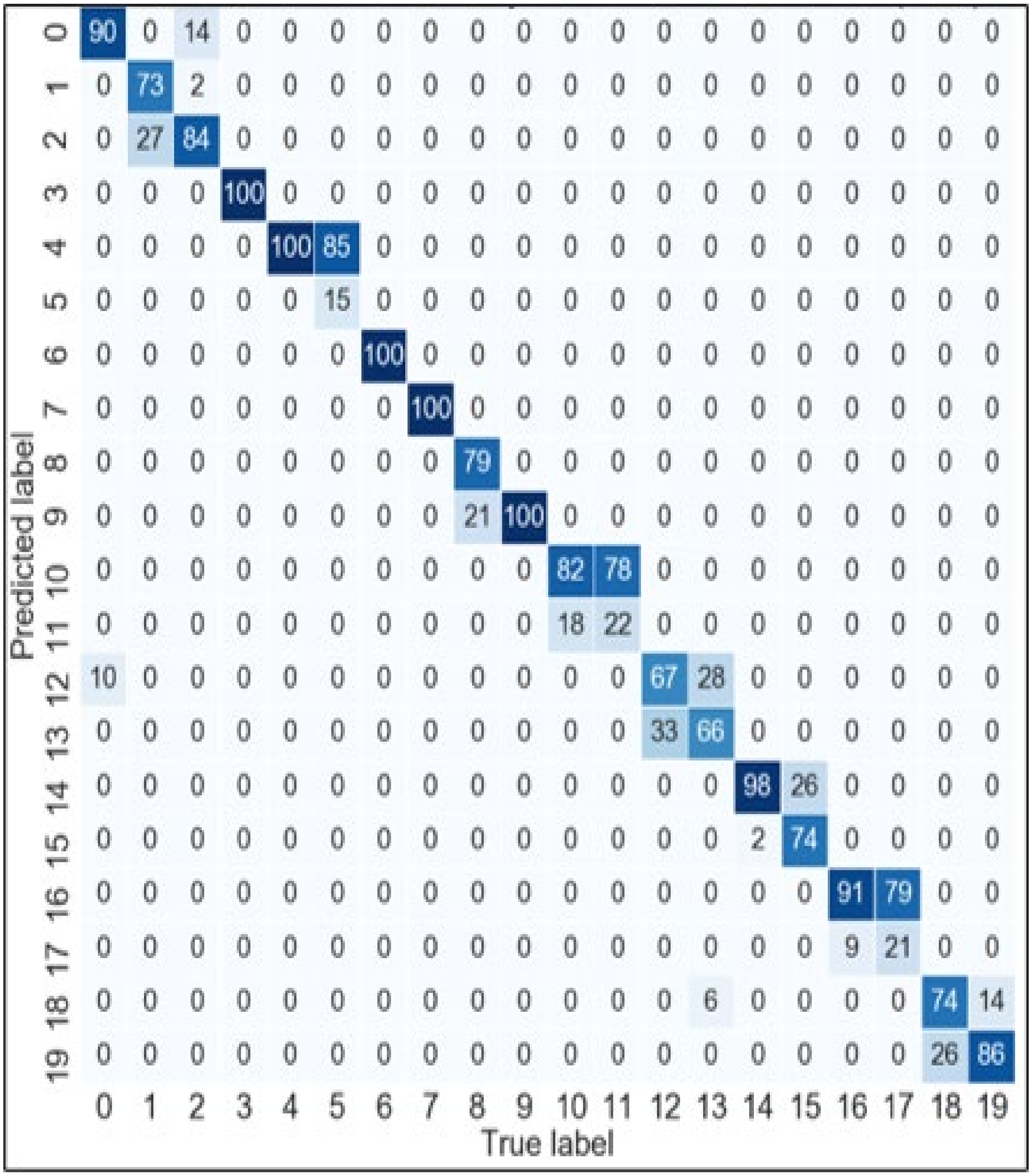


Figure 2.7 Overall classification accuracy using different raw CSI data with LDA classifier based on 100 Monte Carlo runs.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	90	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	73	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	27	84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	100	85	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	79	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	21	100	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	82	78	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	18	22	0	0	0	0	0	0	0	0
12	10	0	0	0	0	0	0	0	0	0	0	0	67	28	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	33	66	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	26	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	74	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91	79	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	21	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	74	14
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	86

Figure 2.8 Classification confusion matrix of 100 Monte Carlo runs when both amplitude and phase difference are used. Since there are 100 samples in each class (true labels), the numbers on diagonal represent the percentage of classification accuracv of classes

For performance evaluation, the classification accuracy of class i is defined as

$$Accuracy_i = \frac{p_i}{n_i} \qquad (2.2)$$

The overall accuracy of all classes is defined as

$$Accuracy = \frac{\sum_{i=1}^{C} p_i}{\sum_{i=1}^{C} n_i} = \frac{\sum_{i=1}^{C} p_i}{N} \qquad (2.3)$$

where C denotes total number of classes, n_i true/known number of samples in class i, $N = \sum_{i=1}^{C} n_i$ total number of samples, and p_i number of correctly predicted samples in class i. Figure 2.7 compares the overall classification accuracy of all classes when different components of the CSI data were used. The comparison indicates that LDA classifier using both amplitude and phase difference performs the best with 86.15% classification accuracy, followed by LDA classifier using phase difference with 83.85% classification accuracy, while the LDA classifier using amplitude alone results in the lowest classification accuracy of 76.10%. Figure 2.8 plots the confusion matrix for the best LDA classifier using CSI amplitude and phase difference, which allows us to dig deeper into the classification results. It is worth noting that the dimensionality is extremely high in this case i.e., 720,000 features and 200 samples. LDA inherently performs dimensional reduction via singular value decomposition. As can be seen from Figure 2.8, the classification accuracy of individual classes ranges from 15% to 100%. It can also be seen that classification accuracy alone is not a good performance indicator. For example, classification accuracy alone would not be able to distinguish the following two scenarios: (1) the actual scenario of misclassifying ten 53.38% MC level samples (class 0 in Figure 2.8) to 16.52 % MC level (class 12); (2) a hypothetical scenario of misclassifying ten 53.38% MC level (class 0) to 51.59 % MC level (class 1). Both scenarios have a classification accuracy of 90%, but with drastically different implications in this application. For example, if MC level is used to control the chemical usage, the former would lead to a significantly worse outcome than the latter. With this point in mind, we see from Figure 2.8 that the classification results using raw CSI data are poor as there are samples misclassified far off their actual classes. In this work, when the predicted class of a sample is off its true class by more than one level, we term it “far-off misclassification” to distinguish it from the scenario of “nearest-neighbor misclassification”, where the predicted class is off true class by one level (either above or below). Based on this definition, there are totally 478 misclassified samples, of which 30 are far-off misclassifications.

2.5. Feature engineering with statistics pattern analysis (SPA)

To address the shortcoming of raw CSI features that lead to not only low classification accuracy but also far-off misclassifications, in this work, SPA is utilized to generate more robust and predictive features. SPA was proposed to supplement the traditional multivariate modeling approaches that directly utilize process variables (e.g., temperature, pressure, etc.) for monitoring, control, and inference purposes. In SPA, the statistics of the process variables, instead of the process variables themselves, are used for modeling. The statistics capture the characteristics of each individual variable (e.g., mean and variance), the interactions among different variables (e.g., covariance), the dynamics (e.g., auto-, cross-correlations), as well as process nonlinearity and process data non-Gaussianity (e.g., skewness, kurtosis, and other higher-order statistics or HOS). SPA is based on the hypothesis that these statistics are sufficient and even better in capturing process characteristics (e.g., static properties and dynamic behaviors) than original process variables. This hypothesis has been supported in various applications, including fault detection [38], [40], [47], [48], fault diagnosis [40], [49], and virtual metrology or soft sensor [39], [50]–[52]. Due to the fact that statistics are computed using a set of observations, they are less affected by noises. In addition, there are robust statistics that are insensitive to outliers. Finally, due to the central limit theorem (CLT), these statistics are asymptotically normally distributed. For these reasons, SPA is selected in this work to extract robust and predictive features from raw CSI data. It is worth noting that SPA does not require preprocessing of the CSI data (i.e., outlier detection and handling, noise removal/reduction) that has been required in previous studies [33]–[35]. The schematic diagram of SPA-based classification is shown in Figure 2.9. In the first step, various statistics are extracted from the CSI amplitude and phase data.

$$\mathbb{P}: \boldsymbol{X} \longrightarrow \boldsymbol{F} \qquad (2.4)$$

Where $\mathbb{P}$ denotes the operator that maps the 3D CSI data array $\boldsymbol{X} \epsilon R^{N\times R\times K}$ containing *N* samples, *R* amplitudes and phase differences of all subcarriers from *K* packets into a feature matrix $\boldsymbol{F} \epsilon R^{N\times S}$ containing *N* samples with each sample now characterized by *S* statistics, such as mean, standard deviation, skewness, and kurtosis of the amplitude of each subcarrier calculated over *K* packets. Note that *K* does not have to be the same across different samples, as long as it is sufficiently large to obtain reliable statistics. This is convenient if different number of packets were received for different samples. For between-variable statistics, between-subcarrier differences are considered, but between-packet statistics are not considered, as packets are independent of each other. In Figure 2.9, $\boldsymbol{Y} \epsilon R^{N\times 1}$ denotes the MC levels for *N* samples. In the second step, a classification model is developed to capture the relationships between the sample features (*i.e.*, statistics) and the response (*i.e.*, MC levels). The SPA framework is a flexible method as different statistics can be added or removed based on how well they capture the relationships between the predictors and the response variables or classes.

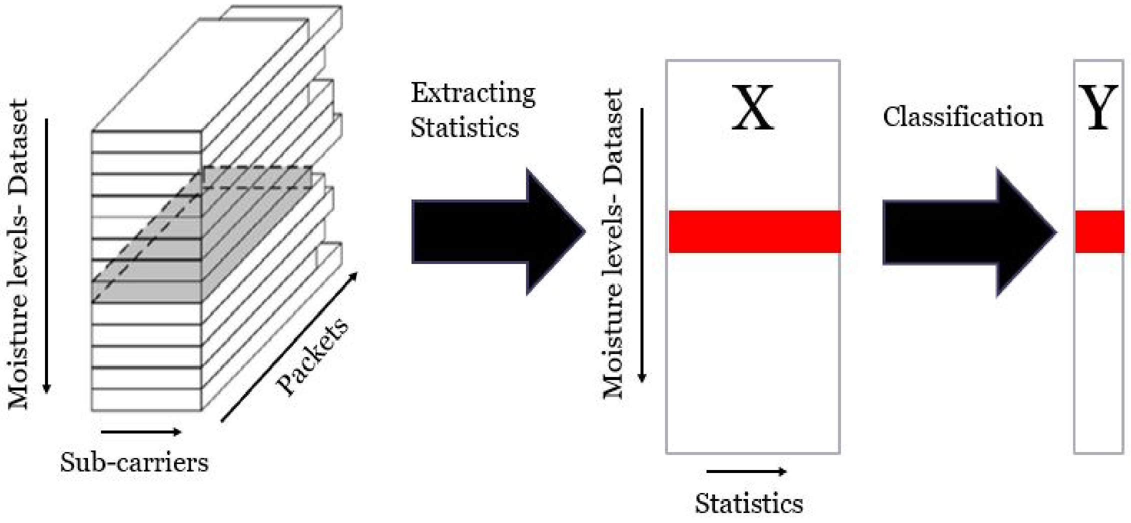


Figure 2.9 Schematic of SPA-based feature extraction for classification

Based on the SPA framework, we extracted 13 statistics (listed in Table 2.1) of 90 amplitude variables (i.e., 3 antennas, each with 30 subcarriers) and 60 phase difference variables (i.e., 2 independent antenna pairs, each with 30 subcarriers). All statistics are computed over 40,000 observations for each of the 200 samples (i.e., 10 samples/shuffles for each of the 20 MC levels). Autocorrelations are not considered because the packets are independent of each other, as evidenced in Figure 2.10, where the sample autocorrelation coefficient of the CSI amplitude from one subcarrier of one antenna is shown, which resembles the pattern of a typical random signal. For cross-correlations, only cross-correlations between subcarriers of the same antenna with lag 0 are considered due to the absence of serial correlation between lags. Figure 2.11 shows the cross-correlation coefficient of CSI amplitude among subcarriers of the same antenna. It can be seen that CSI amplitude (and phase difference, not shown) from different subcarriers are highly correlated, especially the neighboring subcarriers. Because of this observation, we also considered mean difference between consecutive subcarriers. The idea is to capture the relationships between consecutive subcarriers in a more quantitative way than cross-correlation coefficient between them. In this way, the overall shape of the CSI amplitude or phase difference across subcarriers can be captured.

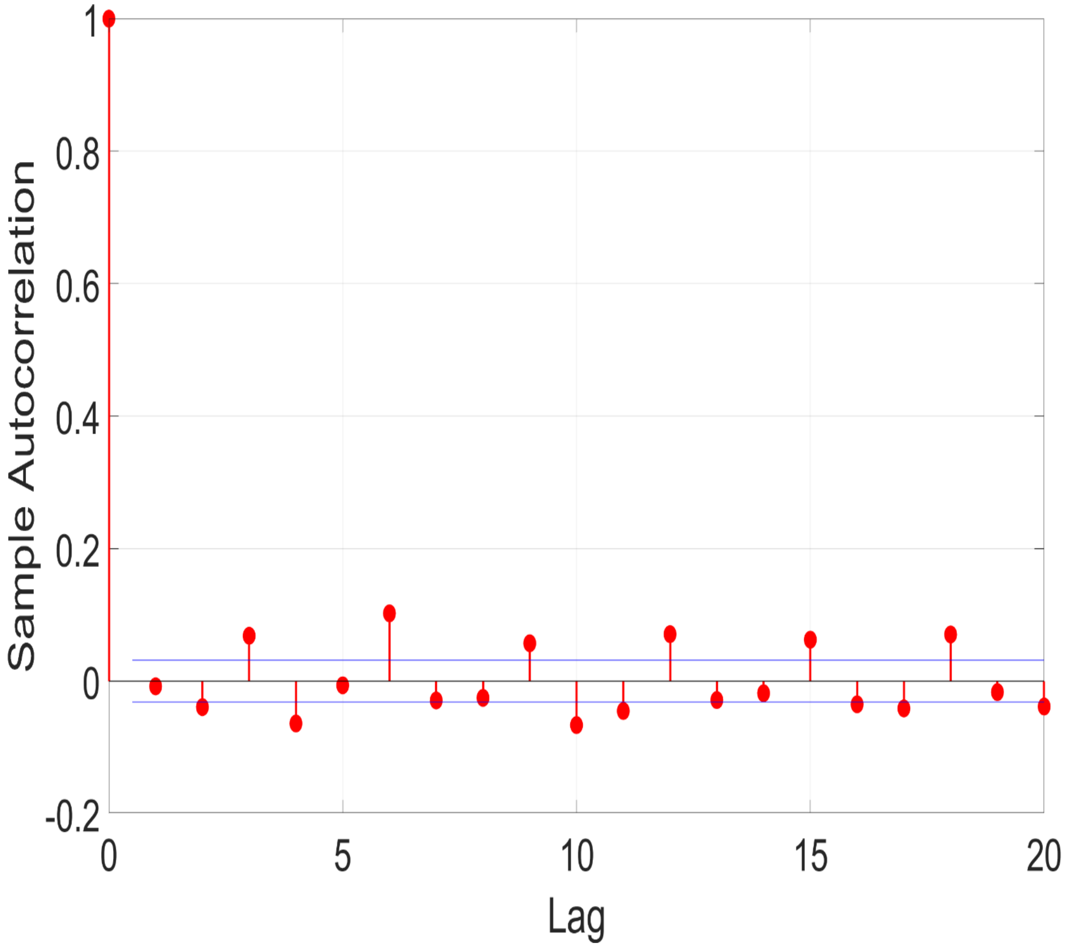


Figure 2.10 Auto-correlation coefficients of CSI amplitude of one antenna subcarrier over 40,000 packets show no significant serial correlation among packets

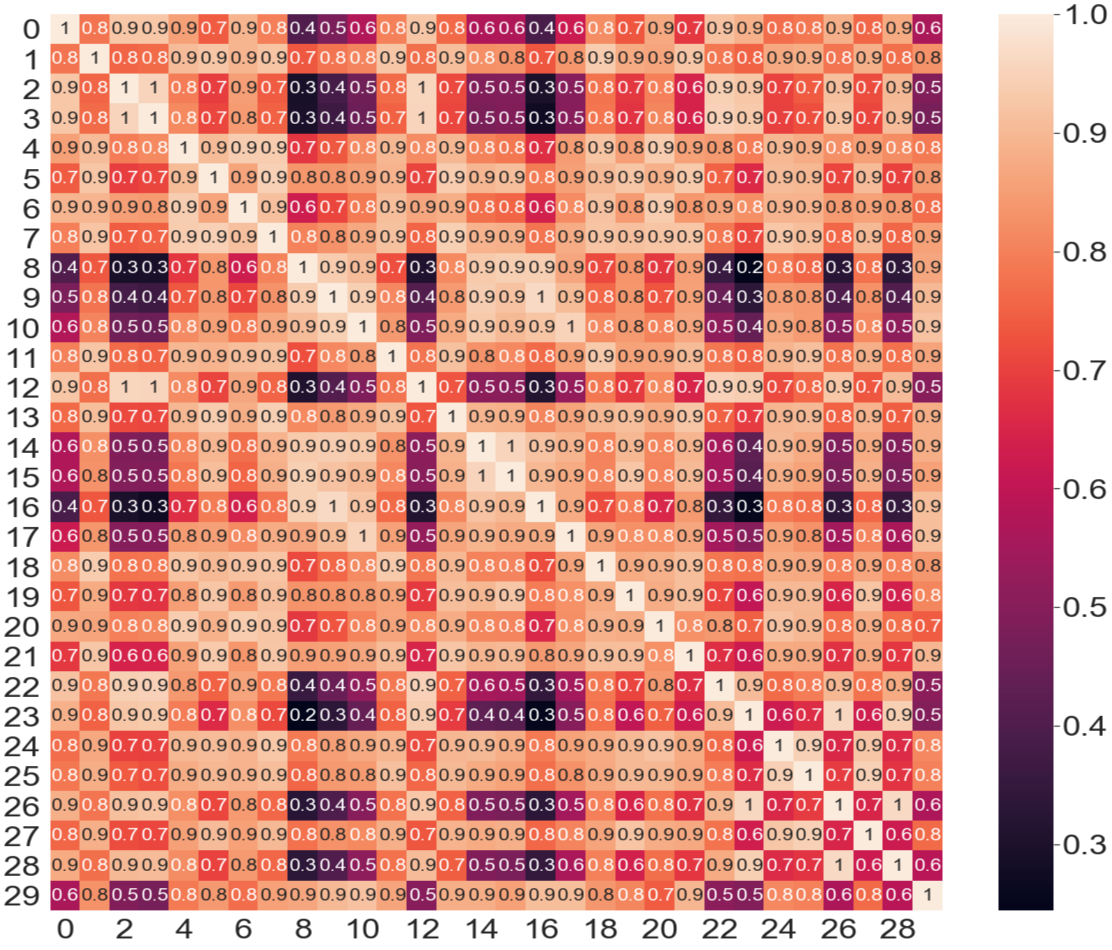


Figure 2.11 Cross-correlation coefficients of CSI amplitude between subcarriers of one antenna show high correlations, especially between consecutive subcarriers.

Table 2.1 Statistics considered as features in this work

Statistics	Definition	Statistics per sample
Mean	$\mu(x) = \frac{1}{K}\sum_{i=1}^{K} x_i$, where x is a CSI amplitude or phase difference variable	150
Median	$Med(x) = \frac{1}{2}\left(\vec{x}_{\lfloor (K+1)/2 \rfloor} + \vec{x}_{\lceil (K+1)/2 \rceil}\right)$ where $\vec{x}$ denotes sorted x in ascending order; $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ denote the floor and ceiling functions, respectively	150
Maximum	$Max(x) = \vec{x}_K$	150
Minimum	$Min(x) = \vec{x}_1$	150
Interquartile range	$IQR(x) = Q_3(x) - Q_1(x)$, where $Q_3(x)$ and $Q_1(x)$ are the upper and lower quartiles of x	150
Standard Deviation	$s(x) = \sqrt{\frac{1}{K-1}\sum_{i=1}^{K}(x_i - \mu(x))^2}$	150
Mean absolute deviation	$D_{mean}(x) = \frac{1}{K}\sum_{i=1}^{K}\lvert x_i - \mu(x)\rvert$	150
Median absolute deviation	$D_{med}(x) = \frac{1}{K}\sum_{i=1}^{K}\lvert x_i - Med(x)\rvert$	150
Coefficient of variation	$C_V(x) = \frac{s(x)}{\mu(x)}$	150
Skewness	$\gamma(x) = \frac{\frac{1}{K}\sum_{i=1}^{K}(x_i - \mu(x))^3}{s(x)^3}$	150
Kurtosis	$\kappa(x) = \frac{\frac{1}{K}\sum_{i=1}^{K}(x_i - \mu(x))^4}{s(x)^4}$	150
Cross-correlation coefficient (lag 0)	$R_{xy} = \frac{\frac{1}{K}\sum_{i=1}^{K}[(x_i - \mu(x))(y_i - \mu(y))]}{s(x)s(y)}$, where x and y are two CSI amplitude variables of the same antenna or phase difference variables of the same antenna pair	$\frac{1}{2}(30 \times 29) \times 3 + \frac{1}{2}(30 \times 29) \times 2 = 2175$
Mean difference of consecutive subcarriers	$MDSC_{xy} = \mu(y) - \mu(x)$, where x and y are CSI amplitude or phase difference variables of two consecutive subcarriers of the same antenna	$29 \times 3 + 29 \times 2 = 145$

Table 2.1 shows that there are 3,970 feature candidates for each sample, which is a rather large feature pool considering that we only have 200 samples. Therefore, a feature selection is desired

before modeling to avoid over-fitting. There are many feature selection methods available. In this work we employ principal component analysis (PCA) for its simplicity and easy visualization, which is detailed in the next section.

2.6. Feature engineering with PCA

The goal of feature selection is to find features that maximize between-class variance (i.e., the distinct difference for samples of different MC levels) while minimizing within-class variance (i.e., high similarity for samples of the same MC level). For simplicity and robustness of features, we compare features by types listed in Table 2.1. This is conducted via unsupervised learning of PCA on each feature type and project them onto low-dimensional principal component subspace (PCS). Each feature was normalized to zero mean unit variance across all 200 samples prior to PCA. It is worth noting that features are selected from potential candidates based on how well they minimize the within-class variance while maximizing the between-class variance through data exploration and visualization. The results are illustrated in Figure 2.12 and Figure 2.13, where the 87 CSI amplitude mean difference of consecutive subcarriers (MDCSs) of 70 samples were projected onto the first three principal component directions to obtain the three "score" plots (Figure 2.13). For comparison, the score plots of 150 CSI amplitude means of the same 70 samples were also generated (Figure 2.12). As can be seen from Figure 2.13, MDCSs show not only significant between-class differences (i.e., samples from different MC levels are far apart in one or multiple score plots) but also significant within-class similarities (i.e., samples from the same MC level but different shuffles form a compact cluster). In contrast, the mean of CSI amplitude is much more sensitive to woodchip packing, indicated by the wide scattering of samples from the same MC level but different shuffles. In addition, compared to CSI amplitude MDCS, the CSI amplitude mean is less sensitive to MC levels, indicated by the less separation of samples from different MC levels. As mean directly resembles raw data behavior, this is an indication of potentially poor performance for classification using raw data, which was verified in the previous section. Through this comparison of all feature types listed in Table 2.1, it was found that the MDCSs of CSI amplitude are the best feature candidates and therefore were selected as the final features for developing classification models. In this way, we reduce the feature space from 3,970 to 87. It is worth noting that further feature selection can be conducted to use MDCSs of selected subcarriers instead of all 30 subcarriers. It is also worth noting that classification performance is expected to improve if more systematic feature selection is conducted, such as combining features from different types. These will be our future work to further improve the technology. However, in this work, we try to strike a balance that leans more towards simplicity and robustness than numerical performance.

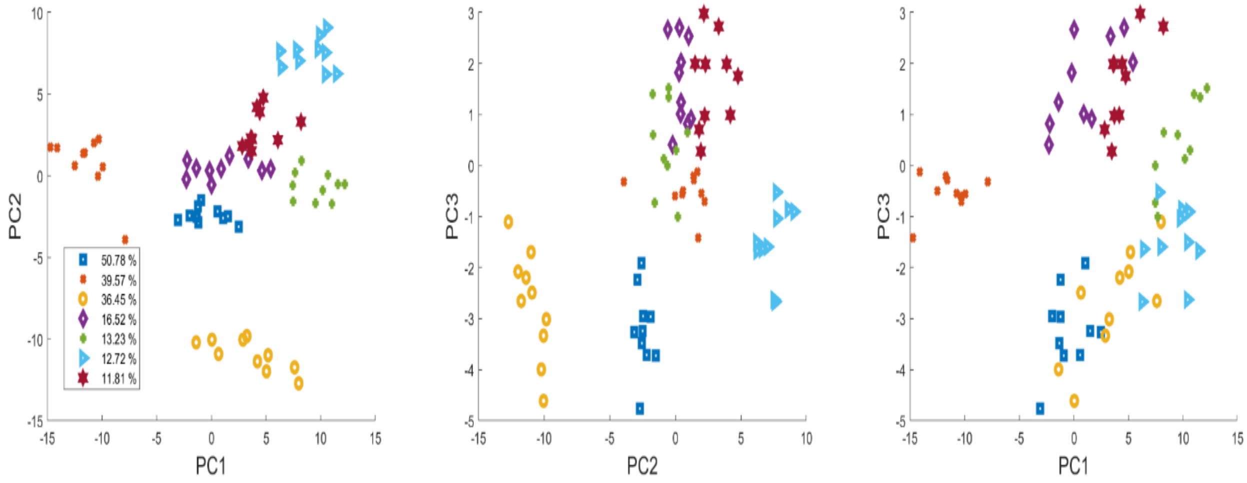


Figure 2.12 PCA score plots of CSI amplitude means of 70 samples at 7 different MC levels (i.e., 10 samples at each MC level)

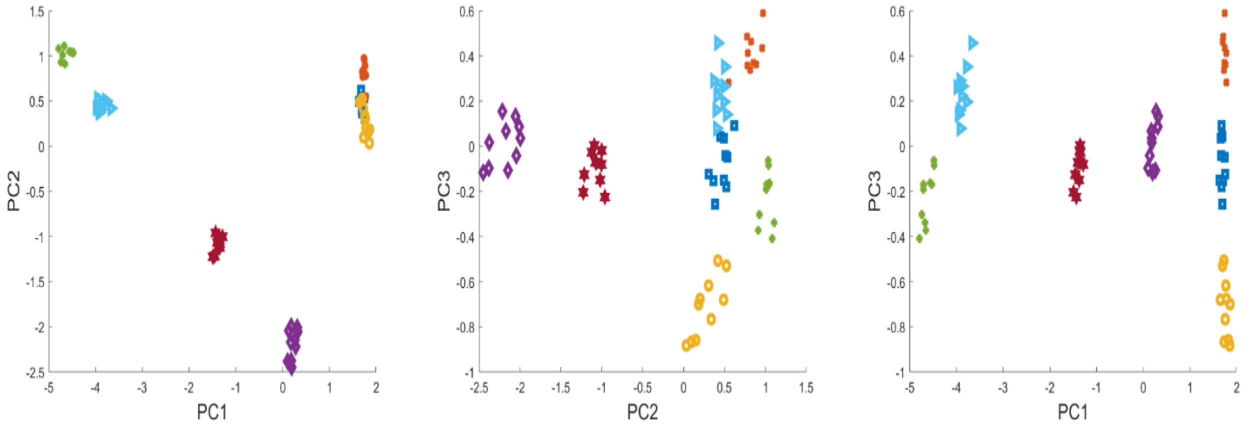


Figure 2.13 PCA score plots of CSI amplitude mean difference of consecutive subcarriers (MDCS) of the same 70 samples. MDCSs show much better quality as features in both maximizing between-class variance and minimizing within-class variance

2.7. Model building

Once the 87 CSI amplitude MDCSs are selected as the features, the next step is to develop classification models. In this work, we compare various state-of-the-art machine learning classification techniques in classifying woodchip MC levels using these features. The procedure is outlined in Figure 2.14. For each classification model, 9 samples are randomly selected from 10 shuffled samples at the same MC level for each of the 20 MC levels, which results in 180 training samples. The remaining one shuffled sample from each MC level is used for independent testing, which results in a total of 20 testing samples. Due to the limited number of samples, a Monte Carlo validation and testing (MCVT) procedure [51] is followed to repeat the random sample selection and model training and testing procedure 100 times. In addition to the mean and standard deviation of the overall classification accuracy (Eqn. 2.4) of 100 such MCVTs, the confusion matrix resulted from the same MCVTs is also used to evaluate the performance of different classification models.

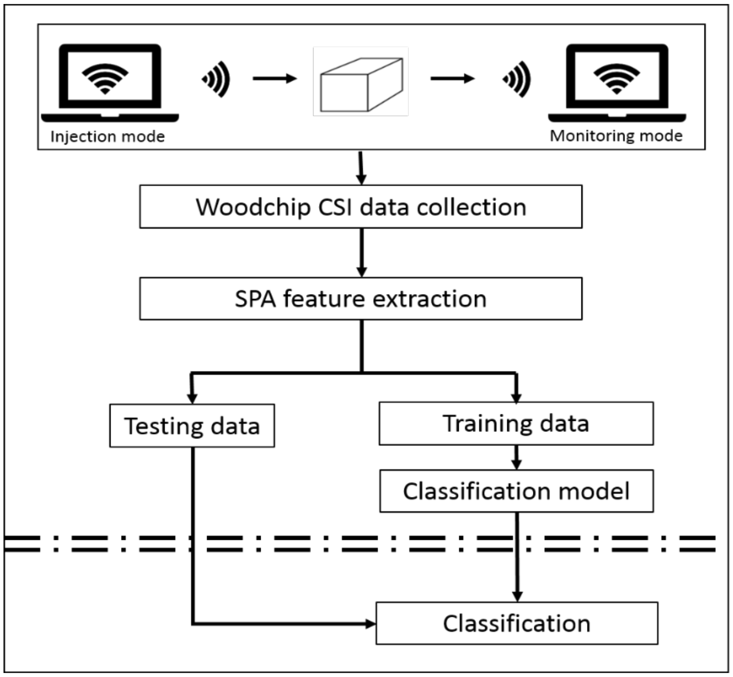


Figure 2.14 Overall process flow diagram of woodchip MC level classification using CSI data

The machine learning classification techniques studied in this work include linear discriminant analysis (LDA), support vector machine (SVM), artificial neural network (ANN), as well as ensemble modeling of bagging with LDA, and ensemble boosting method XGBoost. These methods are briefly reviewed in the following sections.

2.7.1. Linear discriminant analysis (LDA)

LDA is a robust supervised learning technique for multiclass classification. It is a generalization of Fisher's linear discriminant, which find a linear combination of features to separate multiples classes in the dimensional space. Scikit-learn Python library [53] is used to implement LDA in this work, which fits a Gaussian density to each class and estimates the class conditional distribution of data for each class k using Bayes' theorem:

$$P(y = k|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|y = k)P(y=k)}{P(x)} = \frac{P(\boldsymbol{x}|y=k)P(y=k)}{\sum_{l=1}^{C}\{P(\boldsymbol{x}|y = l)P(y=l)\}} \qquad (2.6)$$

where $\boldsymbol{x} \in R^d$ is a sample feature vector of dimension d, y is the class label of that sample, C is the total number of classes. LDA makes predictions by estimating the probability of a new sample belonging to each class. Based on the class with the highest probability, the new sample is assigned to that class. More information on multiclass LDA can be found in [54].

2.7.2. Support vector machine (SVM)

Support vector machine (SVM) is a supervised machine learning technique. In linear SVM classification of two classes, classification is performed by finding a hyperplane that maximizes the separation or margin between the two classes. If the two classes are not linearly separable, the input vectors can be nonlinearly mapped to a high-dimensional feature space through a kernel function that presumably makes the separation easier in the kernel space. In this application, it was found that linear SVM performs better than nonlinear kernels (e.g., radial basis function (RBF) and sigmoid kernels) based SVMs. This is consistent with the preliminary finding in the previous section, where a subset of 7 classes was shown to be linearly separable (*Figure 2.13*). More information on SVM can found in [55]–[57]. In this work, multiclass classification is carried out using scikit-learn [53] with the "one-versus-one" approach where 190 (i.e.,$(20 \times 19)/2$) classifiers are constructed.

2.7.3. Artificial neural network (ANN)

Artificial neural network (ANN), or simply neural network (NN), was developed with the idea of mimicking human brains, which now form the foundation of many deep learning techniques. A neural network consists of several layers, including an input layer that takes input data, one or more hidden layers depending on the complexity of the problem and the representations to be learned, and an output layer that outputs either discrete or continuous values depending on the type of problem, i.e., classification or regression. The constructed ANN represents interconnected input and out units or nodes (called neurons), in which each connection (called an edge) has an associated weight. The training of an ANN for classification is to adjust these weights to optimize the prediction of correct classes for the training data (e.g., through minimizing a cost function such as classification error). Once trained, the ANN takes a new set of similar data and makes class predictions based on the trained model. Keras is used for ANN implementation in this work. Because of the likely linear separability of this particular application, one hidden layer is used in this work. Other hyperparameters, including the number of neurons in the hidden layer, optimizer, activation function in the hidden layer, initialization, epochs, and batch size, are optimized using random search followed by Bayesian optimization. More information on ANN can be found in [58]–[61].

2.7.4. Bagging

Bagging is a bootstrap ensemble method that creates individual models for its ensemble by training each classifier on a random distribution of the training data. Each classifier's training set is generated by random sampling, with or without replacement from all the samples available for training. Individual predictions of each classifier are aggregated based on a voting scheme (hard voting or soft voting) to form a final prediction. Each base classifier can be trained in parallel with the subsamples generated with random sampling. Bagging is known to reduce overfitting or high variance by voting. Different base estimators can be used within bagging. In this work, LDA classifier is used due to the linear separability of the classes. Scikit-learn is used to implement bagging. The hyperparameters, including the number of base classifiers, bootstrapping samples and/or features, and the sample/feature size, are optimized using random search followed by Bayesian optimization. More information on bagging can be found in [62]–[65].

2.7.5. XGBoost

Another ensemble method that constructs multiple regression trees is boosting. In comparison to bagging, boosting approaches combine weak learners into strong learners iteratively by optimizing a cost function along the negative gradient direction. XGBoost is one of the most successful boosting approaches under the gradient boosting framework. The XGBoost algorithm objective combines training loss and regularization terms for a trade-off on bias and variance. Python library xgboost is used for implementation. The hyperparameters, include the number of base learners (i.e., regression trees), learning rate, updater, feature selector, and regularization parameters, are optimized using random search followed by Bayesian optimization. More information on XGBoost can be found in [66].

2.7.6. Hyperparameter optimization

Hyperparameter optimization is very important in training ML models as the model architecture directly affects the model performance. There are three major approaches for hyperparameter optimization, including grid search, random search [67], [68], and Bayesian optimization [69], [70]. Grid search can be quite effective when dealing with a small hyperparameter space. In general, however, random search and Bayesian optimization are more efficient than grid search. For complex models with large parameter spaces, such as XGBoost and ANN, the time required for grid search could be prohibitive. In these cases, random search or Bayesian optimization is preferred. Random search samples random parameter combinations based on certain statistical distributions. The idea is that, provided enough iteration, random search can find an optimum or close to optimum in lesser time than grid search, although random search does not guarantee a global optimum. Both grid search and random search find optimal hyperparameters in an isolated way without considering past evaluations. In contrast, Bayesian optimization considers past hyperparameter values that minimize the cost function by building a surrogate model based on past evaluation results. The surrogate model is presumably computationally cheaper to optimize than the original objective function, so the next input values are selected by applying criteria, such as expected improvement (EI), to the surrogate model. In this work, random search is utilized to explore the hyperparameter space. The final hyperparameters are determined by Bayesian optimization with Tree Parzen Estimator (TPE) using EI as the criterion. The Scikit-learn library is used for random search, while hyperopt [69] is used for Bayesian optimization.

2.8. Results and discussion

In this section, we discuss our findings of woodchip MC level classification using the 87 features extracted following the SPA framework. The classification results from the five different classification methods discussed in the previous section are compared. As discussed previously, due to the limited number of samples, 100 MCVT simulations are conducted. For every classification technique in each MCVT simulation, hyperparameters are optimized using stratified 10-fold cross-validation. The trained model is used for evaluation on the set-aside testing set. The average and standard deviation of classification accuracy of 100 such runs (100 different test sets) are used to evaluate the performance of each classification method. In addition, the overall classification confusion matrix from 100 MCVTs is used to visualize and detect the far-off misclassifications where the predicted class is off its true class by more than one MC level.

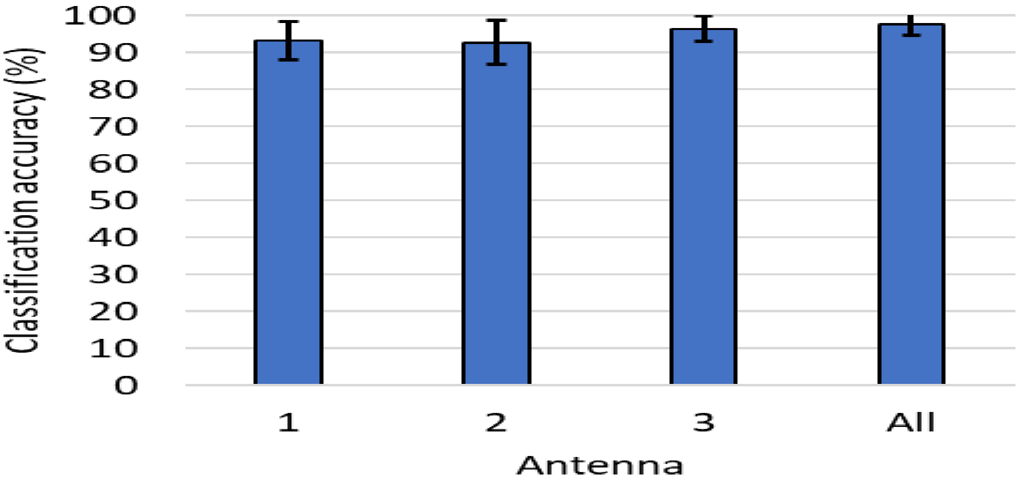
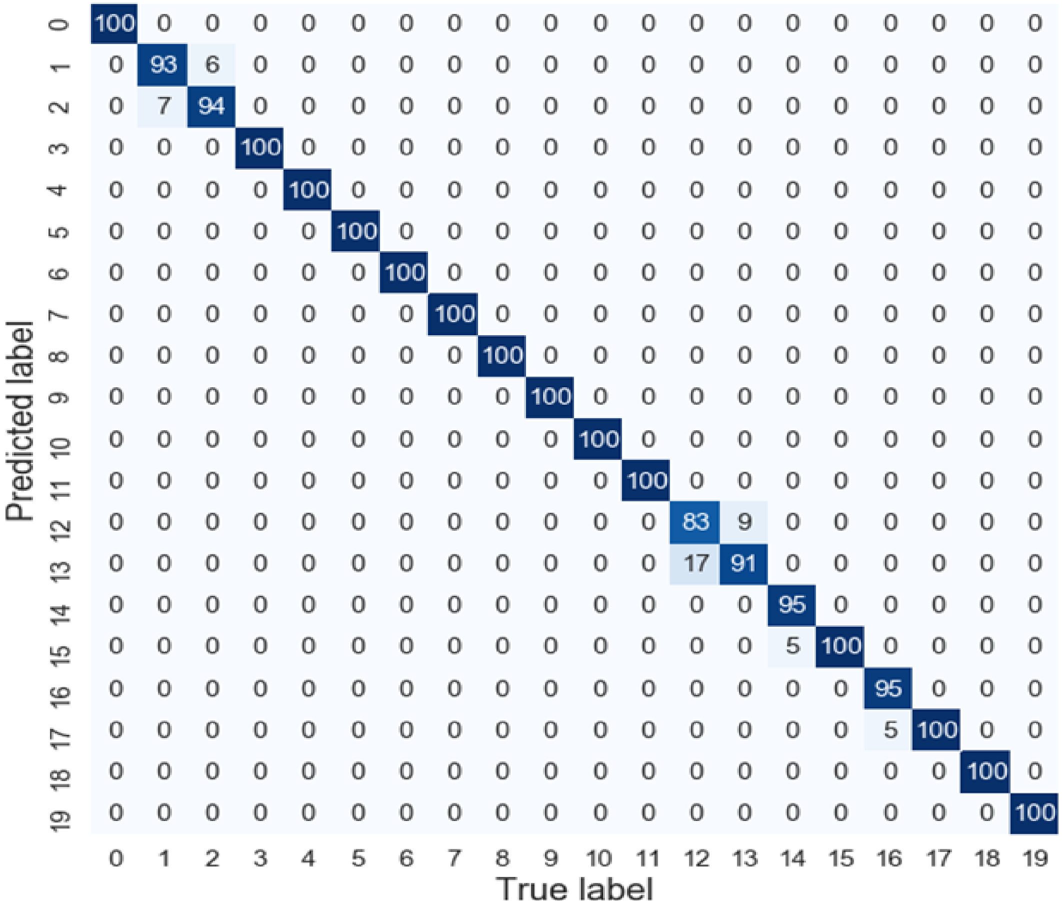


Figure 2.15 Comparison of classification accuracy of LDA when features from different antennas are used

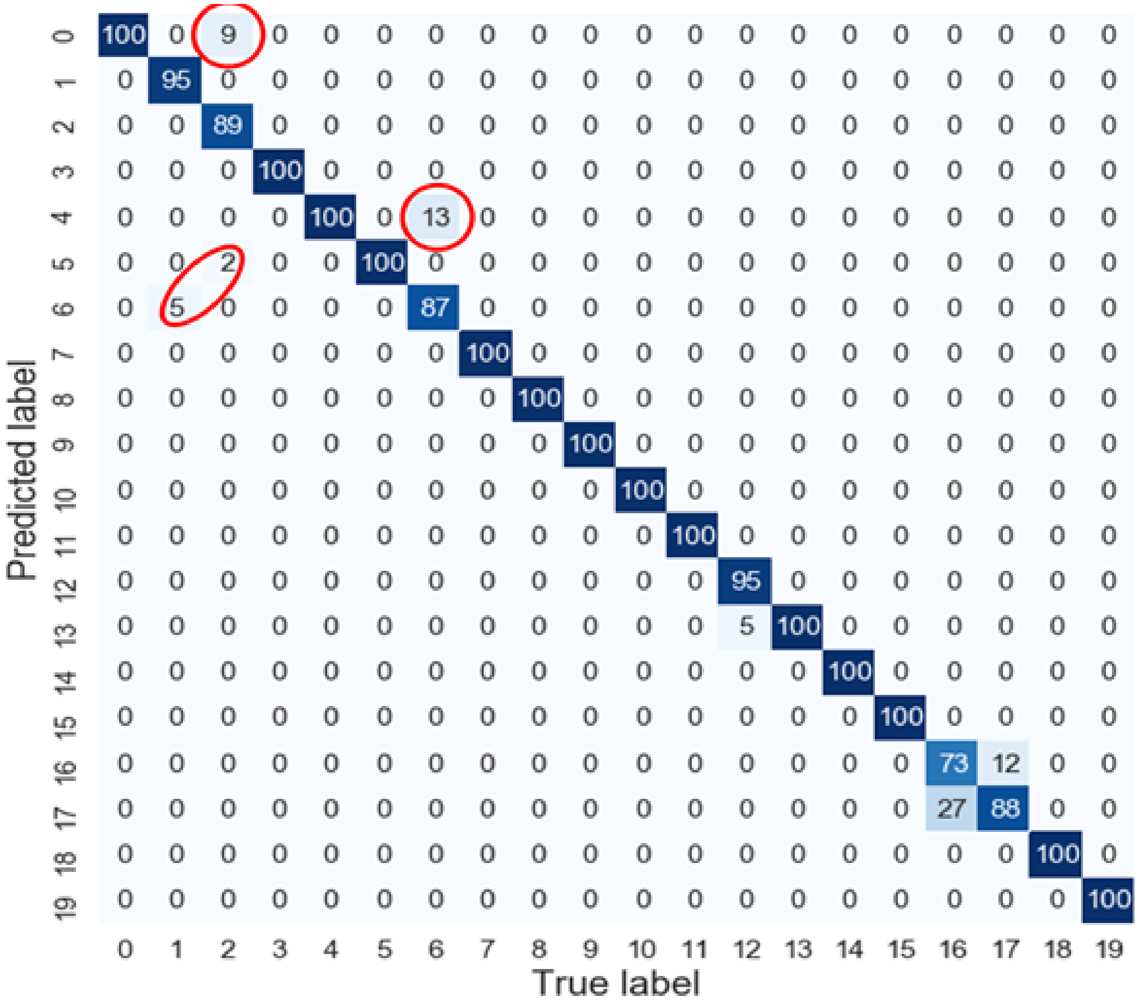
Table 2.2 Overall classification accuracy of LDA when features from single or all antennas are used

Data used	Classification accuracy
Antenna 1	93.05 ± 5.17
Antenna 2	92.6 ± 5.97
Antenna 3	96.35 ± 3.40
All	97.55 ± 2.89



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	93	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	7	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	83	9	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	17	91	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	100	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	100	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 2.16 Comparison of classifcation confusion matrices when all features from all three antennas are used



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	100	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	95	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	89	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	100	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	2	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	5	0	0	0	0	87	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	95	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	5	100	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	73	12	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27	88	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 2.17 Comparison of classification confusion matrices when only features from antenna 3 are used

We first investigate effect of antennas using LDA. The mean and standard deviation of overall classification accuracy is shown in Table 2.2 and Figure 2.15. It can be seen that when a single antenna (i.e., with 29 out of 87 features) is used, antenna 3 provides the best information for classification. The best results, in both mean and standard deviation of classification accuracy, are obtained when all three antennas (i.e., with all 87 features) are used.

Another advantage of using all three antennas is observed when comparing the confusion matrix of different scenarios. Figure 2.16 and Figure 2.17 compares the confusion matrices of using only antenna 3 with that of using all three antennas. It can be seen that there are 29 far-off misclassifications when only antenna 3 is used. When all three antennas are used, there is no far-off misclassification occur. Therefore, for the remainder of this work, all 87 features from all three antennas are used.

Next, using all 87 features from all three antennas, we compare performance of different classification methods. The results are shown in Figure 2.18 and Table 2.3, which indicate that all methods perform well and all achieve greater than 95% overall classification accuracy.

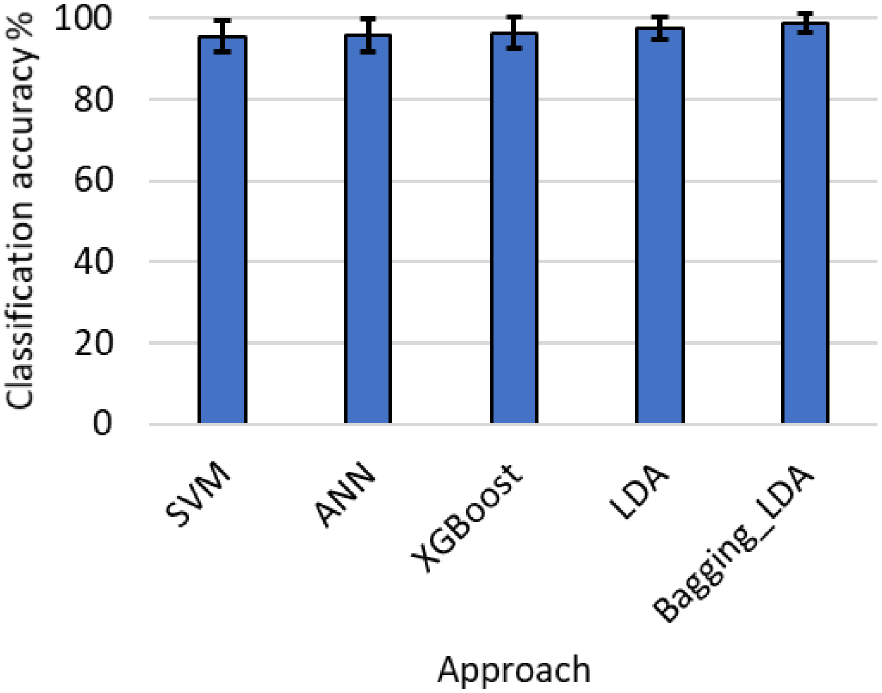
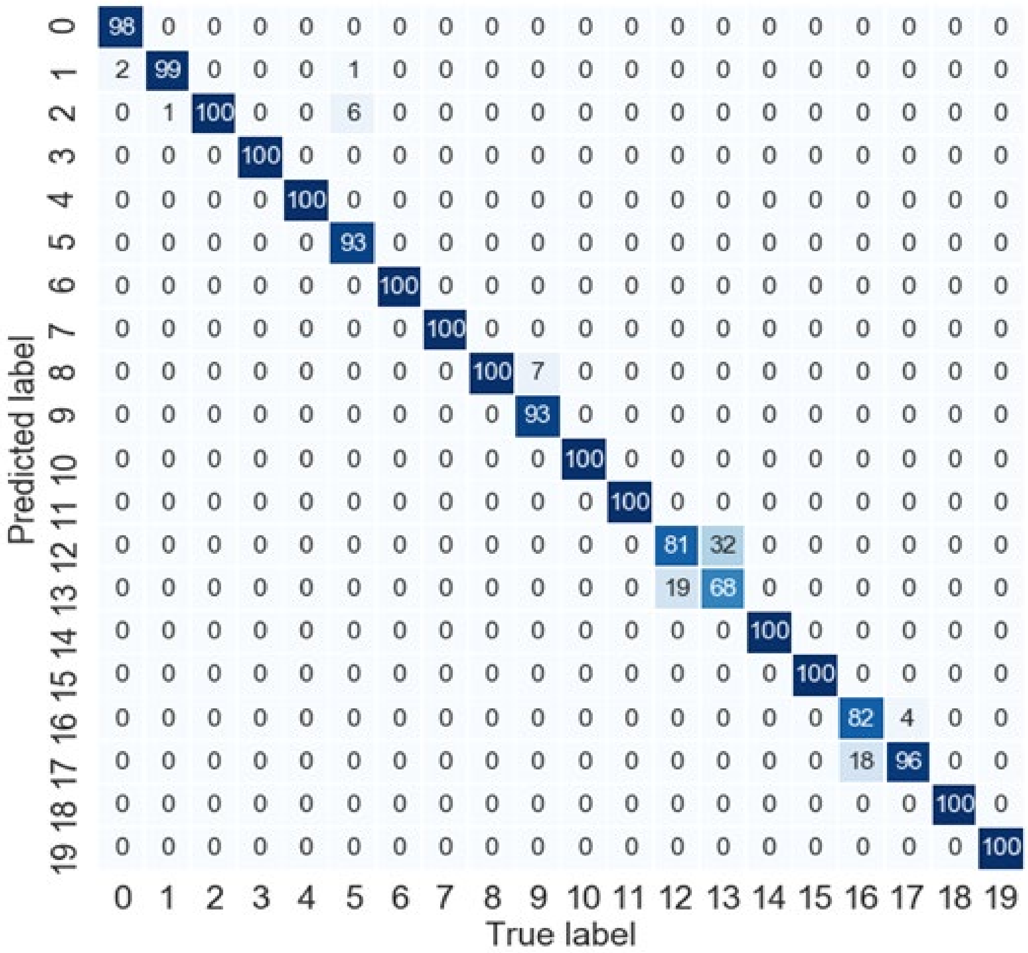


Figure 2.18 Comparison of overall classification accuracy when different classification techniques are used

Table 2.3 Overall classification accuracy when different classification techniques are used

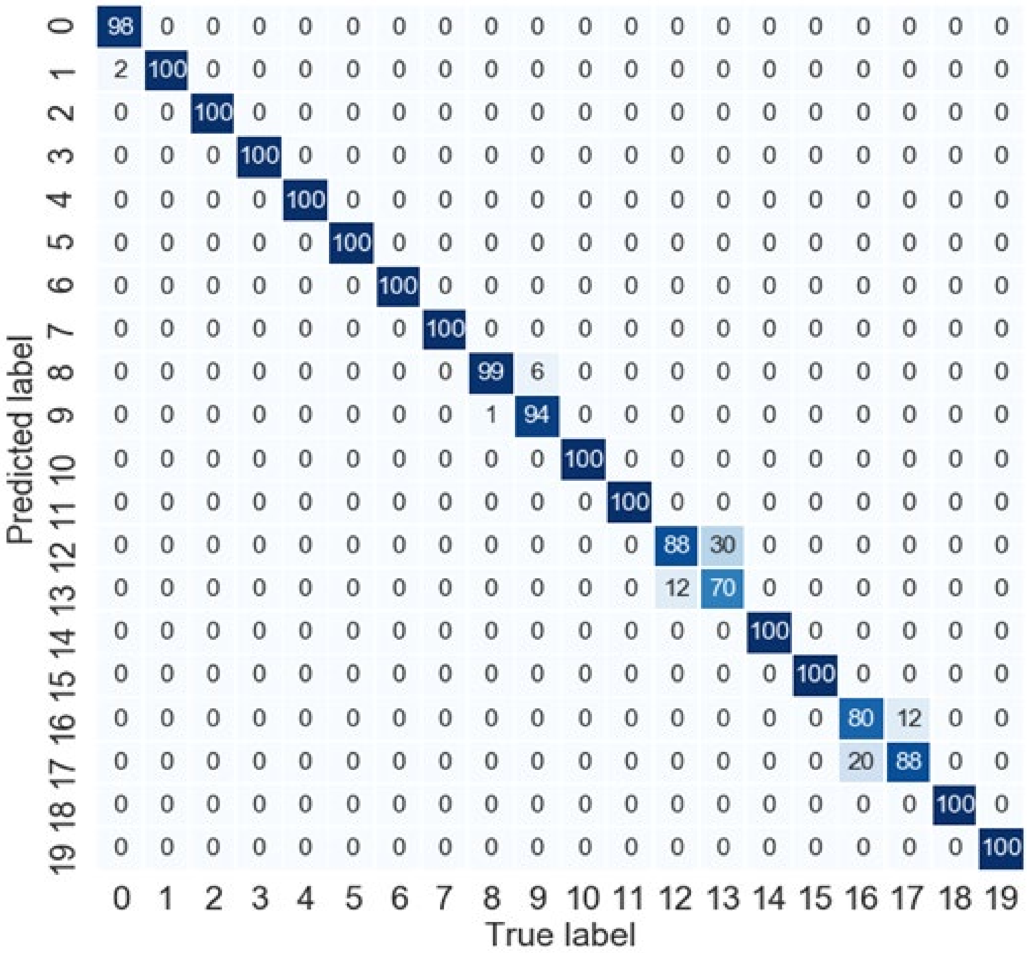
Method	Classification accuracy
SVM	95.50 ± 3.79
ANN	95.85 ± 4.15
XGBoost	96.40 ± 3.70
LDA	97.55 ± 2.89
Bagging (LDA)	98.75 ± 2.29

SVM performs the worst among all methods in terms of mean classification accuracy. ANN performs slightly better than SVM in mean classification accuracy but with a slightly higher standard deviation, indicating lower consistency when different training and testing samples are used. However, an analysis into the confusion matrices shows that SVM results in seven far-off misclassifications while all other methods result in zero far-off misclassification (Figure 2.19, Figure 2.20,Figure 2.21 and Figure 2.22). XGBoost performs slightly better than ANN and SVM but not as good as LDA. This result is somewhat surprising as XGBoost has outperformed other techniques in many Kaggle competitions on real-world datasets and a variety of applications. However, as shown earlier in Figure 2.13, this application is more of a linearly separable case with the features selected, which explains the good performance of LDA. The results also demonstrate the robustness of LDA when dealing with linearly separable cases. Nevertheless, bagging can still improve a base classifier such as LDA in this work. As shown in Table 2.3 Overall classification accuracy when different classification techniques are used, bagging of LDA provides the best performance with the highest average overall classification accuracy of 98.75% and the smallest standard deviation of 2.29% from 100 MCVT's. The confusion matrices of all methods indicate that only SVM results in far-off misclassifications while all other methods only result in nearest-neighbor misclassification. The specific number of the two types of misclassifications are compared in Figure 2.23, where the LDA on raw CSI amplitude data is used as the reference. Figure 2.23 shows that feature engineering and selection play a key role in this application, and all methods based on the 87 CSI amplitude MDCS features easily outperform LDA with raw CSI amplitude data as features.



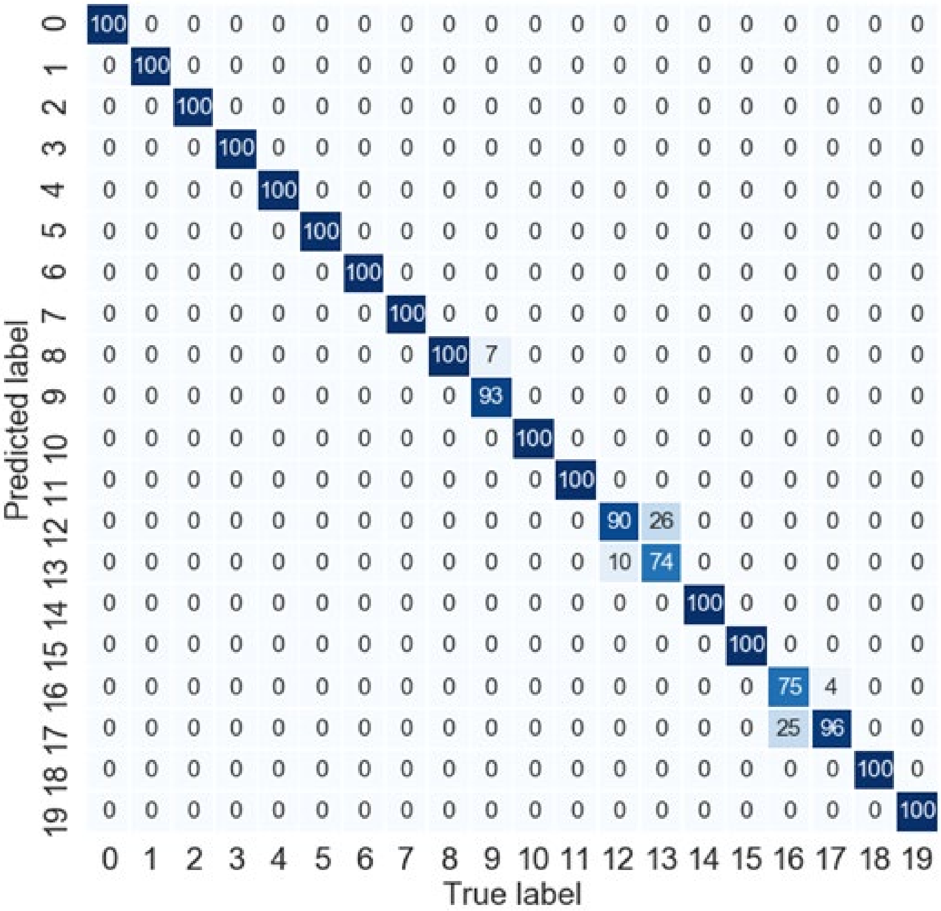
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	99	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	100	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	93	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	100	7	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	93	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	81	32	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	19	68	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82	4	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	96	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 2.19 Classification confusion matrix - SVM



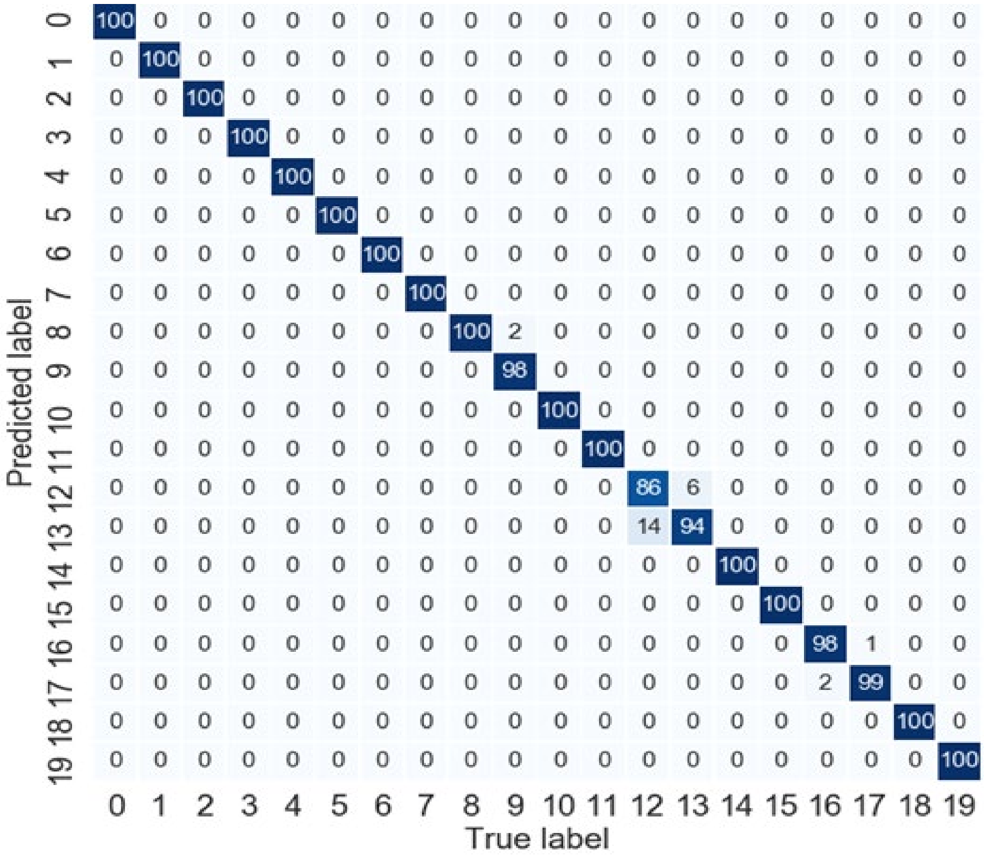
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	99	6	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	94	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	88	30	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	12	70	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	12	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	88	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 2.20 Classification confusion matrix - ANN



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	100	7	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	93	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	90	26	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	10	74	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	4	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	96	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 2.21 Classification confusion matrix for XGBoost



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	100	2	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	98	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	86	6	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	14	94	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	98	1	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	99	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Figure 2.22 Classification confusion matrix for Bagging with LDA as base estimator

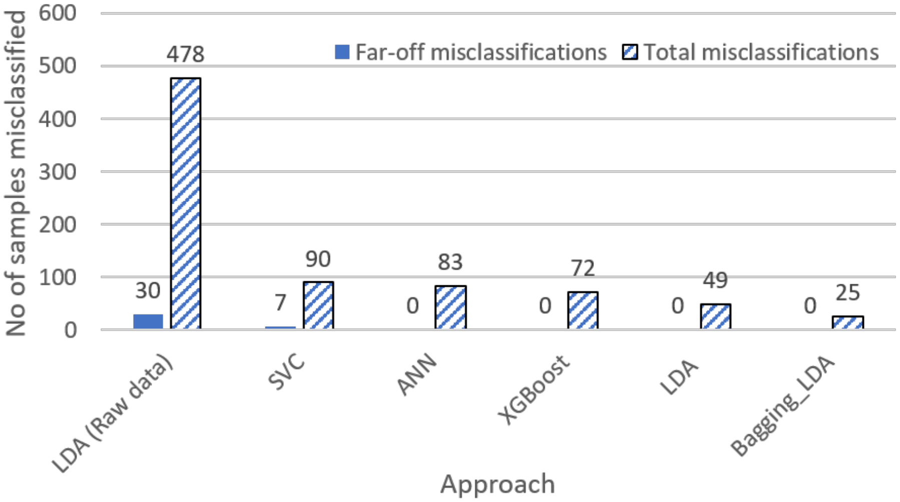


Figure 2.23 Comparison of far-off misclassification of different approaches

We also compared the following two scenarios of hyperparameter tuning:

A) A set of hyperparameters are optimized for each MCVT run using the selected training samples, and that set of hyperparameters are used for evaluation on the corresponding test set. Therefore, different MCVT runs could potentially have different hyperparameter values.

B) The optimal hyperparameters from 100 MCVT's of Scenario A are stored, and the mode of each hyperparameter (i.e., the value that appears most frequently) is selected to construct a universal set of hyperparameters. The universal hyperparameter set is used for model training and testing of the same 100 sets of training and testing samples as in Scenario A.

One potential issue with Scenario B is that a test sample in one MCVT is potentially used as a training sample in other MCVT's. When the hyperparameters from all MCVT's are pooled together to determine the mode, essentially all samples have been used as training samples for hyperparameter tuning, and there are no independent samples left for testing. This is confirmed by the comparison of the classification accuracy of the two scenarios. As shown in Table 2.4, except LDA, all other methods tuned following Scenario B slightly outperform their counterparts tuned following Scenario A. Therefore, the results reported previously in this work are all based on Scenario A for fair evaluation of all methods with independent test samples.

Table 2.4 Comparison of classification accuracy under two hyperparameter tuning scenarios

Method	Scenario A	Scenario B
SVM	95.50 ± 3.79	96.40 ± 3.63
ANN	95.85 ± 4.15	96.35 ± 3.61
XGBoost	96.40 ± 3.70	96.40 ± 3.34
LDA	97.55 ± 2.89	97.55 ± 2.89
Bagging (LDA)	98.75 ± 2.29	99.35 ± 1.69

2.9. Conclusion

In this section, for the pulp and paper industry in the U.S., the pulping process has been identified as a major opportunity for improving energy efficiency and productivity. However, the implementation of model-based optimization, control, and other advanced manufacturing technologies has been hindered by the lack of real-time sensing of woodchip MC under the harsh manufacturing environment. To overcome this bottleneck, we investigate the potential of an IIoT short-range Wi-Fi-based woodchip MC sensing technology. The proposed technology takes the advantages of IIoT devices (e.g., toughness, connectivity, low-cost, small-size, etc.) while overcoming their shortcomings (e.g., the machine learning challenges of messy big data) by SPA-based feature engineering. Specifically, this work demonstrates that woodchip packing is a strong confounding factor to woodchip MC level, evidenced by its significant impact on both amplitude and phase of the collected CSI data. Although randomization is a good strategy to mitigate this confounding factor, it is not sufficient by itself. As a validation, we demonstrated that classification using raw CSI data results in not only low classification accuracy but also many far-off misclassifications where the predicted MC class is off its true class by more than one level. The result also illustrates that classification accuracy alone is not a good performance metric, and the practical implications (e.g., cost) of misclassification must also be considered. We show that the SPA-based feature engineering framework is a systematic approach for generating physically and statistically meaningful features compared to other kernel-type or algorithmically generated (*e.g.*, square, square root, exponential, etc.) features that are often unintuitive. Through simple feature selection such as PCA, the mean difference of consecutive subcarriers (MDCSs) of CSI amplitude were found to be robust features that are not only highly sensitive to MC levels but also highly insensitive to woodchip packing. Using MDCSs as features, we demonstrated the superior classification performance of using CSI data collected off all three antennas compared to that of using any single antenna. Finally, using MDCSs from all three antennas, we investigate the representative state-of-the-art classification techniques, including LDA, SVM, ANN and ensemble learning methods including bagging with LDA and gradient boosting with XGBoost. The results showed that LDA and its bagging extension perform the best among all methods, achieving overall classification accuracy of 98~99%. In addition, when MDCSs are used as features, only SVM results in far-off misclassifications, while all other methods only result in nearest-neighbor misclassifications, which is a significant improvement compared to when raw CSI data were used as features.

3. Chapter 3. Next-generation virtual metrology for semiconductor manufacturing: A feature-based framework

In semiconductor manufacturing, VM is the prediction of wafer properties using process variables and other information available for the process and/or the product without physically conducting property measurement. VM has been utilized in semiconductor manufacturing for process monitoring and control for the last decades. In this work, we demonstrate the shortcomings of some of the commonly used VM methods and propose a feature-based VM (FVM) framework. Unlike existing VM approaches where the original process variables are correlated to metrology measurements, FVM correlates batch features to metrology measurements. We argue that batch features can better capture semiconductor batch process characteristics and dynamic behaviors. As a result, they can be used to build better predictive models for predicting metrology measurements. FVM naturally addresses some common challenges that cannot be readily handled by existing VM approaches, such as unequal batch lengths and/or unsynchronized batch trajectories. Simulated and industrial case studies are used to demonstrate the effectiveness of the proposed FVM method. We discuss how to generate and select features systematically and demonstrate how feature selection affects FVM performance using a case study. Finally, the capabilities of FVM in addressing process nonlinearity are investigated in great detail for the first time, which helps establish the theoretical foundations of the proposed framework for the semiconductor industry.

3.1. Introduction

In semiconductor manufacturing, a wafer undergoes hundreds of different steps to yield the final product. After a processing step, typically, a few (1–3) wafers within a lot are measured at the metrology station, and this sampled metrology data represent the whole lot. Tools such as ellipsometer and Atomic Force Microscope (AFM) as shown in Figure 3.1 are used for offline metrology to ensure the quality of the product manufactured is on par with the standards and the process is on target. However, this methodology using the traditional offline metrology tools becomes insufficient when the device dimensions continue to decrease and the lot-to-lot process control is being increasingly replaced with the wafer-to-wafer (W2W) control. In addition, there has been a tremendous increase in the throughput due to the rising demands for products. Performing offline metrology after each step on the wafers to ensure the quality leads to a very high cost as well as a significant time delay. W2W control requires metrology measurements of every wafer, and it has been proposed to use the integrated metrology (IM) sensors at the processing tool to provide such measurements[71]. However, issues such as the impact on throughput, increase in cycle time, and higher cost make IM less attractive in many process environments.

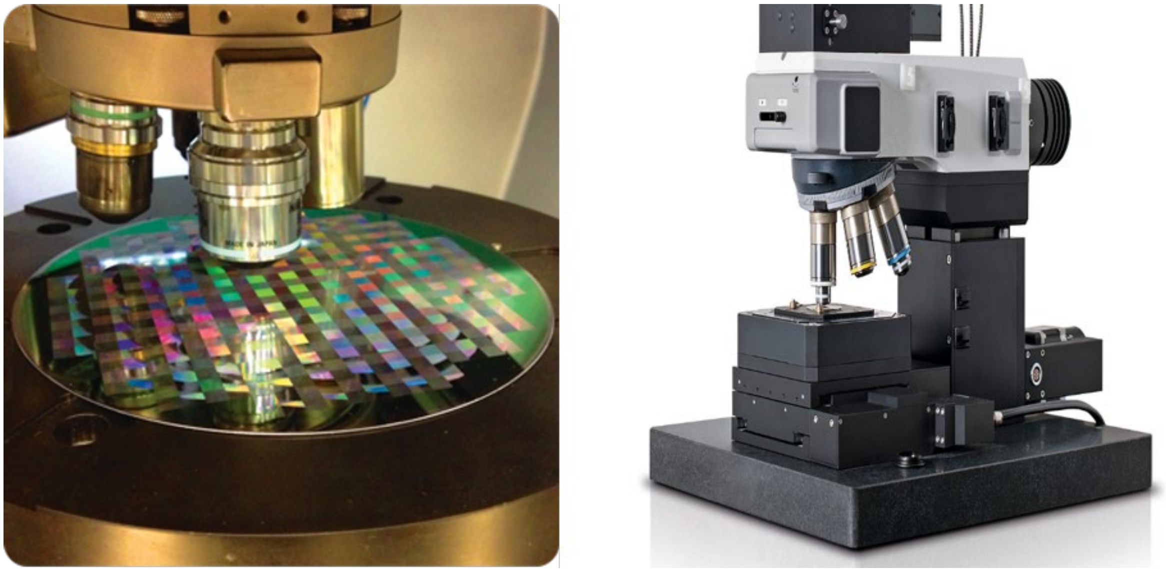


Figure 3.1 Atomic Force Microscope

As a solution to the existing approach, Virtual Metrology (VM) technology (also known as the soft sensor in process industry) has been proposed for 100% wafer measurement to support W2W control[72][49][73]. Frequent sampling is the key for better control to the manufacturing processes. Because machine data are usually sampled much more frequently compared to metrology data, and machine data are instantly available compared to delays often associated with metrology tools, an accurate VM can significantly improve process monitoring and control performance by providing real-time predicted metrology data.

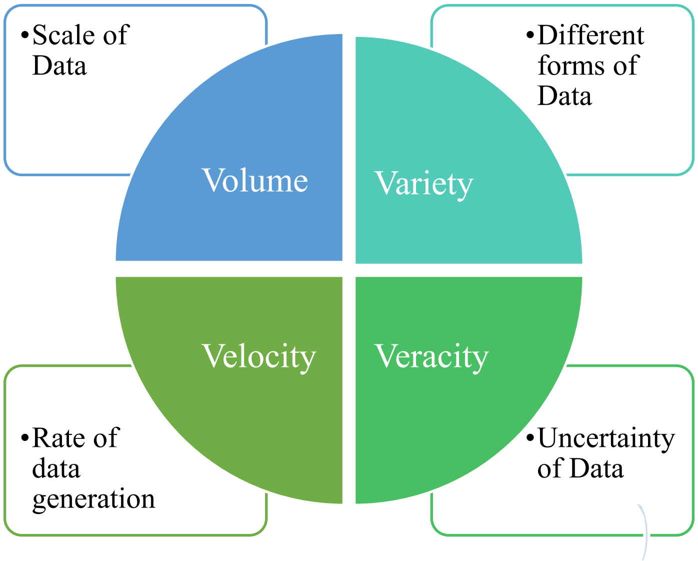


Figure 3.2 Current challenges in Big data

In addition, in today’s industrial scenario, the rate at which information is available, it is impossible to process information and extract useful findings with traditional tools available. This is one of the major focus areas for the next round of transformation in advanced manufacturing. Big data analytics opens up new horizons for managing significantly larger amounts of information. Big data has its own challenges and its characteristics can be summarized by 4 V’s: Variety (different types of data), Volume (systems needs to be able to handle the massive amount of data in real-time), Velocity (the speed at which data is generated) and Veracity (trustworthiness of data in terms of accuracy)[74], as shown in Figure 3.2. Acting on it with analytics for improved diagnostics and prognostics would lead to significant advancements in the field. The goal is to be proactive instead of being reactive, regardless of the volume of data.

3.2. Virtual Metrology (VM)

VM is the process of predicting the product properties based on the relationship between the metrology data of quality variable and process data without physical measurements. The main objective of Virtual Metrology is to achieve total quality management and run-to-run control. The major advantages of Virtual metrology include reduced cost, reduced production time due to no delays pertaining to metrology measurement, predictive maintenance, and detection of faults at faster rates. Figure 3.3 describes the schematic of a typical virtual metrology model[75]. Large volumes of high-frequency machine data and metrology measurements obtained through metrology equipment available are used to build models with very high prediction accuracy. These models, in turn, are used for an online application where newly available machine data for wafers produced is used to predict their product properties thereby, ensuring the process is on target. It is also worth noting that, in semiconductor industry, the use of big data analytics tools is not only limited to prediction of product properties but can also be used fault detection and diagnosis, and process control as well as shown in Figure 3.4. However, this work investigates the use of big data tools in predicting product properties.

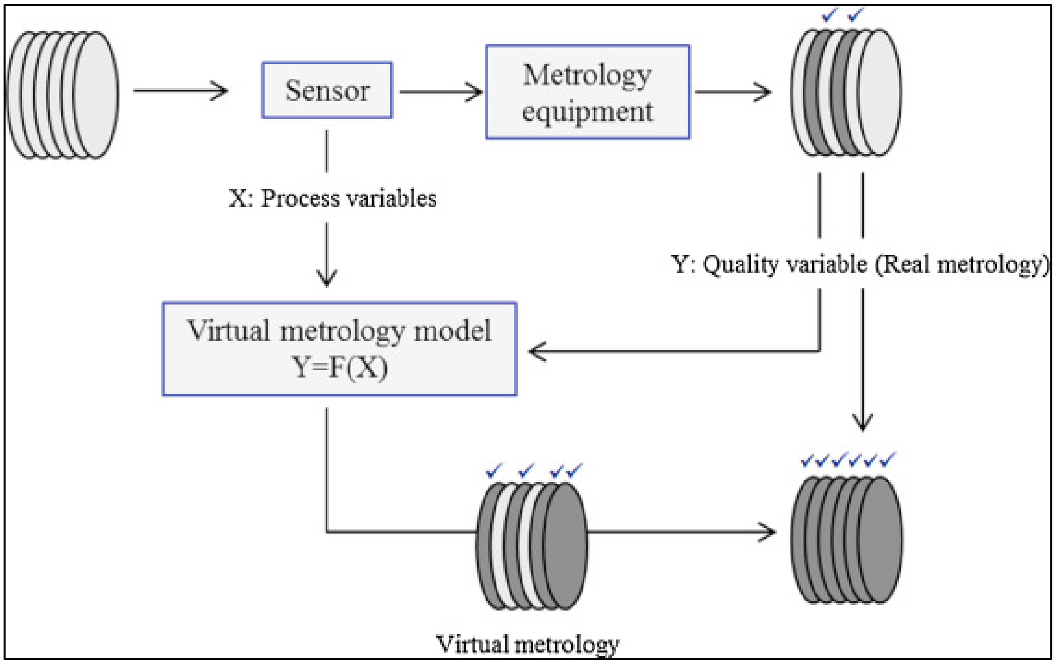


Figure 3.3 Schematic of VM modeling

3.3. Challenges

One of the most important factors that need to be considered when implementing any VM for industrial applications is the level of data pre-processing required. Data pre-processing has a direct and significant impact on the deployment and maintenance of the VM. Fewer data pre-processing steps and/or more automated data pre-processing steps lead to a more sustainable method in a production environment.

Another important factor is the prediction accuracy of the VM approach used in predicting the properties of interest of a wafer. Semiconductor processes often need to eliminate or reduce the effect of process noise, measurement noise, and unexpected drifts in the process. These problems often degrade the quality of control as well as the quality of a product. Hence, these are the main barriers that hamper the development of an accurate VM model. Dealing with these challenges would lead to a better and accurate VM system for semiconductor manufacturing. Through our work, we aim to get a step closer towards an ideal VM model by addressing both the existing challenges mentioned above.

3.4. Research objectives

To address the existing challenges in semiconductor manufacturing, we propose a feature-based VM (FVM) framework based on the SPA process modeling and monitoring framework we proposed previously[38], [47]. The most significant difference between the proposed VM approach and other existing approaches is that instead of extracting correlations between process variables and metrology measurements, the proposed method extracts the correlations between process features and metrology measurements to build VM models. By doing so, the proposed method can not only eliminate most data pre-processing steps but also provide superior prediction performance. SPA for a virtual metrology framework has been tested previously[49]. However, in that work, the features were limited to process variable statistics, and the mechanisms behind SPA were not explored. One major contribution of the present work is to extend and generalize the method to include any features, not just statics, but also non-statistical process features, such as process knowledge-based landmark features[76]; profile-driven features[77]; geometry-based features[78]. A detailed study on how the features are generated, selected systematically, and how feature selection affects FVM performance is presented. Finally, the capabilities of FVM in addressing process nonlinearity are investigated in great detail for the first time, which helps establish the theoretical foundations of the proposed framework for the semiconductor industry.

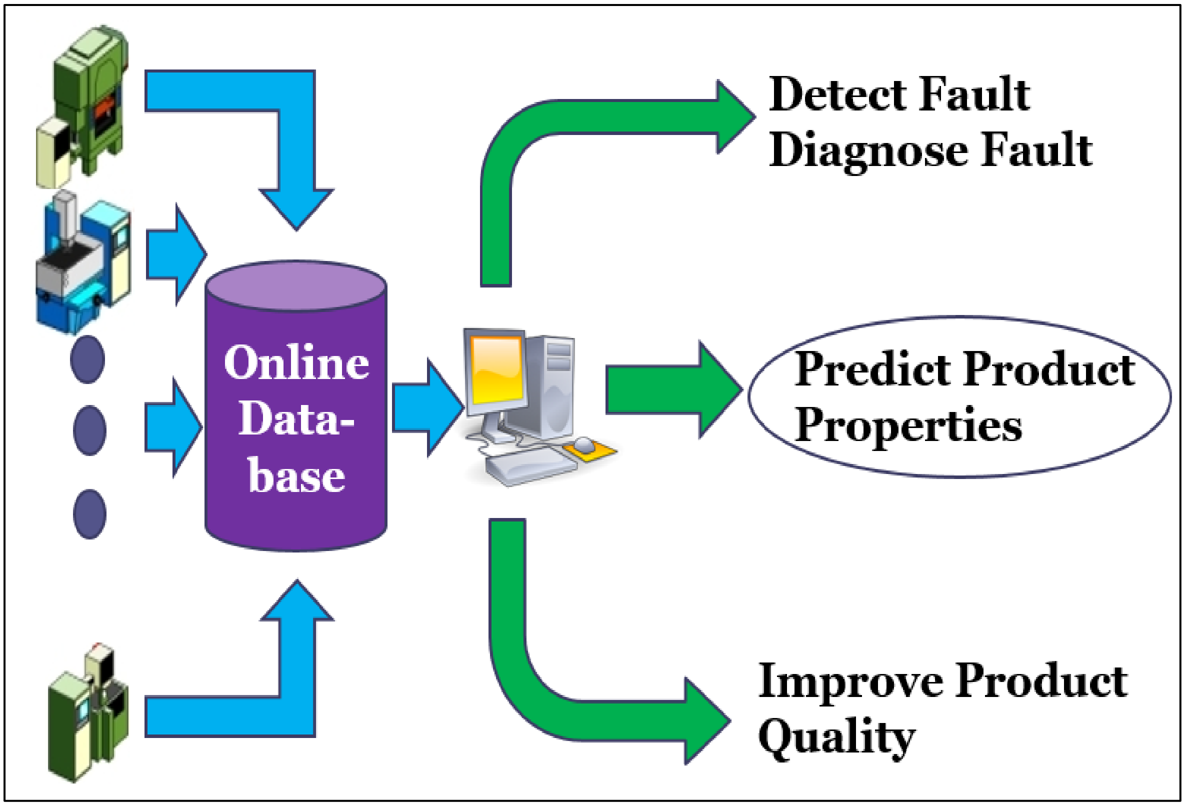


Figure 3.4 Application of Big data analytics to the Semiconductor Industry

3.5. A brief review of existing VM approaches

As discussed previously, VM is not unique to the semiconductor industry, which essentially serves the same purposes as the soft sensor, a term often used in the process industry. VM or soft sensor makes use of secondary variables that are measured online frequently to predict the product quality variables that are not measured online or measured infrequently. VM can be developed using either model-based approaches or data-driven approaches. For industrial processes, data-driven approaches are usually easier to develop and to implement online; therefore they are potentially more attractive. Due to the limited space, only some of the data-driven VM approaches applied to semiconductor manufacturing processes are reviewed in this work.

Among data-driven approaches, the commonly used ones are time series analysis (TSA), Kalman filter (KF), multiple linear regression (MLR), principal component regression (PCR), partial least squares (PLS), and other nonlinear methods such as those based on artificial neural networks (ANNs).

3.5.1. Time series analysis (TSA)

Because the metrology data are generally sequential in time, autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) models can be identified, e.g., following the procedure proposed by [79]. Once the model structure is determined, and parameters are estimated using the historical metrology data, the model can be used to predict the future values of the metrology data. Non-seasonal ARMA models are usually denoted by ARMA(p,q) in the following form that combines AR and MA models.

$$y_t - \alpha_1 y_t - 1 - \cdots - \alpha_p y_{t-p} = \in_t + \theta_1 \in_{t-1} + \cdots + \theta_q \in_{t-q} \tag{3.1}$$

where yt, $y_{t-1} \cdots y_{t-q}$ are present (at time t) and past metrology data. Parameters p and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, and q is the order of the moving-average model. ARMA model assumes that the time series is stationary and it is recommended to difference non- stationary series one or more times to achieve stationary, which results in a more general ARIMA(p, d, q) model where d is the degree/time of differentiation.

3.5.2. Kalman filter (KF)

Kalman filter was proposed in the early 1960s and has been extensively used for the state estimation of dynamic systems[80][81]. It has also been formulated for VM[72].

$$K = P_{dd}C^T(CP_{old}C^T + R)^{-1} \tag{3.2}$$

$$x_{new} = x_{old} + K(y - Cx_{old}) \tag{3.3}$$

$$P_{new} = P_{old} - K(CP_{old}) \tag{3.4}$$

$$y_{est} = Cx_{new} \tag{3.5}$$

Where **K** is the Kalman gain, **P** the state error covariance matrix, **R** the measurement noise covariance matrix, **x** the independent or process variables, the dependent or metrology variable.

3.5.3. Multiple linear regression (MLR)

Multiple linear regression (MLR) aims to model the relationship between multiple explanatory or independent variables from machine data and a response or dependent variable of metrology data by fitting a linear equation to the historical data, which takes the following form:

$$y = Xb + \in \tag{3.6}$$

where $\mathbf{X} \in \Re^{\boldsymbol{N}\times}$ is the independent variable matrix; $\mathbf{y} \in \Re^{\boldsymbol{N}\times 1}$ is a vector of metrology measurements and $\in$ is the random error or residual. The coefficient vector **b** is estimated by minimizing the sum of squares of the differences between the actual and modeled metrology measurements, and the obtained model is used to predict metrology measurement when a new set of process variables are available. In our work, traditional batch wise unfolding is used to convert the 3-dimensional matrix into a 2-dimensional matrix. The potential issue with MLR for VM is that the process variables are quite often (highly) correlated, and the collinearities among **x_i** can cause severe problems for MLR - the estimated coefficients $\hat{\boldsymbol{b}}$ Can be very unstable, which makes predictions by the regression model unstable or poor.

3.5.4. Principal component regression (PCR)

Principal component regression (PCR) is an alternative to MLR, which addresses independent variable collinearities. PCR is a regression analysis technique based on principal component analysis (PCA)[82], [83][84]. In PCR, the matrix of raw data $\mathbf{X} \in \Re^{N \times K}$ is decomposed as follows

$$X = TP^T + \tilde{X} \qquad (3.7)$$

Where $\mathbf{T} \in \Re^{N \times L}$ and $\mathbf{P} \in \Re^{K \times L}$ are the score and loading matrices, respectively. $\widetilde{\boldsymbol{X}}$ is the residual matrix containing mainly the noise. Then **y** is related to **T:**

$$y = Tb \qquad (3.8)$$

which can be solved as

$$b = (T^T T)^{-1} T^T y \qquad (3.9)$$

In short, instead of regressing the dependent variable (i.e., the metrology measurements) on the explanatory or independent variables (i.e., the process variables) directly as in MLR, the principal components (PCs) or scores of the explanatory variables are used as regressors in PCR. Compared to MLR, PCR has the advantage of addressing the multicollinearity problem. In addition, PCR handles noisy process variables better as usually only a subset of all the PCs are used to build the model. However, the PCs are derived without any reference to the dependent variables. In other words, PCs explain the most variation in **X**, which may not be (highly) related to the variation in **y**. Due to this reason, the performance of PCR for VM is not guaranteed.

3.5.5. Partial least squares (PLS)

Partial least squares (PLS)[85] has all the benefits of PCR while also taking the variation of dependent variables into account. Mathematically,

$$X = T\,P^T + \tilde{X} \qquad (3.10)$$

$$\mathrm{y} = \mathrm{U}\mathrm{b}^\mathrm{T} + \tilde{\mathrm{y}} \qquad (3.11)$$

where $\mathbf{U} \in \Re^{N \times L}$ and the decompositions of **X** and **y** are made so as to maximize the covariance between **T** and **U**. In other words, PLS models the inner relation that correlates the scores of independent variables with the scores of dependent variables. Therefore, PLS usually has better prediction performance than PCR, which explains why PLS and its variants are the most commonly used VM methods in industrial applications.

3.5.6. Other methods

Besides the classical VM methods discussed above, driven by the rapid development of machine learning and artificial intelligence in the past few years, other methods have been proposed. For example, RBFNN has been proposed as a VM to predict the film thickness of a chemical vapor deposition (CVD) process[86]. Support vector regression (SVR) has been applied for VM as well[87]. However, these methods have seen few applications because of the complexity involved in implementation and maintenance. In addition, it has been shown that these kernel-based or NN-based non-linear methods may not necessarily outperform linear methods in soft sensor.

3.5.7. Recursive or adaptive VM methods

For all the VM methods discussed in the previous subsections, some of them are intrinsically recursive or adaptive methods such as TSA and KF, while the others can be straightforwardly extended to recursive or adaptive variants such as recursive PLS (RPLS). For PCR or PLS-based methods, there are various adaptation schemes. In this work, adaptation is achieved by a first-in-first-out (FIFO) window-based approach wherein each step, and the latest sample is included in model training while dropping the oldest sample. This is for the sake of simplicity, not computation efficiency or adaptation performance, as neither is the focus of this work.

It is worth noting that due to the high dimensionality of the process variables, in this work, TSA only utilizes the metrology data for model building and prediction, while the process data are completely ignored. For KF based VM, to reduce the model size, $\boldsymbol{x}$ is the batch-mean of process variables. In addition, because KF is developed for dynamics systems, its good performance relies on continuous updates of the model parameters. Therefore, TSA and KF are included only as recursive VM methods.

3.6. Data preprocessing

For MLR, PCR, and PLS, the traditional batch-wise unfolding is employed to convert the 3-D matrix into a 2-D matrix of **X**. In other words, the data matrix $\mathbf{X} \in \Re^{N \times K}$ contains N batches with K variables where $K = V \times M$ with V denoting number of variables being measured, and M denoting the number of measurements taken during a batch. For the simulated CMP process, M is the same for all the batches. Therefore, the unfolding process is straightforward. For the industrial plasma etch case study, different batches have different durations and hence different M. In this work, instead of using more complicated dynamic time warping (DTW) or derivative DTW (DDTW)[88]. , we use a simple cut based on the duration of the shortest batch to remove the last few measurements for longer batches. After that, the batches are unfolded into 2-D matrix **X**.

From the above discussion, we see that all existing VM methods discussed previously make predictions by extracting linear or nonlinear correlations between process variables and metrology measurements. One drawback of utilizing process variables is that some data preprocessing steps are usually required. This is due to the characteristics associated with batch processes, such as unequal batch and/or step length and unsynchronized or misaligned batch trajectories. These preprocessing steps add complexities to VM implementation and maintenance. In addition, studies have suggested that there could be information loss or distortion caused by data manipulation during preprocessing, which could lead to performance deterioration[47]. To address this limitation, in the following section, we present the proposed feature-based VM framework, where batch statistics and other features are used as the regressors to predict metrology measurements, which naturally handles unequal batch/step lengths and/or unsynchronized batch/step trajectories. In addition, we show that the feature-based VM framework provides superior prediction performance compared to the traditional VM methods using industrial and simulated cases.

3.7. Feature-based Virtual Metrology (FVM)

The feature-based VM (FVM) framework is developed based on SPA, a process-monitoring framework we proposed previously. In SPA, various statistics are used to quantify process characteristics, and these statistics, instead of process variables themselves, are modeled for process monitoring. SPA has been applied for fault detection[38], [47], fault diagnosis[89], and virtual metrology[90]. In this work, we extend the features to not just statistics but also other features such as process knowledge-based landmark features[76], profile-driven features[77], geometry-based features[78]. In the FVM framework, framework, we hypothesize that the batch behavior can be better characterized by the *process features* than by the *process variables.* Therefore, in the FVM framework, process features instead of process variables are used as input variables to build the VM model.

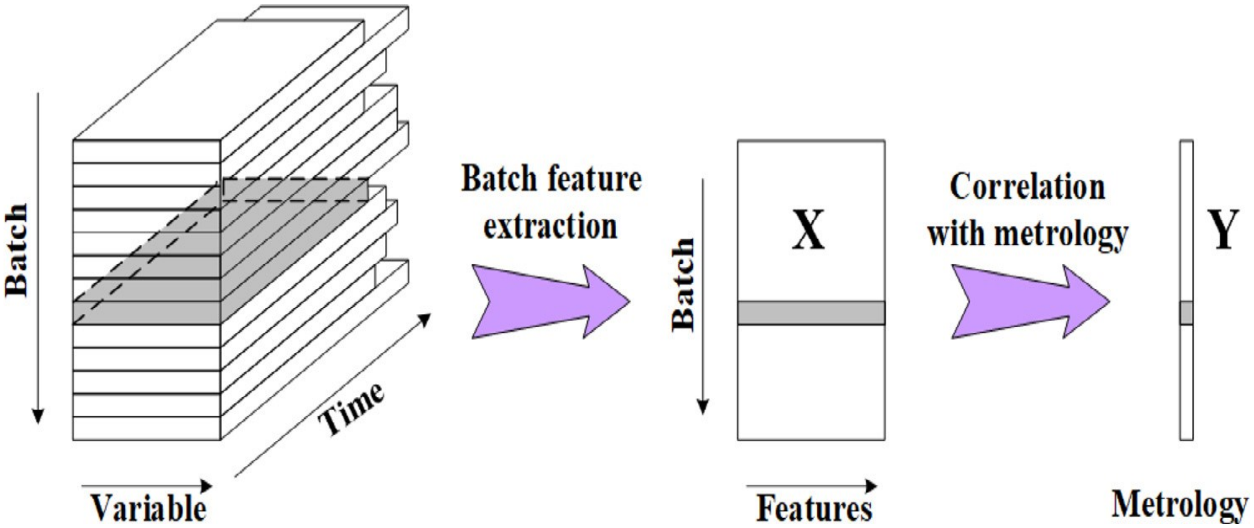


Figure 3.5 The schematic diagram of feature-based virtual metrology

Figure 3.5 provides a schematic diagram of the FVM framework, which consists of two steps. In the first step. Various features ate extracted from batch trajectories:

$$P : X \rightarrow F \tag{3.12}$$

where P denotes the operator that maps the process or machine data matrix $\mathbf{X}\in\Re^{N\times K}$ containing N batches with K variables into a feature matrix $\mathbf{F}\in\Re^{N\times S}$ containing N batches with each batch now characterized by S features. The S features can be anything that characterizes the process behavior, such as various statistics that characterize individual variables (such as the mean, variance, autocorrelation), the interactions among different variables (such as the cross-correlations), as well as other features that characterize the process (such as batch and step durations, the time integrals of power input). The features can also be extracted from each step or phase of the batch instead of lumping all steps or phases together.

In the second step, a regression method, such as PLS used in this work, is utilized to extract the relationships between the features and the metrology measurements

$$F = TP^T \ + \ \tilde{F} \tag{3.13}$$

$$y = Ub^T + \tilde{y} \tag{3.14}$$

where $\mathbf{U}\in\Re^{N\times L}$ and the decompositions of $\mathbf{F}$ and $\mathbf{y}$ are made so as to maximize the covariance between $\boldsymbol{T}$ and $\boldsymbol{U}$, similar to the regular PLS. As seen in Figure 3.5, unequal batch (or batch step) length and unsynchronized batch (or batch step) trajectories will have no effect on the FVM framework. In other words, the data preprocessing steps that are required by most existing methods, including trajectory alignment/warping and data unfolding, are eliminated by FVM.

3.7.1. Inclusion of features

The inclusion of features (i.e., what features to be retained in the mapping of Eq. (3.12)) depends on the process. FVM is a flexible framework, and feature inclusion is carried out through cross-validation to optimize the VM based on the performance measures to be introduced in the next section. Based on our experiences, the following are some general guidelines that can help with the feature inclusion process:

(1) In general, the means and standard deviations of all variables are included due to their general importance in characterizing a process.

(2) Features such as correlations, auto/cross-correlations are added based on the significance of the correlations and dynamics that exhibit between variables in the process.

(3) Higher-order statistics (HOS) such as skewness and kurtosis measure the extent of process nonlinearity and process data non-Gaussianity. Their inclusion will enhance VM performance if such characteristics are present in the process.

(4) Other non-statistical features, such as process profile, or knowledge, or geometry-based features, can also be included.

One example is given in the industrial case study, where it shows that the more features included, the better performance of the FVM model. It is worth noting that the regression methods such as PCR and PLS can naturally handle collinearity among features. Therefore, feature redundancy is usually not an issue for FVM. Although feature selection is out of the scope of this work, it has been shown that variable or feature selection can sometimes improve the performance of the regression methods. Therefore, any feature selection methods can be used as a preprocessing step for FVM if further performance improvement is desired.

As discussed in [91], the ever-increasing prevalence of big data with 4V challenges, i.e., Volume, Velocity, Variety, and Veracity [92], has necessitated the transition from the original variable space monitoring paradigm to the feature space monitoring paradigm. Therefore, we argue that the next generation VM will shift from the original variable space to the feature space as well.

3.8. Performance measures for comparing different methods

In this work, we compare the proposed FVM with the following static VM approaches MLR, PCR, and PLS. Because FVM utilizes PLS to correlate features with metrology, it can be straightforwardly extended to recursive VM by deploying recursive PL S (RPLS), which is termed recursive FVM or RFVM. We compare RFVM with some existing recursive VM approaches, including TSA, KF, and RPLS. The prediction performance of the VM methods are quantified by root-mean-square error (RMSE), the coefficient of determination (R^2), and the mean absolute percentage error (MAPE) is defined below.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \qquad (3.15)$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\,\% \qquad (3.16)$$

Where *n* is the total number of samples, $\boldsymbol{y_i}$ the actual metrology value of the output, and $\boldsymbol{\hat{y}_i}$ The VM predicted value of the output.

$$R^2 = 1 - \frac{SS_{err}}{SS_{tot}} \qquad (3.17)$$

Where $\boldsymbol{SS_{err} = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$, $\boldsymbol{SS_{tot} = \sum_{i=1}^{N}(y_i - \bar{y}_i)^2}$, and $\boldsymbol{\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i}$.

All methods are optimized using cross-validation by minimizing RMSE whenever applicable. PLS and RPLS were used as regression methods for static and recursive FVM methods.

3.9. Application to case studies

3.9.1. Data application to a simulated chemical mechanical planarization process

3.9.1.1. Chemical mechanical planarization simulation

Chemical mechanical planarization (CMP) is a widely used semiconductor manufacturing process to planarize and smooth semiconductor wafers. In CMP, as shown in Figure 3.6[93] , a wafer is held by a rotating wafer carrier, and a downforce (a.k.a. back-pressure) is applied on the wafer carrier to press the wafer face-down against a rotating polishing pad. The slightly corrosive colloidal slurry containing fine abrasive particles is released onto the pad surface[94]. The polishing pad, which is made of porous material that can hold the abrasive particles in the slurry, plays a key role by distributing slurry under the wafer so chemical and mechanical processes can occur. The material removal occurs as a result of a combination of chemical reaction (between the slurry chemicals and the wafer surface) and the repeated mechanical interaction (between the wafer surface and the polishing pad) under an applied down force[94]. Polishing pads can last from about twenty to forty hours and can complete hundreds to even thousands of wafers depending on the particular process[94], [95].

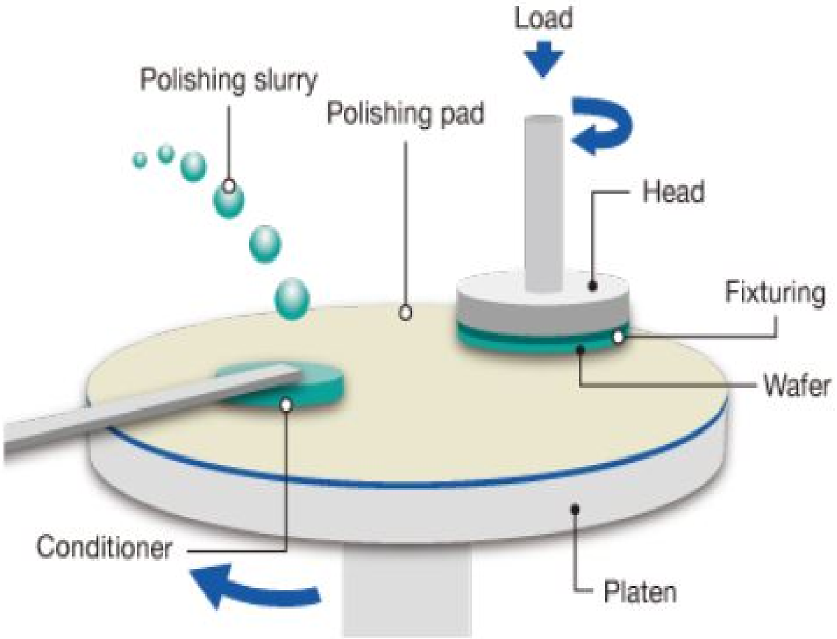


Figure 3.6 Schematic of the CMP process

In this work, the product characteristics of concern are material removal rate and within-wafer non-uniformity. The material removal rate (MRR) is determined by measuring film thickness before and after polish at each of nine sites on the wafer, and then the difference is divided by the polish time. The removal rate is the average of the nine sites on a wafer. The within-wafer non-uniformity (WWNU) is computed for each wafer as the standard deviation of the amount removed over the nine sites on the wafer, divided by the average amount removed over the nine sites, times 100[94], [96]. It is well recognized that MRR and WWNU are difficult to predict and control due to several reasons, including poor understanding of the process, degradation or wear out of polishing pads, inconsistency of the slurry, variation in physical pad properties, and the lack of in-situ sensors[96].

In this work, we adopt an industrial three-input, two-output quadratic CMP process model with linear drift[97][98] as below.

$$y1 = 2756.5 + 547.6\,u_1 + 616.3u_2 - 126.7\,u_3 - 1109.5u_1^2 - 286.1\,u_2^2 + 989.1u_3^2 - 52.9u_1u_2 - 156.9u_1u_3 - 550.3u_2u_3 - 10t + \in_{1t} \tag{3.18}$$

$$y2 = 746.3 + 62.3\,u_1 + 128.6u_2 - 152.1\,u_3 - 289.7u_1^2 - 32.1u_2^2 + 237.7u_3^2 - 28.9u_1u_2 - 122.1u_1u_3 - 140.6u_2u_3 + 1.5t + \in_{2t} \tag{3.19}$$

where the two outputs *y* 1 and *y* 2 are MRR and WWNU, respectively. The three inputs *u*1, *u*2, and *u*3 are wafer carrier downforce applied on the wafer, platen speed, and slurry concentration, respectively. *u*1, *u*2, and *u*3 are normalized to the (−1, 1) range. *t* is time, which is also normalized to (−1, 1) based on the lifetime of the polishing pad, which is set as 100 wafers in this work. $\in_{1t}\ \sim N(0, 60^2)\ and\ \in_{2t}\ \sim N(0, 30^2)$ are white noises. To illustrate the linear drifts of the CMP process, we perform baseline simulations by fixing all the inputs. Figure 3.7 shows that over the life span of a polishing pad, MRR decreases over time while WWNU increases over time (after filtering out the measurement noises), which are consistent with experimental observations[96].

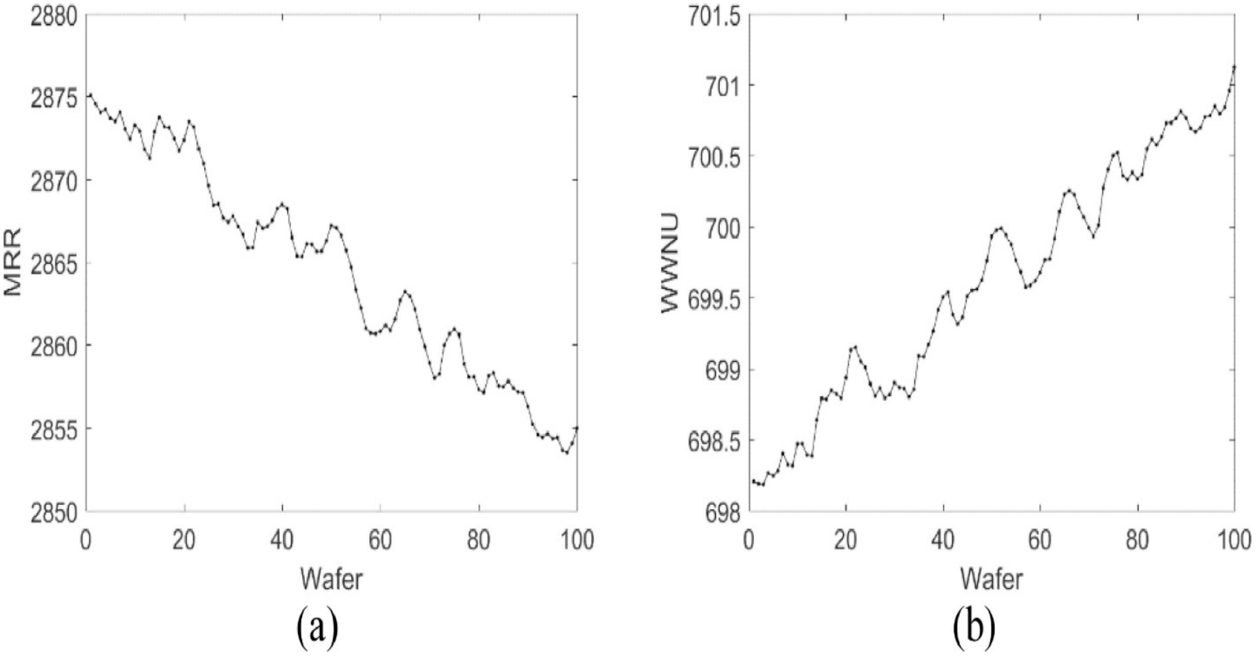


Figure 3.7 Baseline simulations with fixed u1, u2, and u3 indicate a decreasing trend in MRR (a), and an increasing trend in WWNU (b), over a polishing pad life span.

To test various VM approaches, in this work, we simulate open-loop runs without process control. To simulate the fluctuations of the inputs u_1, u_2, and u_3, integrated moving average (IMA) models were used. The sampling interval is one second and the processing time for each wafer is 1 min, i.e., 60s. To mimic production data, it is assumed that only the end of processing values of y_1 and y_2 are available, i.e., one MRR and one WWNU per wafer. The data is generated for 10,000 wafers (i.e., 100 batches with 100 wafers per batch).

3.9.1.2. Static VM approach comparison

Data from 25 batches (i.e., 2500 wafers) are used for building VMs. 25 batches are used for validation, and the rest 50 batches are used for testing. For a fair comparison, 20 Monte Carlo (MC) runs are carried out to select random batches for training, validation, and testing. Since there is no clear correlation between MRR and WWNU, separate MRR and WWNU models based on different approaches are trained, validated, and tested. The unfolded original process variables (i.e., u_1 - u_3 and t) are used as ***X*** for MLR, PCR, and PLS, while MRR or WWNU is the metrology data. For FVM of both MRR and WWNU, the following eight types of features are included: mean (*mn*), standard deviation (*st*), skewness (*sk*), kurtosis (*ku*), auto- and cross-correlations with zero to one lag (*xc*), and time integral (*it*) of u_1-u_3 , the mean of pair-wise products among u_1-u_3 (*mn2*), the wafer index in the batch (*id*). Table 3.1compares the average R^2, MAPE and RMSE over 20 MC runs for the two models developed based on different approaches. For PCR, PLS, and FVM, the optimal number of PCs used for prediction are also listed in Table 3.1 , which are obtained through validation during each MC run. The optimal number of PCs may vary from run to run due to the change of training, validation, and testing samples.

As can be seen from Table 3.1, MLR based VM performs the worst among all approaches. PCR and PLS perform similarly with reasonably high R^2 and MAPE for both MRR and WWNU. RMSE is harder to judge as it is unit or scale dependent. FVM significantly outperforms MLR, PCR, and PLS in this case study with ~0.98 R^2 and ~1% MAPE for both MRR and WWNU. These results are visualized in Figure 3.8 where the predicted and measured MRR and WWNU are plotted for MLR, PLS, and FVM. Figure 3.8 (c) and (f) demonstrates the superior performance of FVM where the predicted MRR and WWNU values agree with the measurements very well.

To investigate what factors contribute to the superior performance of FVM, the variances of **X** and **y** captured by the first three principal components (PCs) are examined. Since MRR and WWNU models behave similarly as indicated by the consistent trends in Table 3.1, only MRR models are examined. As shown in Table 3.2, among the three methods, PCR captures the most variance in ***X*** with 3 PCs, which makes sense as PCs in PCR are determined solely based on **X** without considering **y** (i.e., MRR). On the other hand, although PLS based VM captures slightly less variance in **X**, it captures more variance in MRR, which is consistent with its compromising mechanism that maximizes covariance between **X** and MRR. However, this higher variance of *y* captured by PLS does not translate into better VM performance in this case study. Compared to PCR and PLS, FVM captures significantly less variance in **X**. Since FVM **X** consists of features instead of the original variables, the captured variance in **X** cannot be directly compared to those of PCR and PLS. However, the variance captured in MRR can be directly compared, and it shows that with 3 PCs, FVM captures 96.0% of the total variance in MRR, which is significantly higher than PCR and PLS. This might explain the significantly better performance of FVM in predicting MRR than PCR and PLS, which also means that many included features are probably not relevant to MRR. This suggests that feature selection may further improve the performance of FVM.

Table 3.1 Performance comparison of various static VM approaches in predicting MRR and WWNU

	MRR				WWNU			
Approach	# of PC	R^2	MAPE (%)	RMSE	# of PC	R^2	MAPE (%)	RMSE
MLR	-	0.413	5.59	205.9	-	0.087	5.32	52.34
PCR	4-5	0.592	4.58	176.2	3-5	0.462	3.97	42.59
PLS	1-3	0.584	4.64	177.9	1-3	0.453	3.99	42.90
FVM	7	0.980	1.16	38.5	7	0.977	0.98	8.70

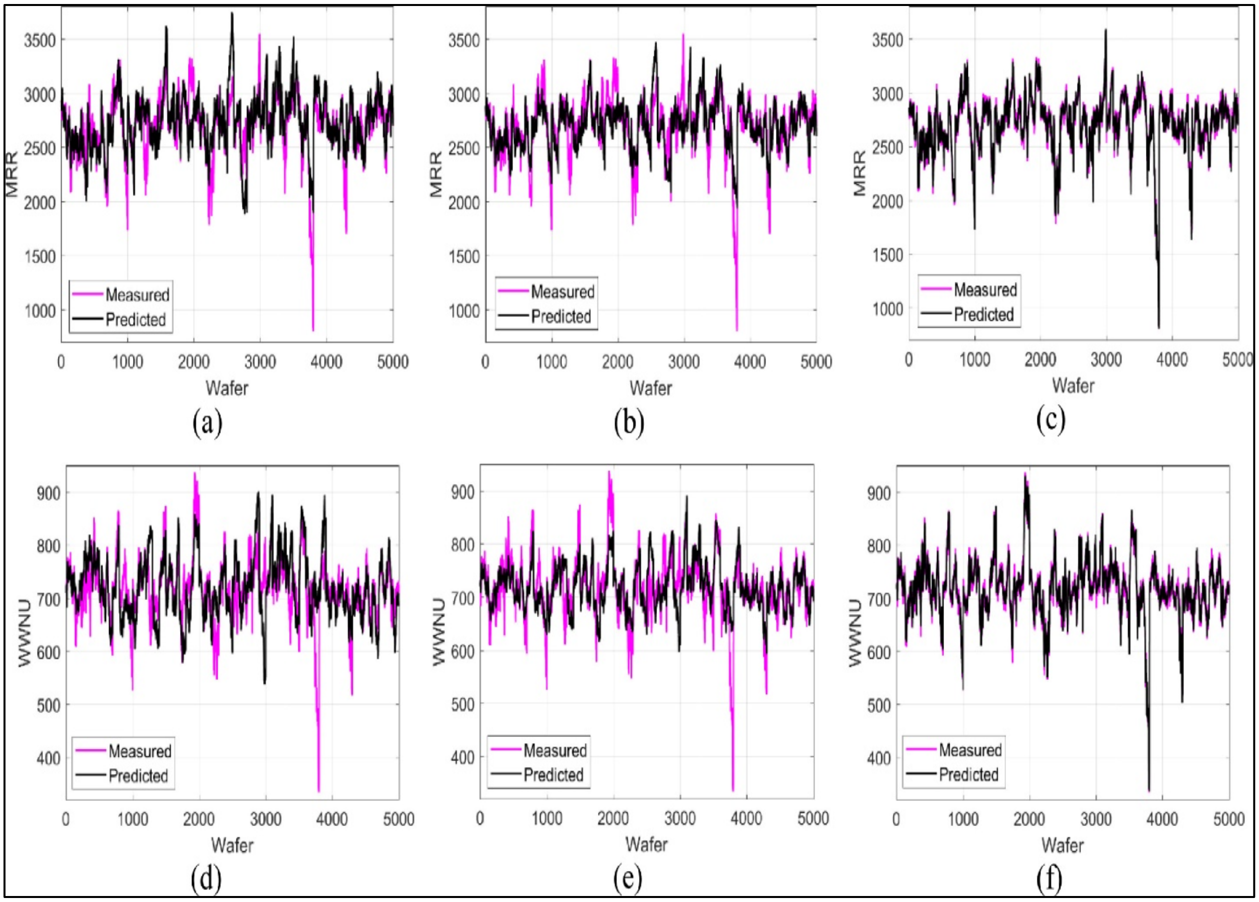


Figure 3.8 VM predicted vs measured MRR (top row) and WWNU (bottom row) based on MLR(a and d), PLS (b and e), and FVM (c and f)

Table 3.2 Variances captured by the first three PCs of different VM approaches (Averages over 20 MC runs)

Approach	Variance captured in X by first 3 PCs (%)	Variance captured in y $_1$ (i.e., MRR) by first 3 PCs (%)
PCR	78.4	58.6
PLS	75.0	67.3
FVM	39.2	96.0

Another way to compare different VM approaches is to check the linearity between PCs and MRR. Here linear regressions are performed to fit MRR to each individual PC, then R^2 of the linear regression and *p*-value of the *F* -test on the significance of the coefficient is examined. Generally speaking, R^2 measures how well the model explains the data. In this case, because the models are linear, R^2 quantifies the fraction of the variance in MRR explained by the model. The p-value measures if there is a statistically significant (linear) relationship between MRR and a particular PC. These results are listed in Table 3.3, indicates that for PCR, the PC directions may not be related to the variability in MRR at all. For example, although the first PC captures the most variance in **X**, it only captures 5.8% of the variance in MRR. On the other hand, PC 3 captures the most variance in MRR among the first 3 PCs. Once again, this is attributed to the fact that the PCs are determined solely based on **X**, and their relationship to MRR is established afterward. For PLS, because of its mechanism of considering the covariance between **X** and MRR, its first PC naturally captures the most variance in MRR, and this amount decreases monotonically with PC order. In this case, only the first PC is useful, while the other two PCs do not contribute much in capturing variance in MRR. For FVM, since it is PLS applied on features, it follows the decreasing trend of R^{2} with PC order. Here it shows that the first PC of the features can capture over 80% of the variance in MRR while the second PC also contributes to 12.6% of the total variance in MRR. Since PCs are orthogonal to each other, these R^2 values add up to the total variances in MRR captured by the first 3 PCs. Table 3.3 indicates that the features extracted from the original process variables have significantly improved the linear relationship with MRR. This is validated by the scatter plots of PC 1 and MRR for different approaches, as shown in Figure 3.9.

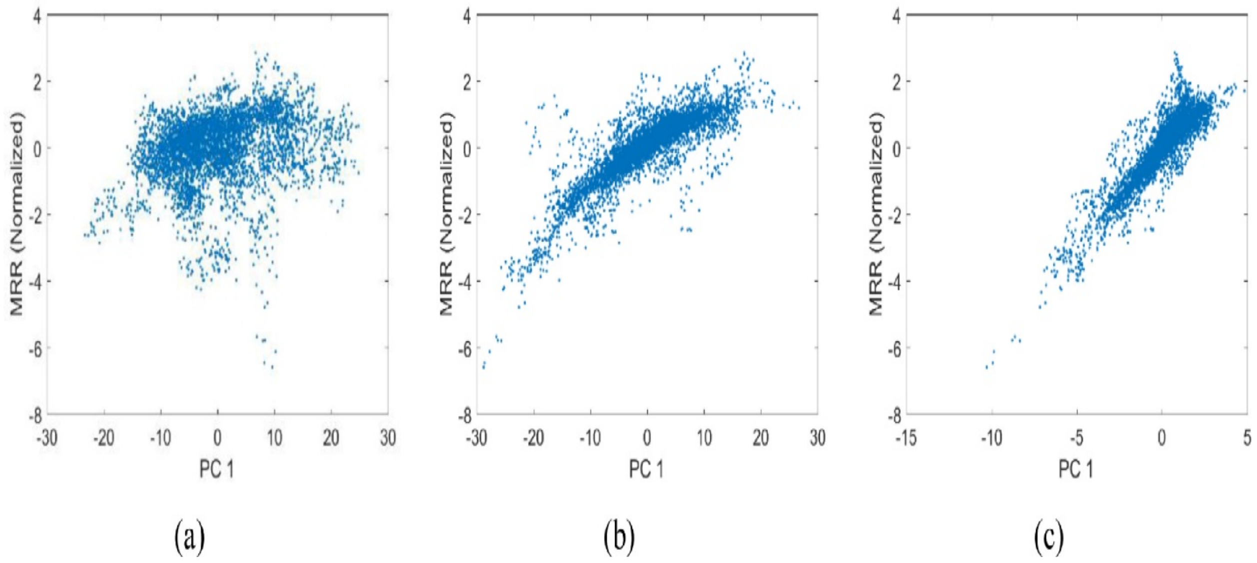


Figure 3.9 Scatter plots of the normalized MRR vs. the first PC of PCR (a), PLS (b) and FVM (c).

Figure 3.9 (a) indicates that PC 1 of PCR has the weakest linear relationship with MRR (normalized). PC 1 of PLS has much improved the linear relationship with MRR as shown in Figure 3.9(b). However, a clear curvature of the scatter plot indicates the noticeable nonlinearity between PLS PC 1 and MRR. In comparison, PC1 of FVM shows the strongest linear relationship with MRR as shown in Figure 3.9(c).

Table 3.3 R2 and p-value of linear regression between MRR and individual PC for a particular MC run

	R^2			p value of F-test		
Approach	PC 1	PC 2	PC 3	PC 1	PC 2	PC 3
PCR	0.058	0.211	0.392	< 0.001	< 0.001	< 0.001
PLS	0.699	0.004	< 0.001	< 0.001	< 0.001	0.707
FVM	0.806	0.126	-0.037	< 0.001	< 0.001	< 0.001

In summary, despite the clear nonlinearity between **X** and MRR (also WWNU) as indicated by the process models (i.e., Eqs. (18) and (19)) and illustrated in Figure 3.9(b), the features extracted show much improved linearity, which in our view, contributes the most to the much improved performance of FVM compared to other existing approaches

3.9.1.3. Recursive VM approach comparison

In this section, we compare the recursive VM approaches. Because MLR performs the worst in static VM comparison, it is not included in the comparison of recursive methods. In addition, since PCR and PLS perform similarly in the static case, only RPLS is included. As discussed in the Recursive and adaptive VM methods section. TSA and KF are recursive in nature, they are included in comparison to recursive FVM (RFVM). The same features used in the static FVM are used for RFVM. For every method, parameter tuning/optimization is done through validation similar to the static case, where the first 25 batches are used for training, the next 25 batches for validation and the remaining 50 batches for testing. One difference is that 20 MC runs are used in the static case to get the average performance of different static approaches, which is not implemented for the comparison of recursive approaches due to the online nature (i.e., the time sequence must be followed) of recursive approaches. The comparison results are listed in Table 3.4 in terms of $R^{2,}$ MAPE, and RMSE for two separate models of MRR and WWNU. As discussed above, because Table 3.4 is obtained based on a particular composition of training, validation and test samples (i.e., they are divided sequentially in time), the results in Table 3.4 cannot be directly compared to those in Table 3.1, which are the average of 20 MC runs with randomly selected training, validation and test samples. Table 3.4 shows that RPLS and KF perform similarly, which is consistent with our previously established theoretical equivalency between RPLS and KF in state estimation[99]. TSA performs significantly better than KF and RPLS while RFVM performs the best in both MRR and WWNU predictions. To further investigate the performance metrics in Table 3.4 , we plot the measured vs. predicted MRR for RPLS, TSA, and RFVM in Figure 3.10.

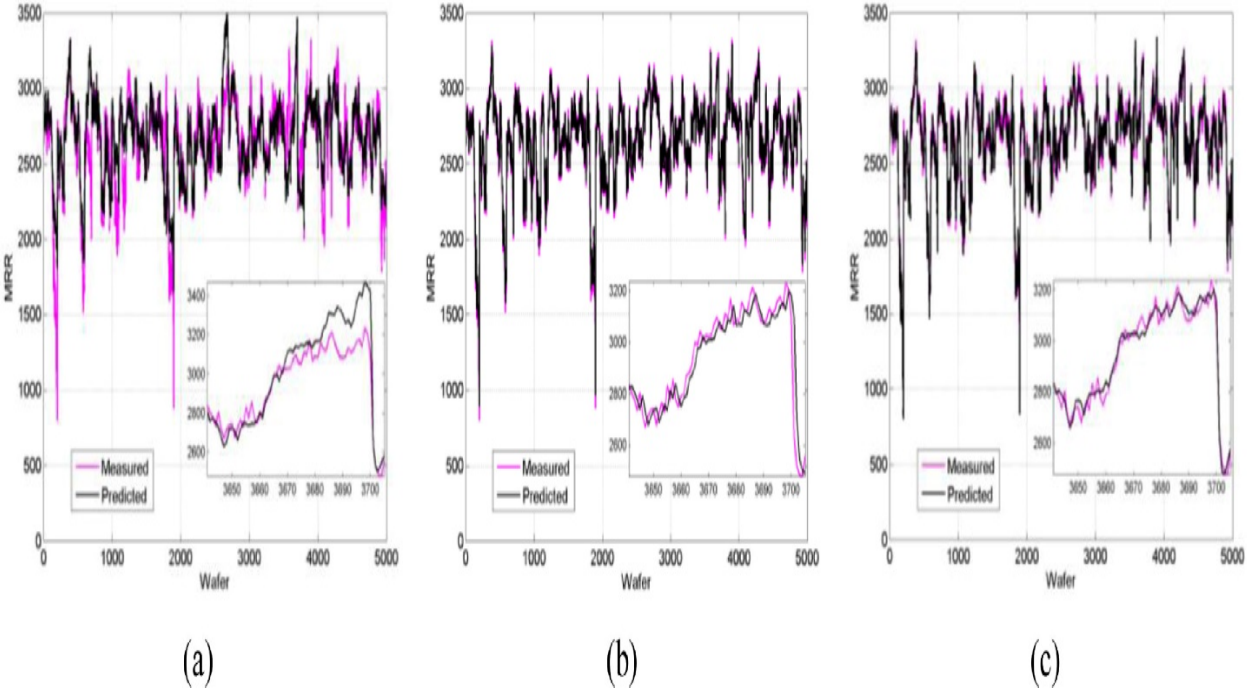


Figure 3.10 Predicted vs. measured MRR based on RPLS (a), TSA (b), and RFVM (c).

Figure 3.10 (b) shows that TSA predicted MRRs follow measurements closely. However, the zoomed-in view of a small segment in the insert of Figure 3.10 (b) shows that there is a clear one-step delay in prediction, indicating that the prediction is predominantly determined by the last measurement, which makes sense given the nature of the ARIMA models without input(s). Figure 3.10(a) shows that RPLS does not have such one-step delay in prediction, but the discrepancies between predictions and measurements are significant at places. It is worth noting that the simple FIFO window-based scheme is used to implement RPLS in this work, which means that the latest measurement weighs as much as the oldest measurement in the training data. If a weighting mechanism (e.g., exponentially weighted moving average or EWMA) is employed, we expect much-improved performance from RPLS. Similar to RPLS, RFVM does not have a one-step delay in prediction since it makes use of the current process variable (i.e., x_t) in the model. In addition, the model predictions agree with the measurements very well. Since RFVM is implemented using RPLS, the performance of RFVM could be further improved if, for example, EWMA is implemented instead of FIFO.

Table 3.4 Performance comparison of various recursive VM approaches in predicting MRR and WWNU

	MRR				WWNU			
Approach	# of PC	R^2	MAPE (%)	RMSE	# of PC	R^2	MAPE (%)	RMSE
RPLS	3	0.607	5.35	191.0	3	0.409	4.59	46.4
KF	-	0.608	5.25	190.8	-	0.413	4.45	46.2
TSA	-	0.934	2.05	78.2	-	0.923	1.69	16/7
RFVM	11	0.984	1.16	38.5	13	0.979	0.99	8.7

3.9.2. Application to an industrial case study

In this section, a dataset collected from a plasma etch system at one of Texas Instruments' wafer fabs[72] is used to compare the proposed feature-based VM and other VM methods. The dataset contains the recorded values of 18 Optical Emission Spectroscopy (OES) signals collected every 0.1 s for 1121 wafers. The dataset also contains the metrology measurement values of the sheet resistance, which is one of the most important electrical- test parameters used in the semiconductor manufacturing industry to assess the electrical quality of a product. Figure 3.11 shows a schematic of a typical etching Plasma etch process.

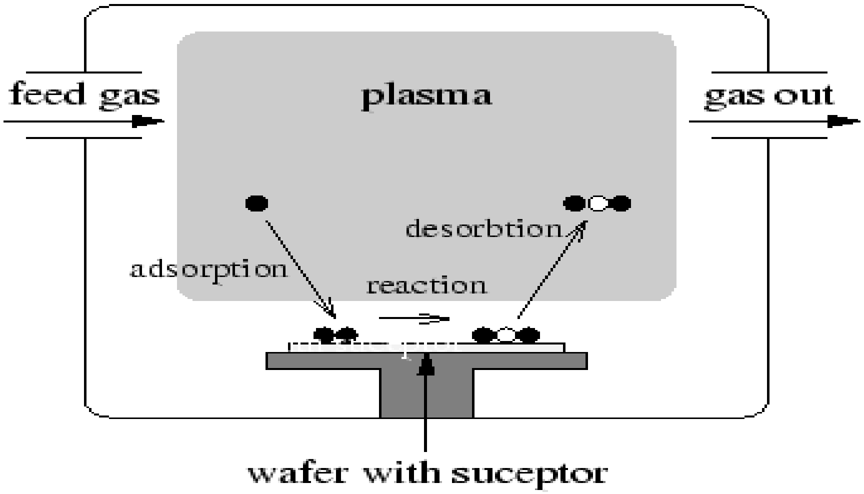


Figure 3.11 Schematic view of the etching process

The goal of VM is to predict the end-of-batch sheet resistance using the OES signals. Sheet resistance is defined as the resistance of a square sheet of material with current flowing parallel to the plane formed by the square sides. One OES signal of several wafers is plotted in Figure 3.12, which shows the typical characteristics of a semiconductor machine data: unequal batch length or process duration; large variations between wafers and unsynchronized trajectories. To apply traditional VM methods such as PLS on this type of data, several data pre-processing steps have to be taken, including trajectory alignment or time warping to make trajectories equal length, and trajectory unfolding to flatten the 3-D structure into a 2-D matrix. As discussed in the Data pre-processing section, for simplicity, we use a simple cut based on the duration of the shortest batch to remove the last few measurements for longer batches. After that, the batches are unfolded into 2-D matrix X following the traditional batch-wise unfolding as described in the Data pre-processing section.

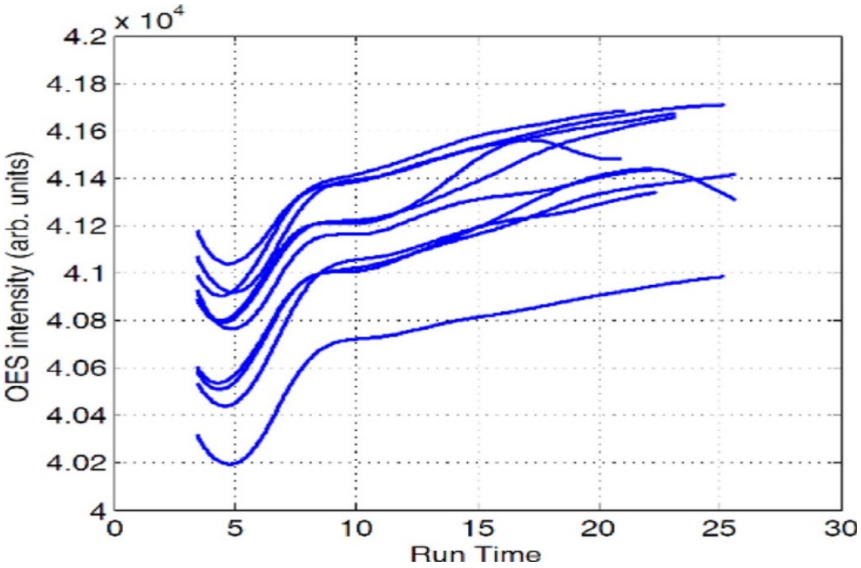


Figure 3.12 A sample OES signal of several wafers

3.9.2.1. Static VM approach comparison.’

In this subsection, the static FVM is applied to the dataset discussed previously to predict the sheet resistance using the OES data.

Table 3.5 Comparison of different Static VM methods

Model	# of PC	R^2	MAPE (%)	RMSE
MLR	–	0.049	10.27	0.0313
PCR	21	0.396	8.55	0.0253
PLS	50	0.437	8.12	0.0245
FVM	18	0.718	5.94	0.0173

The features used in FVM include: time integral of the OES signals (*it*), univariate statistics including mean (*mn*), standard deviation (*st*), skewness (*sk*) and kurtosis (*ku*), as well as the means of pair-wise products (*mn2*) of all 18 variables. The performance is compared with other VM methods. For all VM methods, 70% of the data (784 wafers) are utilized for model building and the rest 30% of the data (337 wafers) are used for testing. Table 3.5 compares R^2, MAPE, and RMSE of FVM to those of MLR, PCR, and PLS. MLR performs poorly due to the high dimensionality of the independent variables after unfolding and the multicollinearity among them. In this industrial case study, PLS performs slightly better than PCR while FVM significantly outperforms all other methods in terms of R^2, MAPE, and RMSE.

3.9.2.2. Recursive VM approach comparison

In this subsection, RFVM is applied to the dataset and its performance is compared with those of other recursive VM methods. The initial VM model is built based on the training data of 784 wafers and is updated when new data becomes available. The same features used in the static FVM are used for RFVM. The comparison results are summarized in Table 3.6. RPLS and KF perform similarly, which resembles the simulated case study. TSA performs better than RPLS and KF without the use of the inputs (i.e., the OES measurements).

Figure 3.13 shows the comparison of measured vs. predicted sheet resistances of RPLS, TSA, and RFVM. Figure 3.13 (b) reveals a persistent one-step delay in the prediction of TSA. This is again similar to the simulated case study, indicating that TSA prediction is predominantly determined by the last measurement. RPLS and RFVM do not have this phenomenon, as shown in Figure 3.13 (a) and (c). Both Figure 3.13 and Table 3.6 demonstrate the superior performance of FVM compared to RPLS, KF, and TSA.

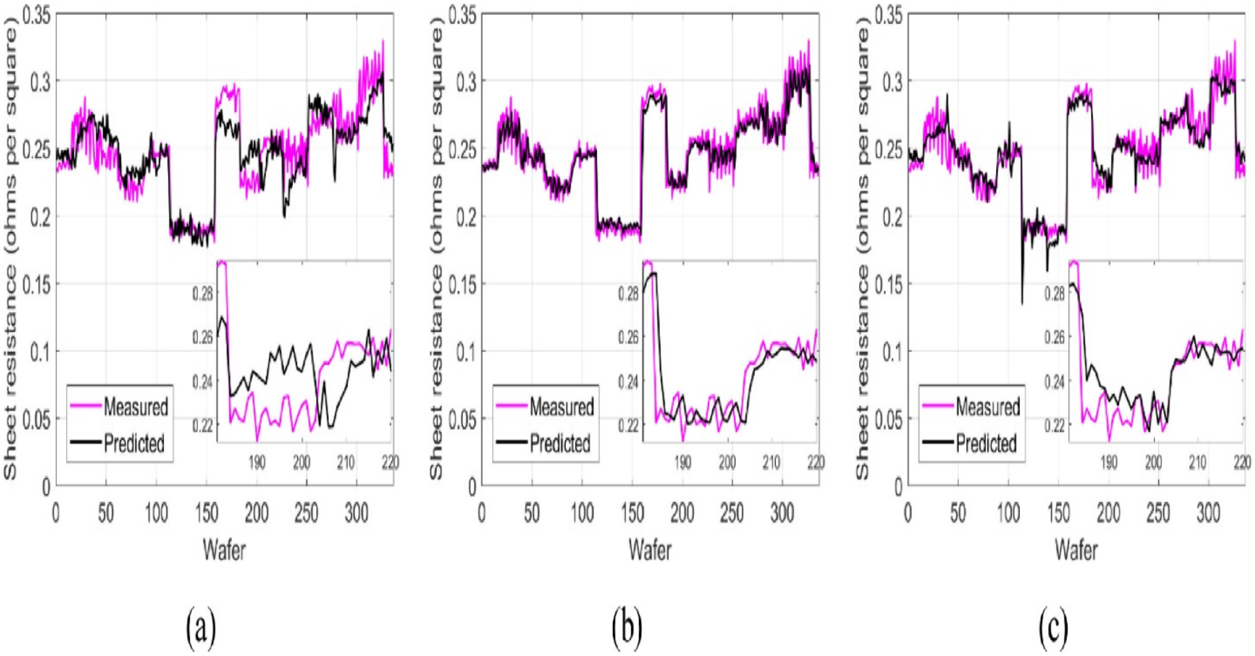


Figure 3.13 Predicted vs. Measured Sheet resistance based on RPLS (a), TSA (b) and RFVM (c).

Table 3.6 : Comparison of different recursive VM methods

Model	# of PC	*R* 2	MAPE (%)	RMSE
RPLS	15	0.689	5.73	0.0182
KF	–	0.697	5.31	0.0179
TSA	–	0.776	4.20	0.0154
FVM	72	0.855	3.77	0.0124

To investigate the effect of feature inclusion on the performance of FVM, we performed RFVM by including different sets of features in RFVM. Table 3.7 lists the features included, the number of principal components determined or optimized through validation, and the resulted performance measures of RFVM. By comparing Table 3.6 and Table 3.7, it can be seen that by including *mn* alone, RFVM achieves good performance similar to RPLS. By including *mn, st* and HOS (i.e., *sk* and *ku*), RFVM outperforms RPLS and KF, which demonstrates the importance of including HOS as features in FVM. By including *mn* and *mn2*, RFVM outperforms all other VM methods listed in Table 3.6 , which indicates that the process nonlinearity is significant. In addition, although the nature of the non-linearity is unknown due to the complexity of the plasma etch process, the means of pair-wise products (*mn2*) provide a good capture of its nonlinearity. Finally, by using all features, including *mn, st, sk, ku, mn2*, as well as *it* (the time integral of the OES signals as a measure of total power input at different frequencies) of all 18 variables, RFVM provides the best performance among all cases listed in Table 3.7. Table 3.7 indicates that the more features included, the better the performance of RFVM. This is generally true based on our experiences and can be explained by the fact that PLS can naturally handle collinearities among features. In other words, including more features can add process information to the model while feature redundancy poses no issue for FVM. It is worth noting that variable or feature selection can sometimes improve the performance of the regression methods. Therefore, any feature selection methods can be used as a preprocessing step for FVM if further performance improvement is desired. This subject is outside the scope of this work. Further investigation is worth pursuing and feature selection can be integrated as part of the FVM framework.

Table 3.7 : Effect of feature inclusion on the performance of RFVM

Features in RFVM	# of PCs	*R* 2	MAPE (%)	RMSE
Mn	11	0.607	6.57	0.0204
mn, st	16	0.627	6.22	0.0199
mn, st, sk	13	0.685	5.88	0.0183
mn, st, sk, ku	19	0.725	5.45	0.0171
mn, st, sk, ku, it	55	0.797	4.65	0.0147
mn, mn2	63	0.802	4.55	0.0145
mn, st, sk, ku, it, mn2	72	0.855	3.77	0.0124

3.10. Conclusions

A FVM framework and its recursive/adaptive variant RFVM are proposed in this work to address the challenges presented in semiconductor VM applications, such as unequal batch/step duration and/or unsynchronized trajectories; and a large number of variables caused by data unfolding. Because FVM does not require any data preprocessing steps, it is uniquely suited for automatic online applications. The performances of FVM and RFVM are compared with several commonly used VM approaches using a simulated and an industrial case study.

Among static or off-line VM approaches, both simulated and industrial case studies demonstrate that MLR is not a good VM approach, especially where there are many independent variables (e.g., partly due to batch unfolding) and there exists multicollinearity among them. We have also demonstrated that PCR could be problematic for VM as the selected PCs are based on their capabilities in capturing variance among the independent variables, which may not be relevant to the variance of the metrology data. In the simulated case study, the first PC only captures 5.8% of the variance in MRR while PC 3 captures 39.2%, the most among the first 3 PCs. PLS models the inner relation that correlates the scores of independent variables with the scores of dependent variables, which theoretically to enable PLS to have better performance than PCR. Although this point is not shown in the simulated case study, PLS does perform better than PCR in the industrial case study. The proposed FVM approach performs the best in both simulated and industrial cases studies. The analyses reveal that when there exists nonlinearity between independent and dependent variables, the extracted features show much-improved linearity, which enables FVM to capture a significantly larger amount of variance in the dependent variable(s) with only the first few PCs. This point, in our view, contributes the most to the much-improved performance of FVM compared to other existing VM approaches.

Among recursive or online VM approaches, KF performs similarly to RPLS. This is expected as the theoretical equivalency between RPLS and KF in state estimation has been established. TSA performs surprisingly well, even without any consideration of any input, in both simulated and industrial case studies. Analysis reveal that this is due to the fact that the metrology data in both cases are highly autoregressive time series, and the TSA prediction is predominantly determined by the last measurement. The performance of TSA without input is not guaranteed if metrology time series are not significantly autoregressive. RFVM outperforms all existing recursive VM approaches in both simulated and industrial case studies and its performance can be potentially further improved if a weighting mechanism such as EWMA is implemented instead of FIFO for highly autoregressive metrology measurements such as the ones in this study.

4. Chapter 4. Machine learning techniques for process modeling and condition monitoring using non-invasive IIoT vibration sensors

4.1. Introduction

Centrifugal pumps and compressors are one of the most important types of equipment in the process industry and are used for the transfer of oil and gases from one location to another. The goal of this work is to predict important properties using non-conventional, non-invasive IIoT sensors. Data-driven soft sensors have been used to capture many complex relationships[100], [101] between process information, such as product quality, and other process properties, which are easier to measure. Soft sensors aid the data-driven decision-making process, so that faster and more informed control actions can be taken in an industrial process. For this work, a centrifugal testbed with IIoT sensors was used. A centrifugal pump is a system with several 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 [102]–[104], such as the detection of faults or defects in gears, rotors, shafts bearings and couplings. 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 information regarding different operating stages of a pump will captured and in turn be successfully modeled. The centrifugal pump setup can be seen in Figure 4.1 below.

It is worth noting that this work is an extension to this previous work[50]. The motor speed, i.e., RPM and the water flow inside the pipe of the system, i.e., flowrate are important properties of a centrifugal pump, and predicting them can provide useful insights into the condition of the

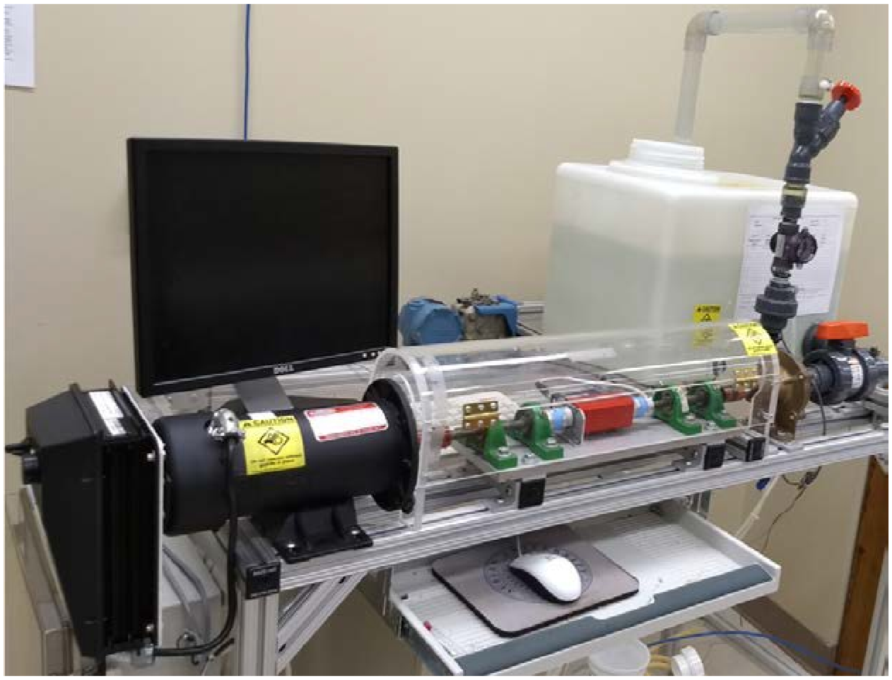


Figure 4.1 Multi stage centrifugal pump setup

centrifugal pump. The goal of this work is to build a model that can predict the RPM and flow rate for any existing condition using the vibration data collected for the centrifugal test bed with the help of IIoT sensors.

4.2. Experimental setup

The pump assembly contains a variable drive motor, pump impeller, impeller casing, coupling, electrical connections, and knobs for changing the pump 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 the pump discharge valve or by changing pump RPM. The pump RPM and flow rate are continuously indicated and will be used as a base value for building predictive models. The pump is operated in the RPM range of 1500-2500. For the discharge valve, the minimum flow at 1500 RPM, and the minimum discharge opening is 5 gallon per minute (GPM) as measured by the flow meter. Also, maximum flow rate can be achieved at 2500 RPM, which is about 16 GPM.

Digital accelerometers were used in this work to collect vibration data. These digital accelerometers have an advantage over analog accelerometers due to the fact that analog accelerometers have more manual connection points, thus increasing the sensor failure points while increasing the size of overall sensor setup.

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 samplingrates, as well as a wider temperature range (-40°C to 85°C), a smaller size (3mm X 5mm X 1mm), and strong community support[105]. Raspberry Pi is used as a master device to control each sensor based on user requirement, as well as allow data tracking and labeling to increase productivity and save time. Raspberry PI has extremely low power drawing, small form factor, no moving parts resulting in a smaller chance of failure and can work with multiple types of sensors and devices. More details on Raspberry Pi can be found in here[106]–[108].

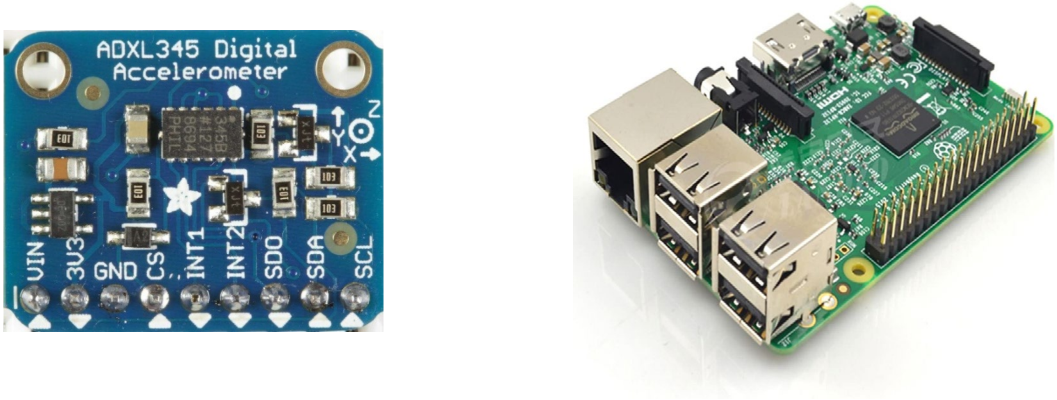


Figure 4.2ADXL 345 accelerometer sensor and Raspberry pi

5 ADXL345 sensors are used in the data collection process. Sensor locations were selected keeping in mind the properties of interest, i.e., RPM and flow rate, so variations within the system could be well captured. Sensors were mounted on the motor casing, impeller casing, coupling joint motor, impeller, the pipe fitting, and the loose end of the pipe. The sensors were fitted with maximum contact to ensure data quality. Figure 4.3 below shows the sensors marked on the experimental setup.

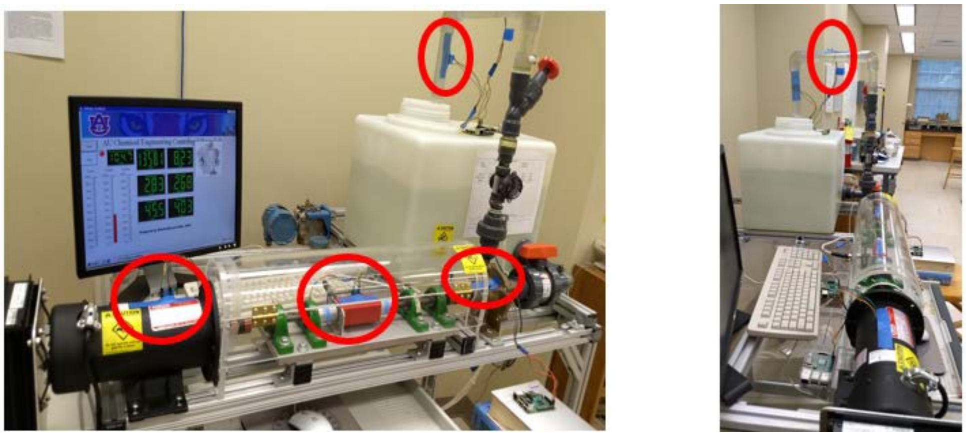


Figure 4.3 IIoT enabled centifugal pump testbed (Sensors are marked in red)

Also, the Figure 4.4 below shows the schematic of the testbed with sensor location for better understanding.

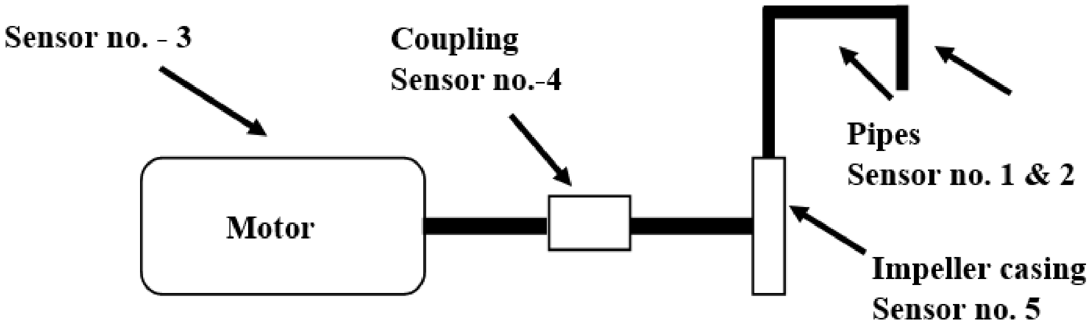


Figure 4.4 Schematic of testbed showing sensor location

4.3. Data collection and observations

The goal of this work was to develop a monitoring framework by estimating process information such as RPM & flowrate with the help of vibration signals obtained from non -invasive IIoT sensors. The relationship between RPM and vibration signals was identified first, followed by flow rate in the previous work. As mentioned before, the goal of this work is to develop a model using knowledge-based feature engineering to predict the RPM and flow rate of the system. As flow rate of the system can be independently controlled with pump discharge valve or RPM knob, the relationship between vibration signals & flow rate can be different for distinct setting of discharge valve and RPM.

Keeping the above-mentioned considerations in mind, the vibration signal data was collected for different combination of conditions. First, RPM of the pump is fixed and then vibration signals are collected at different flow rates. This process was carried out at different RPM values covering entire range of pump operation. Each unique combination of RPM & flowrate is considered a condition. A list of the conditions can be seen in Table 4.1.

Table 4.1 List of conditions and corresponding Flowrate and RPM

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		

Table 4.1 indicates the approximate values of flow rate around which the pump drifts for an RPM value. 10-minute data were collected for each condition. For example, for an RPM value of 1500, the flow rate is set to 5 GPM, and the vibration data is collected for a duration of 10 minutes. This constitutes data collected for a single condition. Microsecond version Unix epoch time (UET) was used to synchronize the time of all sensors. It was observed that, on average the flowrate value changes every half second, so the sampling frequency of RPM & flowrate values were fixed at 3 Hz.

4.3.1. Data characteristics

As seen in the Figure 4.5, RPM values deviate due to drifting in the pump over a minute range. This behavior is observed for all RPM conditions, with lower drift in the low RPM

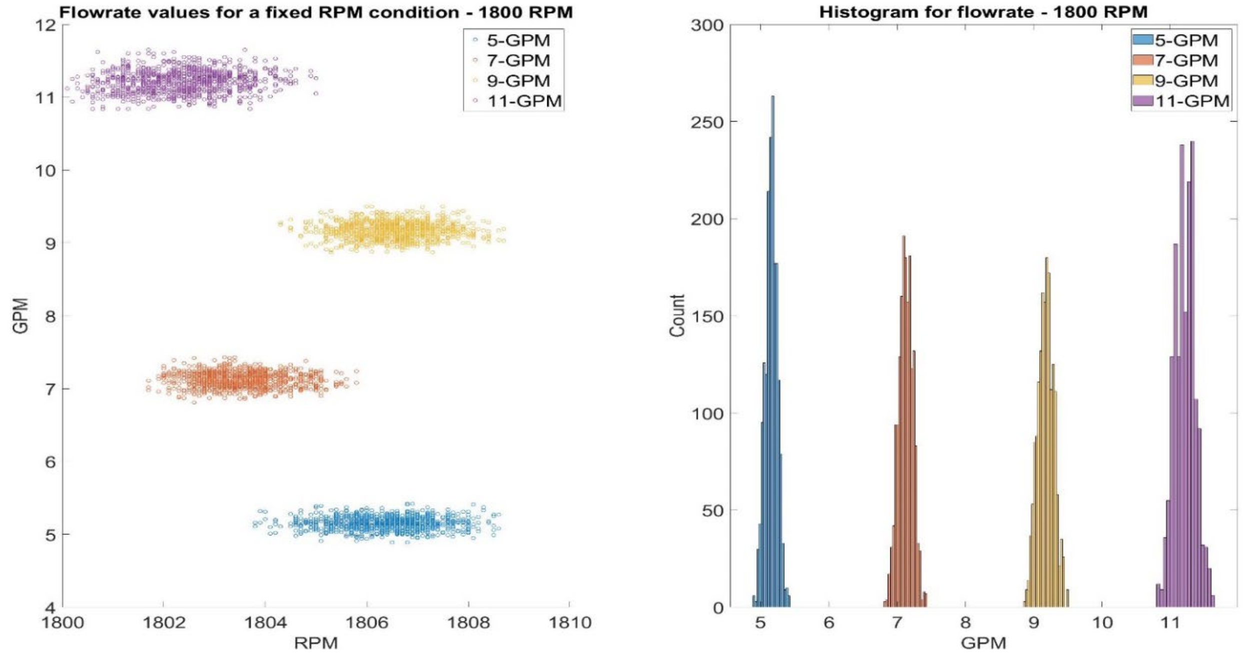


Figure 4.5 GPM values at 1800 RPM and corresponding Histogram

conditions. The Figure 4.5 also shows the histogram for flowrate values measured at 1800 RPM. The flow measurement clearly shows a gaussian distribution. This can be due to the drift in the pump as well as noisy response from the flowmeter. Vibration sensors were tested for a couple of sampling rates. For experimental data collection, 1600 Hz was selected instead of 3200 due to less noisy characteristics.

The data also shows some typical big data characteristics such as extremely noisy high frequency data, unequally spaced real-time data, large sections of missing data, non-periodic and non-stationary symbols, etc. More information can be found in the previous work here [109].

4.4. Data pre-processing & modeling and raw data

As discussed before, the data is extremely noisy and needs to be cleaned and pre-processed. Also, the sampling frequency of the measurements, i.e., RPM and Flowrate, is 3 Hz and that of the vibration signals is 1600 Hz. Vibration data from sensor-4 is used throughout this work as it is most relevant for predictive modeling of RPM and Flowrate. Sensor 4 is located on the coupling. UET was used to synchronize the vibration signals with the measurements, i.e., RPM and flowrate. For a UET in measurement as a reference point, the closest point in the vibration signals was detected. Further, 400 points before and 400 points after the detected point were considered as samples corresponding to the reference measurement. This way, each measurement corresponding to a particular condition will have 800 samples in the X, Y, and Z direction for the data collected from ADXL345. For example, if a condition had 1800 samples corresponding to an RPM and Flowrate with deviations within the range, the corresponding raw data would be 1800x800x3 in dimension. Out of all the conditions, data collected for 10 different RPM conditions were used for model building starting from 1500 RPM - 2500 RPM with intervals of 100 RPM. 80% of the samples from a condition set were considered for training, while the last 20% in chronological order from each condition were considered for testing. In an industrial application, the data collected in real-time is used to predict the condition, so it makes sense to use the later 20% of the data for testing.

4.4.1. Modeling on raw data and PLS

We use PLS to build a baseline model. In condition monitoring, it is highly unlikely that the raw time-domain data can extract relevant information and can be used to predict the properties of interest. However, PLS will be used in later stages to compare performance of various scenarios after feature engineering. PLS regression is a well-established approach for regressing independently measured variables which are highly correlated, have high measurement noise and have high dimensionality. PLS regression first extract orthogonal variables and then using Ordinary Least Squares (OLS), relationships are extracted between the orthogonal variable and the measurement variables, in this case RPM and Flowrate. We use PLS based on Nonlinear -iterative partial least squares (NIPALS) algorithm developed by Wold et al. More information can be found in here[110]. The execution is carried out in python via Scikit-learn[53].

PLS is one of multivariate statistical techniques to find the relationship between predictor variables and response variables. PLS aims to extract the PLS components that satisfy three objectives; (1) the best explanation of the X matrix (predictor variables); (2) the best explanation of the Y matrix (response variables); (3) the greatest relationship between X matrix and Y matrix. Nonlinear-iterative partial least square (NIPALS) developed by Wold [111] is a popular algorithm to implement PLS. More information on the algorithm and its properties are discussed in [112]–[115].

$\boldsymbol{X}_{n\times m}$ denotes the predictor matrix, which consists of n samples and m predictor variables; $\boldsymbol{Y}_{n\times l}$ denotes l response variables for the n samples. The regression equations are the following:

$$\boldsymbol{X}_{n\times m} = \boldsymbol{T}_{n\times p}\boldsymbol{P}_{m\times p}^{T} + \boldsymbol{E}_{n\times m} \qquad (4.1)$$

$$\boldsymbol{Y}_{n\times l} = \boldsymbol{U}_{n\times p}\boldsymbol{Q}_{l\times p}^{T} + \boldsymbol{F}_{n\times l} \qquad (4.2)$$

where p is the number of principal components; $\boldsymbol{T}_{n\times p}$ and $\boldsymbol{Q}_{l\times p}^{T}$ are the score matrices; $\boldsymbol{P}_{m\times p}$ and $\boldsymbol{Q}_{l\times p}$ are the loading matrices; $\boldsymbol{E}_{n\times m}$ and $\boldsymbol{F}_{n\times l}$ are the error or residual matrices, respectively. The PLS model maximizes the covariance between $\boldsymbol{T}$ and $\boldsymbol{U}$.

The training set is further divided into training and validation data for hyperparameter optimization, in this case, the number of orthogonal components or PCs. Root mean square error (RMSE) is used as a performance metric and is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \qquad (4.3)$$

Where N= number of samples. Figure 4.6 shows the predictions of RPM using PLS. The RMSE of the test set is 190.97, and 25 PCs are selected using Cross-validation. It is seen clearly that PLS on raw vibration data in the time domain fails to capture any relationship between the predictors and RPM.

Similarly for Flowrate, the predictions using PLS on raw data can be seen in the Figure 4.7. The results are extremely poor as expected. The RMSE on test set is 2.94 and 20 different PCs were selected using Cross validation.

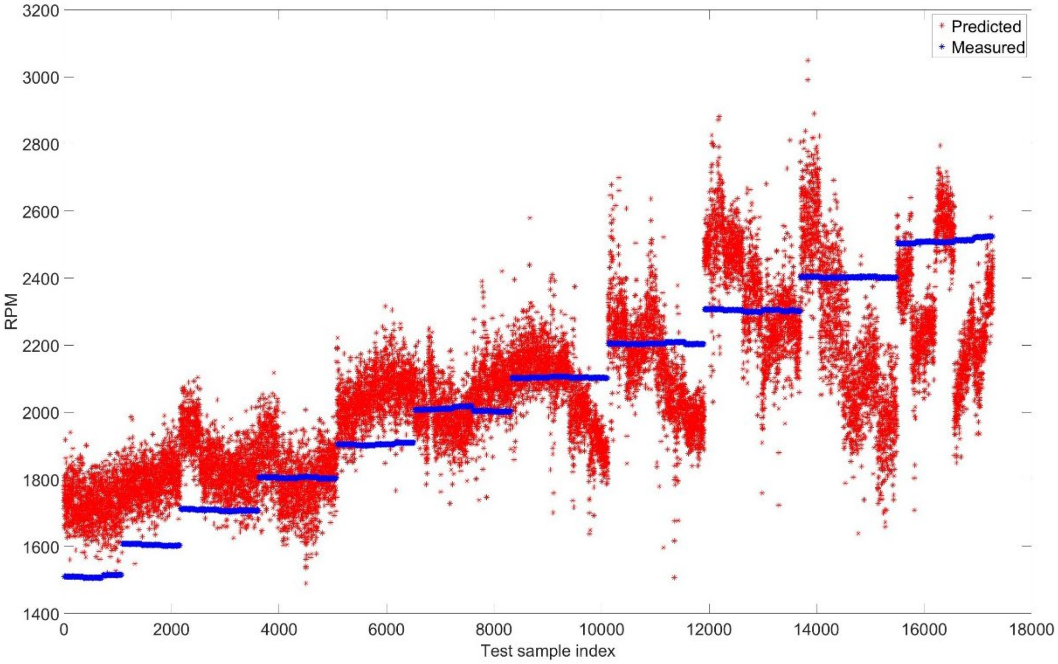


Figure 4.6 PLS model prediction performance for RPM (Raw data)

It is clear at this point that the raw vibration data cannot be used to predict RPM and flow rate. The data needs to be processed to extract meaningful information or features that can aid the prediction process. We extract features in frequency domain to extract meaningful information that can be modeled. Before extract features in frequency domain, we perform Outlier detection as well to refine the data further to remove extremely noisy data points.

4.4.2. Outlier detection

The local outlier factor (LOF) is a technique that attempts to harness the idea of nearest neighbors for outlier detection. Each example is assigned a scoring of how isolated or how likely it is to be outliers based on the size of its local neighborhood. Those examples with the largest

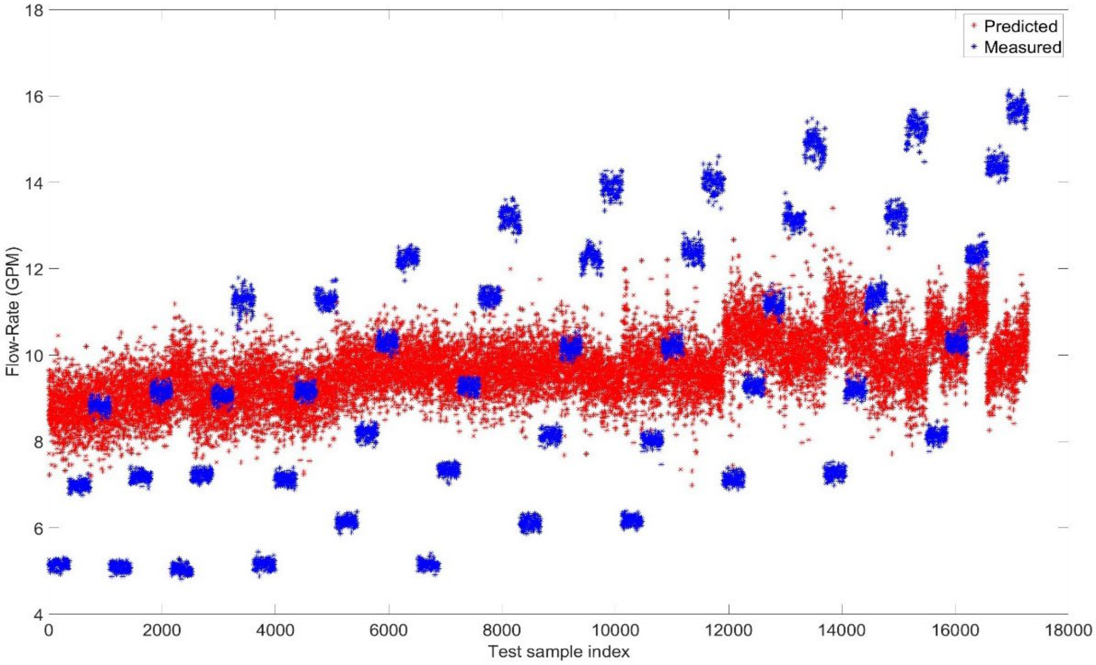


Figure 4.7 PLS model prediction performance for Flowrate (Raw data)

score are more likely to be outliers. More information can be found in here[116]. For LOF, we used scikit-learn[53] in python to remove the outliers from the data.

A simple schematic of the approach can be seen in Figure 4.9 below. We use a two-step approach for outlier detection. First, data for each condition is filtered for outliers using LOF. LOF is chosen because of its inherent nature to cluster nearest neighbors. We already know the conditions and any deviations outside the conditions, i.e., RPM and GPM are due to abrupt changes in the system and noise and can be filtered out. This way each condition vibration data is filtered separately for outliers. Further, samples for which the measurement data exceeds by 3 standard deviation thresholds are removed as they are samples that are noisy or appear due to relatively larger fluctuations in the system.

Corresponding vibration data samples are removed as well. Figure 4.8 shows an example of points that lie outside the 3 standard deviations region and are due to relatively larger fluctuations during data collection for the 1500 RPM and 5 GPM conditions. A similar approach is used to remove outliers from all conditions considered for modeling.

Further, as mentioned before, the initial 80% of the data each condition is used for testing while the rest of the data is used for testing using different ML approaches for predicting the flow rate and RPM. After outlier removal within each condition, the data for all the conditions are used together for training and testing as the goal of this work is to use a universal model for the prediction of RPM and Flowrate. 65563 samples are used in training and 16388 samples in testing the proposed approach.

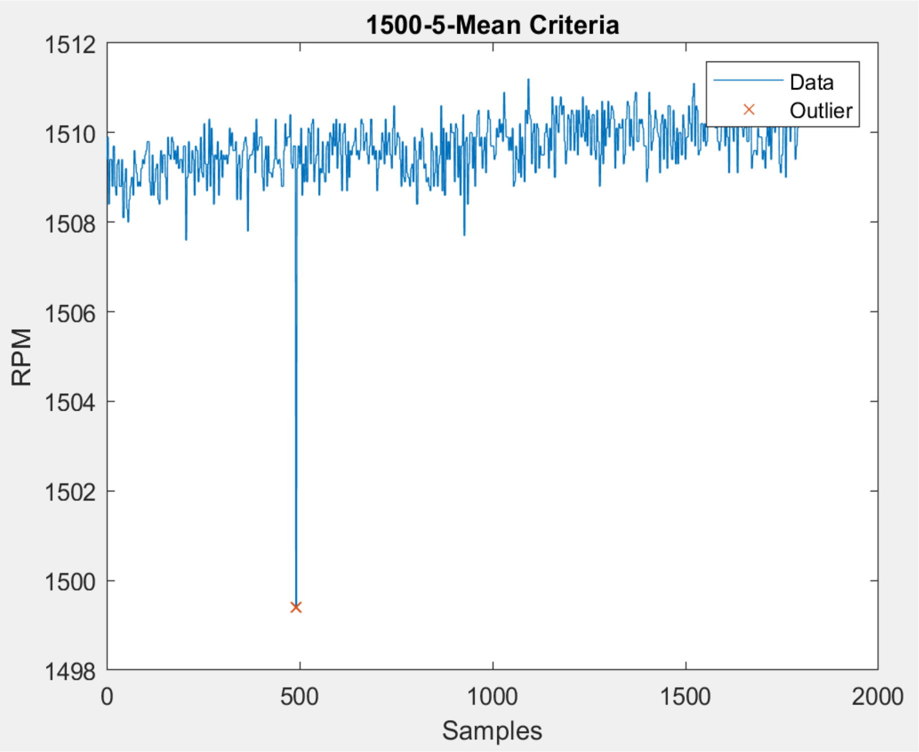


Figure 4.8 Outlier removal in measurements and corresponding vibration signals using standard deviation threshold

4.5. Feature extraction in the frequency domain – Lomb-Scargle algorithm

Fourier analysis or fast Fourier transform (FFT) is a widely used approach that converts a signal from its original domain (often time or space) to a representation in the frequency

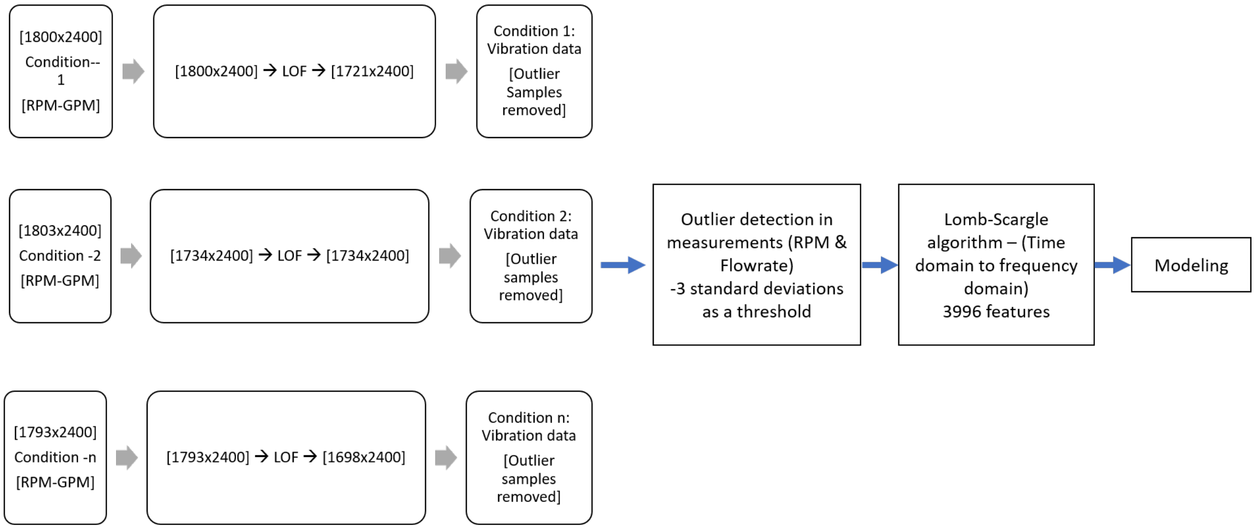

[1800x2400]
Condition--1
[RPM-GPM]
[1800x2400] → LOF → [1721x2400]
Condition 1: Vibration data
[Outlier Samples removed]
[1803x2400]
Condition -2
[RPM-GPM]
[1734x2400] → LOF → [1734x2400]
Condition 2: Vibration data
[Outlier samples removed]
Outlier detection in measurements (RPM & Flowrate)
-3 standard deviations as a threshold
Lomb-Scargle algorithm – (Time domain to frequency domain)
3996 features
Modeling
[1793x2400]
Condition -n
[RPM-GPM]
[1793x2400] → LOF → [1698x2400]
Condition n: Vibration data
[Outlier samples removed]

Figure 4.9 Schematic of outlier detection approach for vibration data and measurement data

domain and vice versa. The discrete Fourier transform (DFT) is obtained by decomposing a sequence of values into components of different frequencies. More information on the FFT can be found here[117]. However, FFT works when the signals are equally spaced. FFT fails when the data points are unequally spaced and cannot give a reliable spectrum.

To deal with this limitation, Lomb-Scargle's algorithm is used in our work. The output of Lomb-Scargle's algorithm is the power spectrum density (PSD) of the signal under consideration. Lomb-Scargle's algorithm does not require the samples to be equally spaced. This approach is widely used in astronomy. The Lomb-Scargle periodogram (after Lomb and Scargle) is a commonly used statistical tool designed to detect periodic signals in unevenly spaced observations. More information can be found here[118]–[120]. We use astropy package in python to implement the algorithm. The raw vibration signals in all 3 directions, X, Y, and Z that are collected from ADXL345, are used to extract the frequency domain features. For each direction, frequency spectrum was obtained from 1-800 Hz with a resolution of 0.2 Hz, i.e., amplitude values at 3996 frequencies. When features from all 3 directions are augmented together, the new feature dimension is 11988. Hence, the training dataset after outlier removal and frequency domain extraction has 65563 samples and 11988 features and the testing data comprises of 16388 samples and 11988 features.

4.5.1. Modeling using PLS on frequency domain features

Even though the feature dimension is very high, PLS is used, as it is a robust approach for feature dimension reduction and can deal with multicollinearity inherently. As mentioned earlier, again the training set is split into calibration and validation to train the hyperparameters, i.e., the number of PCs.

Shown in Figure 4.10 below is the performance of PLS for prediction of RPM using Frequency domain data. The RMSE for the test set is 11.35 and 80 PCs were selected based on Cross validation. This is a huge improvement over modeling using Raw data in time domain. This shows how frequency domain analysis can help establish a meaningful relationship to predict the RPM. The results demonstrate the RPM range is predicted well and the 10 different levels are predicted well with no or very few test samples predicted out of the RPM class range. This shows the robustness of our proposed approach.

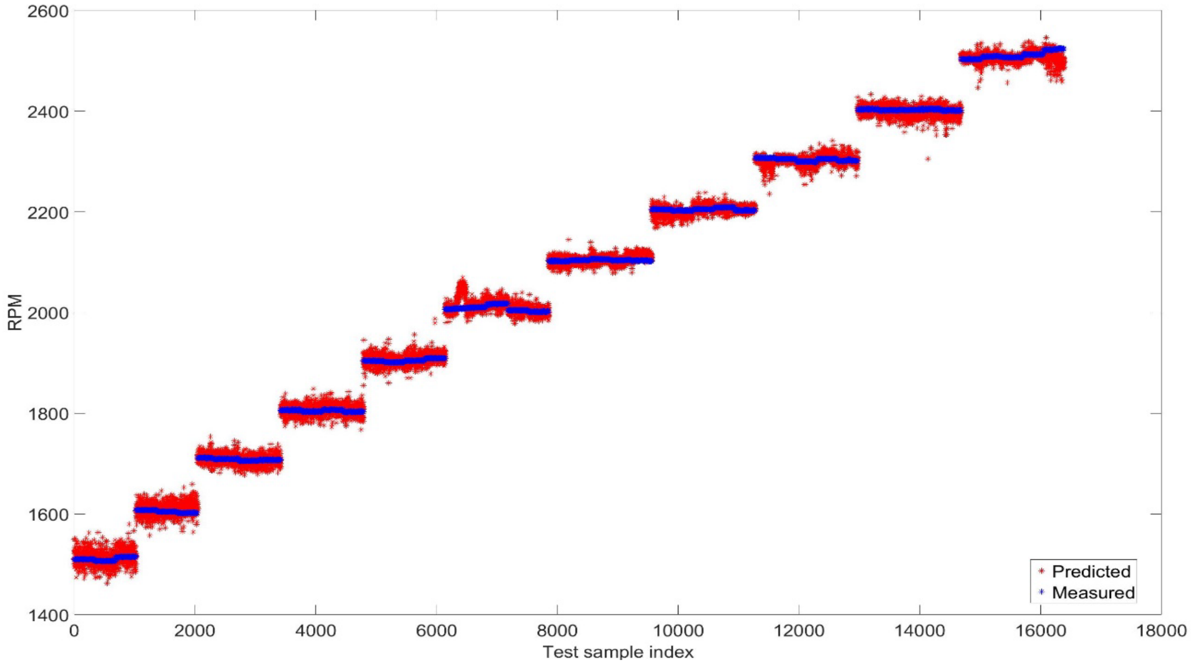


Figure 4.10 Full PLS model RPM prediction performance (based on frequency domain)

Similarly, Figure 4.11 shows the performance of PLS in predicting the flow rate using one single model for all flow rate conditions. It is worth noting that for different RPM conditions, data has been collected for the same flow rate levels making the prediction significantly challenging. As mentioned earlier, the goal of our approach is to create a model that can predict the flow rate and RPM with the least error possible. For the features based on the frequency spectrum, the RMSE of the test set is 0.73, which is a significant improvement when raw data was used. However, the predictions are still poor, and the error in predicting flow rates is relatively higher.

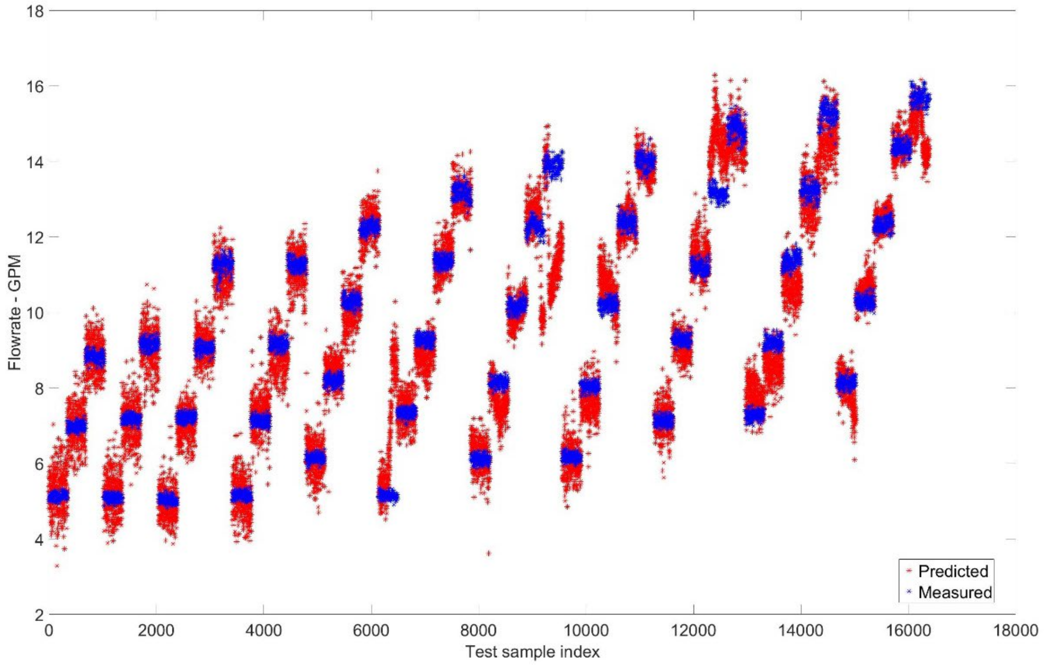


Figure 4.11 Full PLS model Flowrate - GPM predicition performance (Based on frequency domain)

4.6. System engineering enhanced modeling approach

In the previous work[109], taking advantage of the system knowledge, physics, pump operations, and data modeling techniques, a binary matrix approach was proposed to accurately identify pump RPM. Our work incorporates the binary matrix approach along with domain knowledge-based time domain and frequency domain features, which are used to predict the RPM and flow rate with high accuracy using a single set of features.

4.6.1. Binary matrix approach – Select peak frequencies

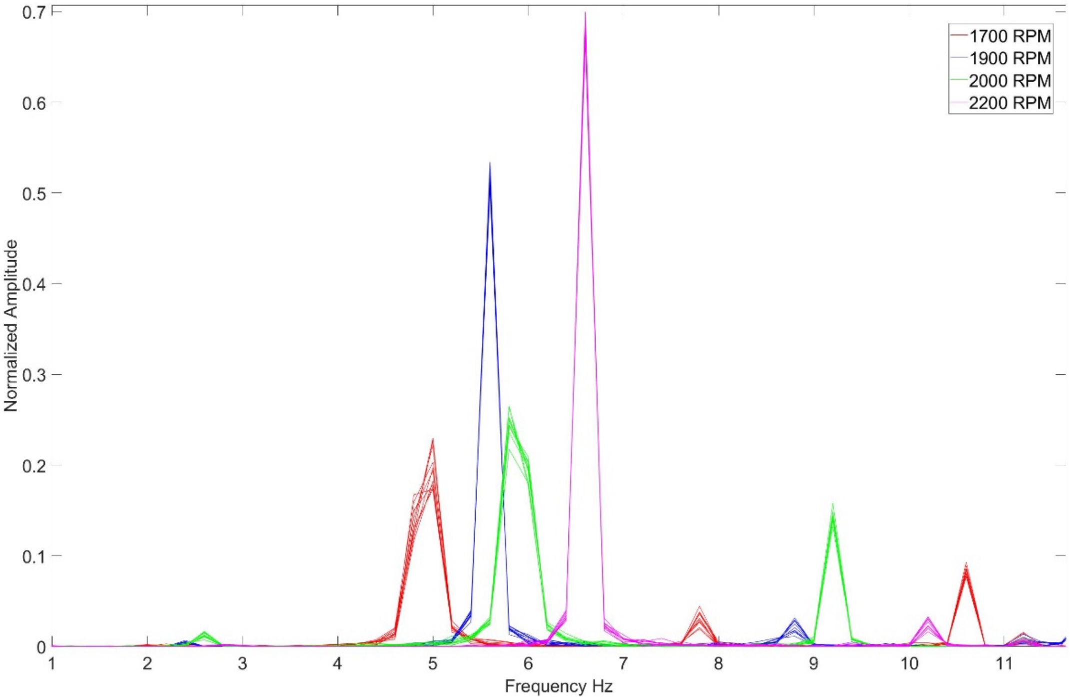


Figure 4.12 Peaks occur at fixed frequencies for a fixed RPM condition

The binary matrix approach is based on the idea of identifying the max frequencies that are related to corresponding RPM conditions. Figure 4.12 shows how RPM stages are linked to distinct frequencies, and there is a relation between them.

We have plotted several data samples for 4 different RPM conditions. As seen above, for a fixed condition, the samples are related to frequencies. Thus, peak frequencies can be utilized well for predicting the accurate RPM. Also, most of these peaks occur at low frequencies, and thus all for our binary matrix approach, only the spectrum from 1-100 Hz is considered to extract the peak frequencies and amplitude.

In the binary matrix approach, we use a zero vector of length 1x500, and instead of using the peak frequency directly, we in our work replace the index related to the frequency having the highest amplitude as 1. 10 variables, 5 on either side are also replaced with 1s to reduce the effect of noise. This procedure is carried out for all samples to form the binary feature matrix with 0s and 1s. It was observed that information from the X and Z direction was more relevant, and hence Y direction information was not considered in this work.

4.6.2. Analysis of flow rate at same RPM conditions

Figure 4.13 shows how the amplitude corresponding to peak frequencies shows different behavior for the same RPM but different Flowrate conditions. Keeping this in mind, we include corresponding Amplitudes for the peak frequency as well as features in predicting the RPM and flow rate. It is worth mentioning that prediction of Flow rate is extremely challenging, but RPM is explicitly related to the peak frequencies. Also, for the different RPM conditions, data is

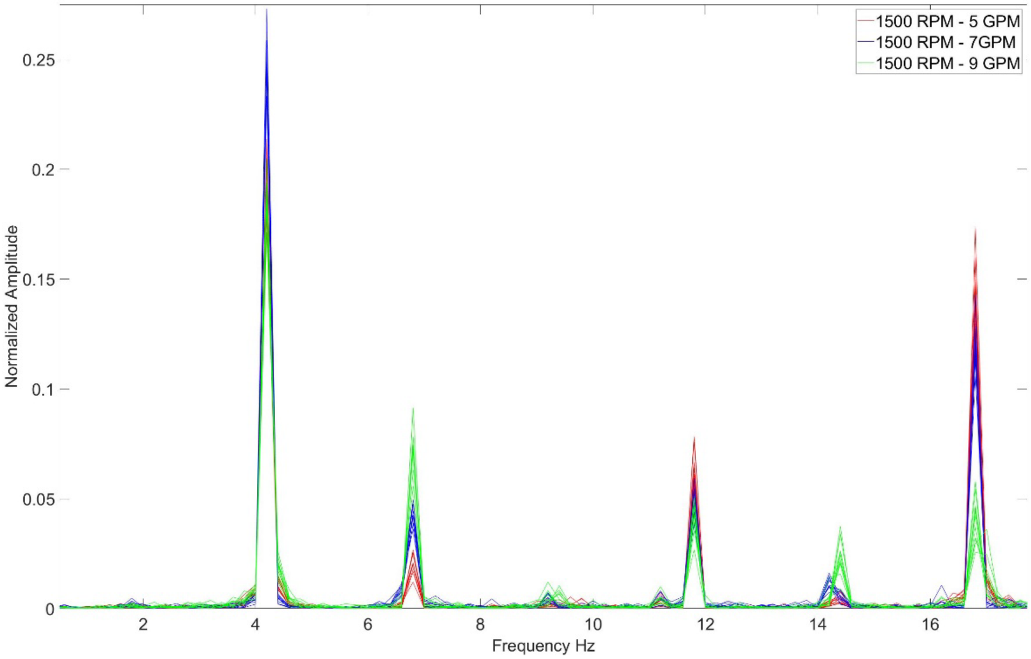


Figure 4.13Amplitude for different flowrate but same RPM conditions

collected for similar flow rate ranges as well and a robust model would be able to distinguish these conditions clearly.

4.6.3. Use of coefficient of variation to select features

The coefficient of variation (CV) is a statistical measure of dispersion of data points in a data series around mean. It represents the ratio of standard deviation to the mean and is a useful statistic for comparing the degree of variation.

Coefficient of variation is as follows:

$$C_V(x) = \frac{s(x)}{\mu(x)} \qquad (4.4)$$

As CV shows the extent of variability of data in a population, we in our work use CV to filter out the features extracted after Lomb-Scargle algorithm. The hypothesis is that the higher the CV the higher the dispersion and the higher the variability explained by any frequency within the spectrum 1-100Hz. We use a threshold of 3 that is set empirically using training data set to filter out those frequencies that have a higher explained variability and use it to predict the RPM and flow rate as well. Following this procedure, 16 features are selected from the X direction, and 32 features/frequencies related data are selected from the Z direction. These features are augmented to the binary matrix and the peak frequencies and amplitude data.

4.6.4. Use of domain knowledge-based features in the time domain

Numerous statistical functions can be used for monitoring vibration data. We, in our work, use certain features in the time domain that can be used to predict the RPM and flow rate of the system. Thus, our proposed approach is a fusion of time domain as well as feature domain feature using system knowledge. Features based on vibration signals are statistical metrics, and changes in these features can indicate the status of a system. Basic statistics include mean, standard deviation, root mean square (RMS), and shape factor. In addition, we use higher-order statistics due to the fact that they provide an insight into the system behavior through the fourth moment, i.e., kurtosis and third moment i.e., skewness of the vibration signal. These statistical features have been used in literature for fault detection and condition monitoring, but this is the first attempt to predict the RPM and flow rate using accelerometer vibration signals to the best of authors knowledge. Impulsive metrics such as peak value, impulse factor, crest factor and clearance factor are also incorporated. Impulsive metrics define properties of the vibration signal related to the peaks of the signal. Last but not the least, signal-processing metrics that consist of distortion measurement functions have been used in our work as well. The hypothesis behind using signal processing metrics is that abrupt or changes in system behavior can cause an increase in noise, changes in harmonics relative to fundamentals or both at once. Signal-to-noise ratio (SNR), total harmonic distortion (THD) and signal to noise and distortion ratio (SINAD) are the signal processing metrics used in our work. A detailed description of each of these features can be found below.

Shape factor: It is the ratio of RMS divided by the mean of the absolute value. Shape factor is dependent on the signal shape and does not depend on the signal dimensions.

$$X_{SF} = \frac{X_{rms}}{\frac{1}{N}\sum_{i=1}^{N}|X_i|} \qquad (4.5)$$

Kurtosis: Defines how outlier prone the signal is. The changes in the RPM directly affect the vibration signals and therefore can lead to an increase in the kurtosis metric.

$$X_{kurt} = \frac{\frac{1}{N}\sum_{i=1}^{N}|X_i - \bar{x}|^4}{\frac{1}{N}\sum_{i=1}^{N}|X_i - \bar{x}|^{2^{2}}} \qquad (4.6)$$

Skewness: Defines the asymmetry of a signal distribution. Changes in RPM and flowrate can impact the distribution symmetry.

$$X_{skew} = \frac{\frac{1}{N}\sum_{i=1}^{N}|X_i - \bar{x}|^3}{\left[\frac{1}{N}\sum_{i=1}^{N}|X_i - \bar{x}|^{2^{3/2}}\right]} \qquad (4.7)$$

Peak value: Maximum absolute value of a signal and is used to calculate other Impulse metrics.

$$X_p = \max |X_i| \qquad (4.8)$$

Impulse factor: Compare the height of the peak to the mean level of the signal.

$$X_{IF} = \frac{X_p}{\frac{1}{N}\sum_{i=1}^{N}|X_i|} \qquad (4.9)$$

Crest factor: Peak value divided by RMS. The crest factor can represent changes in the system.

$$X_{crest} = \frac{X_p}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}X_i^2}} \qquad (4.10)$$

Clearance factor: Peak value divided by the squared mean value of the square roots of absolute amplitudes.

$$X_{clear} = \frac{X_p}{\frac{1}{N}\sum_{i=1}^{N}\sqrt{|X_i|}^2} \qquad (4.11)$$

Signal-to-noise ratio (SNR): Ratio of signal power to noise power.

Total harmonic distortion (THD): Ratio of total harmonic component power to fundamental power.

Signal to noise and distortion ratio (SINAD): Ratio of total signal power to total noise plus distortion power.

4.6.5. Modeling based on systems engineering enhanced features

After considering all the features together, we have 244 features in total that are used to build our model that can predict the RPM and flow rate. The relationship between RPM and explanatory features is defined well by Partial least squares approach as mentioned in the previous section. Also, keeping the interpretability of models in mind we use PLS to predict the RPM based on system engineering-enhanced features. The 244 features include the features mentioned in the above sections incorporating the binary matrix, coefficient of variation and features based on vibration signals in the X and Z direction.

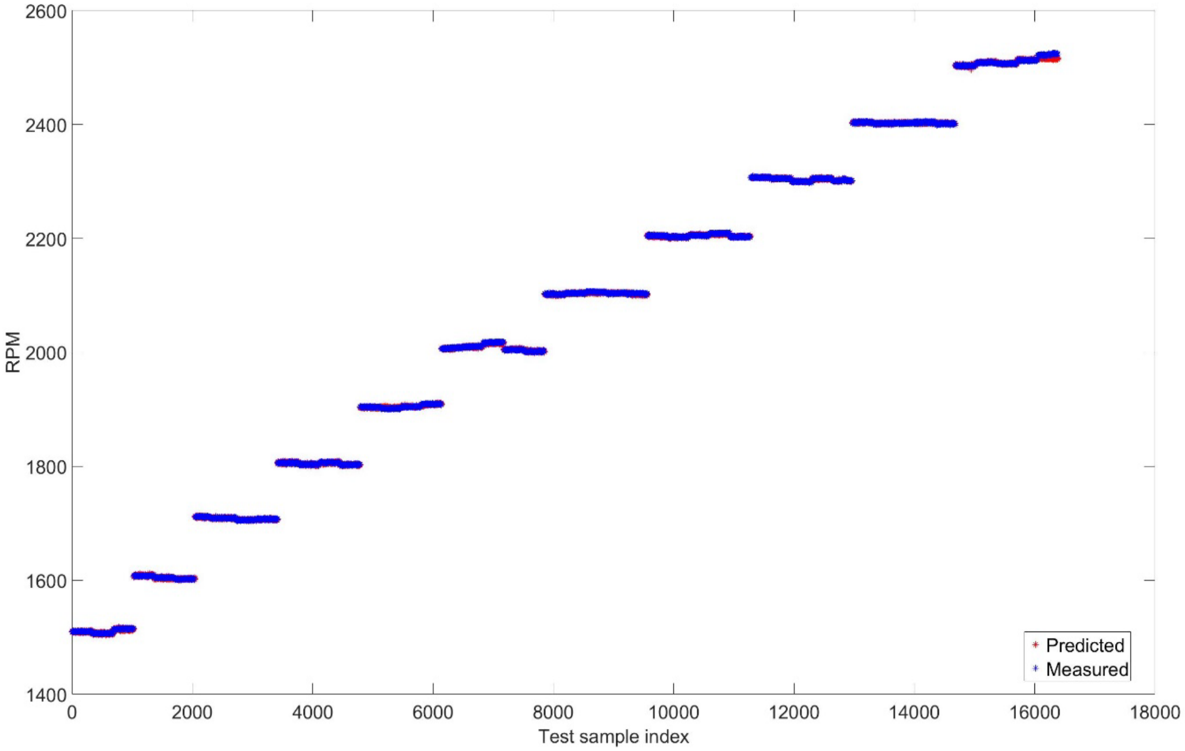


Figure 4.14 RPM predictions based on Systems engineering enhanced features -PLS

Figure 4.14 below shows the prediction of RPM based on PLS model. As seen the predictions are extremely accurate and can identify RPM correctly based on the proposed features. 20 PCs were chosen based on cross-validation. The dimensions of training data are 65563x244 and the dimensions of the test data are 16388x244. The RMSE of prediction for RPM dropped from 11.35 to 1.52. This is a huge gain considering the new set of features, and as seen in Figure 4.14 above, the predictions are extremely accurate. To sum it up, the RMSE of predictions based on raw data was 190.97. After extracting all the data from raw vibration data into the frequency domain using Lomb's Scargle algorithm, the RMSE dropped to 11.35, but the feature space was 11988. Further, after systems engineering enhanced feature engineering, the RMSE dropped to 1.53, and 244 features are used. This clearly shows how feature engineering plays a key role in developing successful data-driven ML models for predicting key process information in this IIoT testbed.

Figure 4.16 below shows the prediction for flow rate using PLS model. As seen, PLS fails to capture the relationship between the predictors and flow rate. The RMSE is 0.60 which is an improvement over the case where 11988 features extracted from Lomb-Scargle algorithm were used. This demonstrates that knowledge-based feature engineering aids the prediction of RPM and flow rate. However, the relationship is not linear for flow rate and hence linear models such as PLS fails to capture the relationship. This requires the need of a model that can capture a non-linear relationship well. The use of kernel versions of PLS or other models such as k-neighbors regression can be considered for flow rate predictions.

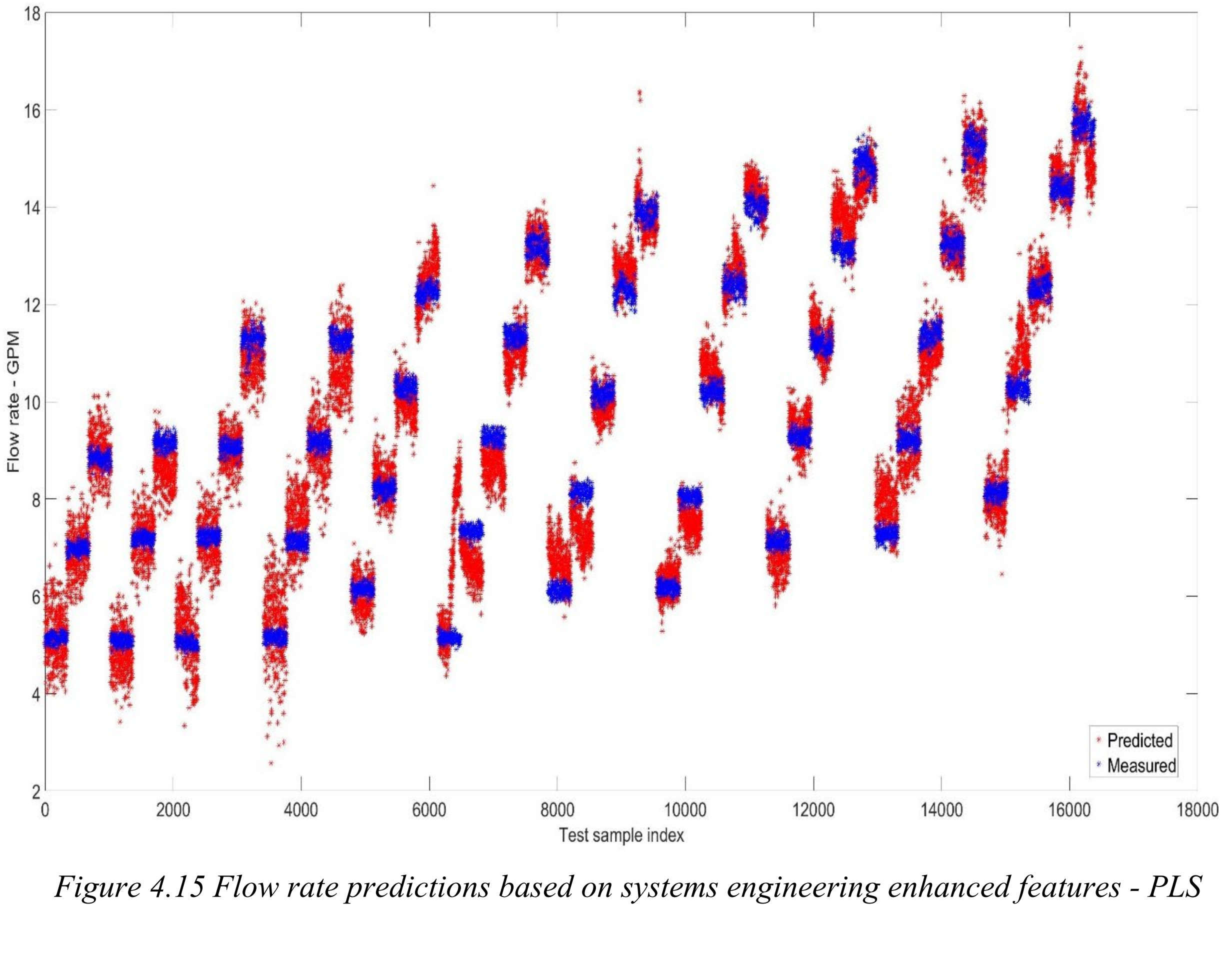


Figure 4.15 Flow rate predictions based on systems engineering enhanced features - PLS

As mentioned earlier, the modeling of flowrate is extremely challenging in comparison to the modeling of RPM. While the PLS model captures the linear relationship between the predictors and RPM well, it fails to do so for the prediction of flow rates. To model this complex relationship between the explanatory variables and flow rate, the k-neighbors regression is used in our work. Regression using this ML algorithm is based on nearest neighbors, and the target is predicted by local interpolation of targets associated with the nearest neighbors in the training set. Nearest neighbors' regression can be used in cases similar to this work where the data labels are continuous rather than discrete variables. The label assigned to a query point is computed based on the mean of the labels of its nearest neighbors. The implementation of k-neighbors regression is performed in scikit-learn [53]. The parameters under consideration to tune the performance using cross-validation are the number of neighbors, the algorithm used to compute the nearest neighbors, and the weights to decide the weight function used in prediction. A combination of random search and Bayesian optimization is used to tune the hyperparameters in our work.

It is worth noting that the sampling frequency of measurements was 3 Hz. That means 3 samples are collected every second. Generally, in a process industry, a prediction horizon over a couple of seconds is considered to predict the properties. In our work, we consider a span of 10 seconds to predict the flow rate of the system. We divided the 10-minute data for each condition to training and testing using a 80-20 split i.e., approximately 8 minutes of data is used for training while the last 2 minutes of data are used for testing. For our predictions, we consider an aggregation (e.g., average or median) of multiple measurements or random sampling with multiple instances and consider the aggregate. The flow rate is assumed to be constant over a period of 10 seconds in a industrial scenario. Thus, this is equivalent to considering multiple measurements within a time window and considering an aggregate of those sampling instances. We use median in our work, as an aggregation function due to the fact that mean is relatively more sensitive to some faults or abrupt changes in the system. Our predictions should be less sensitive to those abrupt changes as such changes or spikes generally represent some fault reading or outlier. If it prevails for a longer time, it does indicate a fault in the system.

Hence for our work, we divide the predictions into equal 10 second chunks considering them as multiple sampling instances with the hypothesis that the system fluctuation is minimal in that time span. We report the median of such multiple test samples. Thus, for each condition i.e., a fixed RPM and GPM, we have multiple test samples. The predictions of flow rate based on K-neighbors regression are shown in the Figure 4.16 below. 10 neighbors are selected based on

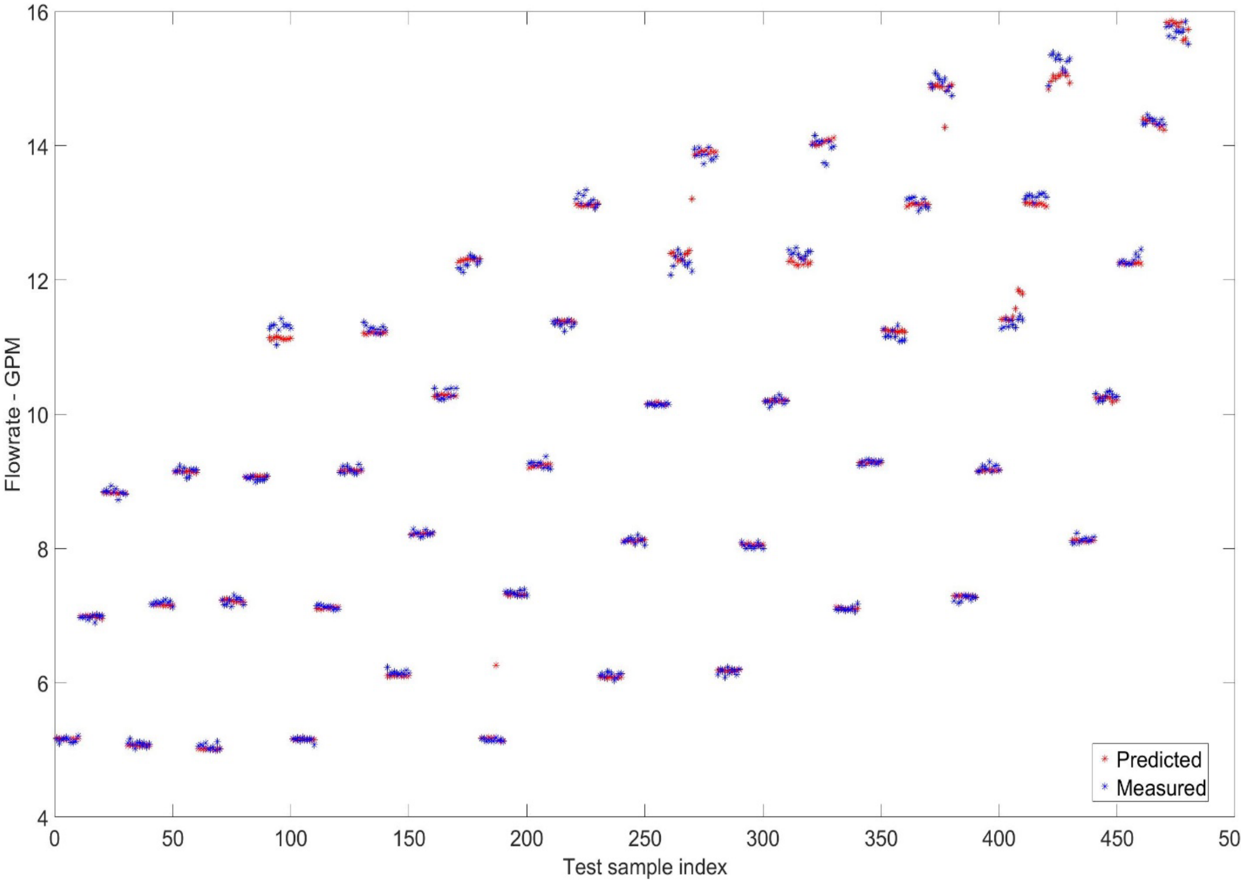


Figure 4.16 Flowrate GPM predictions based on systems engineering enhanced features – k-neighbors regression

hyperparameter optimization. The RMSE of prediction is 0.12, which is a significant improvement over the previous predictions based on raw data and the data based on Lomb-Scargle algorithm. Thus, even with a single model for flowrate, we were able to predict the flowrate with minimal error even though the modeling for flowrate is extremely challenging due to the fact that different RPM levels have the same flowrate values, which is quite common in the process industry.

Similar to k-neighbors regression predictions over 10 second interval, the predictions from PLS are also shown in Figure 4.17 for a fair comparison. There is a clear difference between the predictions in Figure 4.16 and Figure 4.17. The median predictions over 10 second interval was poor for PLS and deviate a lot from their actual measured flow rate in comparison to the predictions from K-neighbors regression. The RMSE for K-neighbors regression is 0.12 while the RMSE for PLS predictions is 0.45, which is substantially higher than the former one.

It is worth noting here that Deep learning (DL) models are known to capture non-linear and complex relationships well, given ample data and enough computation power. These two conditions are satisfied in our work, but the interpretation of DL models is extremely difficult, and relatively simpler models are more desirable in process industries for interpretation.

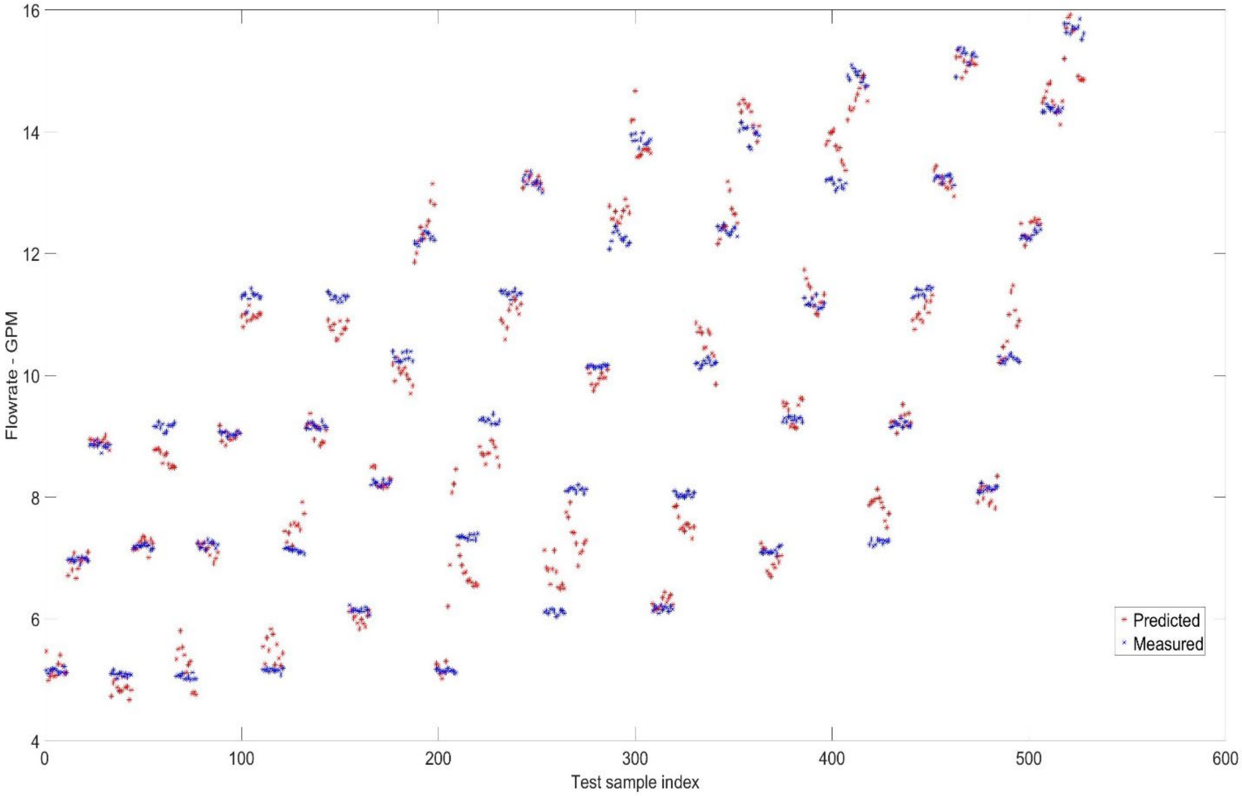


Figure 4.17 Flowrate GPM predictions based on systems engineering enhanced features – PLS regression

Thus, it is again demonstrated through flowrate predictions that knowledge-based feature engineering combined with ML plays a key role in the prediction of process properties in comparison to rote application of ML models.

4.7. Conclusions

Table 4.2 compares the performance of RPM and flow rate at different levels of feature engineering. The results show that feature engineering is the key to successful modeling of RPM and flow rate for data collected from this IIoT testbed. In the first step, there is a significant improvement when considering the vibration data in the frequency domain in comparison to when raw vibration data in the time domain is used. Further, we extract relevant information using domain-based feature engineering. The RPM at different conditions have a strong relationship with different frequencies which is expected since at different RPM levels, the vibrations will be completely different leading to peaks at different frequencies which is shown in Section 4.6.1. In addition, to the best of authors' knowledge, this is the first attempt to avoid a hierarchical approach for prediction of flow rate. A fusion of features extracted in the time domain and knowledge-guided features in the frequency domain are used to enhance the performance further. It is worth noting that with RPM values distributed in such a wide range, an RMSE of 1.53 with knowledge guided feature engineering is quite accurate. For flow rate, the modeling is challenging given the complex relationship and no clear trend. Thus, linear models such as PLS fail to predict the flow rate accurately and we use k-neighbors regression to predict the flow rate. The comparison between the performance of PLS and k-neighbors regression is shown in Table 4.3. The RMSE for PLS is substantially higher than k-Neighbors regression. As shown in Figure 4.16, the predictions significantly improve and capture the trend well with few exceptions while in Figure 4.17, the predictions deviate significantly from their measured flow rate values.

The use of neural network models such as ANN and DNN can be considered for future work. They are known to capture nonlinear and complex relationship well, but the interpretation remains and issue, when using these models in industrial systems.

Table 4.2 Predictions performance of RPM and flow rate at different levels of feature engineering

Features used	Dimension of features	RMSE - RPM	RMSE – Flow rate
Raw vibration data – Time series	2400	190.97	2.94
Lomb-Scargle algorithm – Frequency domain data	11988	11.35	0.73
System engineering enhanced features – Fusion of features in time domain plus frequency domain	244	1.53	0.60

Table 4.3 Prediction performance comparison for flow rate. PLS regression vs K- neighbors regression with system engineering enhanced feature engineering

Features used	Dimension of features	RMSE –flow rate PLS regression	RMSE – flow rate K-Neighbors regression
System engineering enhanced features – Fusion of features in time domain plus frequency domain – Average of samples collected over 10 seconds	244	0.45	0.12

5. Chapter 5. Summary and future work

The dissertation addresses several existing challenges in a variety of industrial processes to move towards advanced manufacturing. The major contribution areas include woodchip moisture content estimation using IIoT Wi-Fi sensors and ML for the pulp and paper industry; feature-based virtual metrology framework for the semiconductor industry; and process modeling and condition monitoring using IIoT vibration sensors and ML for process industries.

5.1. Summary

In this work, the author aims to explore the utilization of novel non-invasive IIoT sensors, including accelerometers and 5G Wi-Fi, for industrial applications. The author also investigates the challenges and limitations in ML modeling when the traditional pure data-driven ML techniques are directly applied to model the data collected from IIoT sensors. The author shows how modeling big data directly with data-driven ML techniques lead to incomplete or misleading information and insights. The author investigates how extracting and processing the information within the data collected from IIoT sensors with the right tools is more critical to the data-driven decision-making process to enhance and optimize operations in an industrial setting.

5.1.1. Moisture detection in woodchips using IIoT Wi-Fi and ML techniques

The US pulp and paper industry is an energy intensive sector, and the pulping process is one of the most critical process with substantial room for improvement. For the pulping process, the incoming woodchip MC is a significant source of disturbance, and this disturbance is unmeasured due to the lack of affordable, reliable and easy to maintain sensors, which leads to significant loss in pulp yield, overuse of heat, energy and chemicals. As a solution, the author proposes a non-destructive, economic, and robust woodchip MC sensing approach using CSI from IIoT based Wi-Fi sensors.

An experimental design and an algorithmic technique were proposed to handle the confounding factors. Specifically, to address the challenge that raw CSI data are very noisy and sensitive to woodchip packing, the author proposes a feature-based classification system based on SPA. The author investigates two different aspects of CSI data and shows how amplitude is critical to smart sensing, while phase difference of CSI data is less reliable given the heterogeneous woodchip structure. Effects of diversity in a multi-antenna receiving system are also investigated and it is shown that information utilized from more than one antenna on the receiving side aid the MC estimation in woodchips. The drawbacks of modeling on raw data are discussed. Through detailed investigation and robust modeling, the author shows that the key to smart sensing is the synergistic integration of domain knowledge and ML techniques. Different linear and non-linear ML classification techniques are reviewed, and their performances are also compared in this work.

5.1.2. VM for semiconductor manufacturing

In semiconductor manufacturing, a wafer undergoes hundreds of steps to yield the final product. After a processing step, typically, a few wafers from a lot are measured at the metrology station and they represent the whole lot. In this work, the author proposes a novel next-generation feature-based virtual metrology (FVM) framework, to address the challenges and limitations of existing VM techniques in the semiconductor industry. FVM also targets wafer-to-wafer control, which is replacing lot-to-lot process control at an increasing pace. It is shown in this work how an efficient approach for predicting wafer properties without physically conducting measurement can reduce costs and downtime and increase the overall efficiency of the process.

The author provides a detailed description of the proposed FVM approach in this work. It is shown in detail how the proposed FVM approach can eliminate some of the data pre-processing steps such as data mismatch, trajectory shift and alignment inherently. These pre-processing steps are a common issue in modeling batch processes such as a typical semiconductor manufacturing process. The author demonstrates how different statistics can be used to achieve superior performance and can capture process characteristics including non-linearity and non-Gaussianity. The superior performance of FVM based approach is evaluated based on two case studies i.e., a simulated CMP dataset and a real industrial plasma etch dataset. Both static and recursive modeling approaches are explored and investigated to mimic the industrial scenario..

5.1.3. Machine learning techniques for process modeling and condition monitoring using non-invasive IIoT vibration sensors

A chemical process of manufacturing plant can be considered a warehouse of data where many process measurements are collected and stored every second. However, often, these measurements are not available due to the lack of reliable sensors or due to the nature of the process. Incorporation of IIoT sensors along with robust analytics has a potential to mitigate such problems and enable smarter manufacturing processes. This work focuses on the use of non-invasive IIoT sensors such as accelerometers or vibration sensors, for predictive modeling and condition monitoring in the process industry.

This work is an extension to a previous work and the major focus of this work is to accurately predict the motor speed and the flow rate of fluid in the system using vibration data collected through non-invasive IIoT sensors. Data collected from a centrifugal pump testbed with multiple IIoT sensors are used in this work for predictive modeling of RPM and flow rate. The performance of ML techniques such as PLS on unprocessed data are shown and investigated. The author demonstrates how rote application of ML fails to capture the relationship between explanatory and response variables. To address the poor performance, the author proposes feature engineering to extract features relevant to the peak frequency in the frequency domain and the time domain. Different levels of feature engineering and their performance based on ML approaches such as PLS and k-neighbors regression are compared. The author also shows, how modeling the flow rate is challenging and linear approaches such as PLS fail. As a solution, the author proposes the use of complex approaches such as k-neighbors regression to accurately predict the flow rate of the system.

5.2. Potential directions for future work

In this section, the author sheds some light on potential future directions in the areas of this research and thus enhance these proposed approaches further.

5.2.1. Moisture detection in woodchips using IIoT Wi-Fi and ML techniques

It is worth noting that although woodchip packing has a significant impact on the collected CSI data (both amplitude and phase responses), its impact on MC classification is eliminated after we selected SPA features that are completely insensitive to packing. Although the randomization is done by shaking the same woodchips within a given volume - which means the volume density of the sample is about the same, the linear density (*i.e.*, linear void/packing fraction) varies significantly. If we assume linear paths of the Wi-Fi signal propagation, shuffling even the same woodchips can introduce significant variations to the linear void (or packing) fraction along the straight lines between the injector and the three receivers, as evidenced by the significant changes in the amplitude and phase responses of the CSI data. However, our results show that the selected

SPA features (*i.e.*, mean difference of consecutive subcarriers of CSI amplitude) are insensitive to the shuffling, as evidenced by the high classification accuracy of independent (*i.e.*, differently shuffled) testing samples. Therefore, we can conclude that the selected SPA features are insensitive to the void fraction (or packing density) of the woodchips. This is particularly convincing when we consider the excellent performance of the technology at the low MC range where there is only 0.05% change in MC level but significant change in linear void fraction along the Wi-Fi propagation paths due to shuffling. Nevertheless, it is desirable to test woodchips with different sizes to further validate the technology. We envision that, when implemented in real industrial applications, some form of random sampling can be implemented to obtain multiple MC estimations, and some form of aggregation (*e.g.*, average) of different measurements can be used to obtain a reliable estimation of the MC level for a large number of woodchips.

It is also worth noting that this work only establishes the feasibility of this technology in the lab using a box. Whether the technology can be applied in more flexible settings, such as woodchips not in a box but in a pile on a fixed or moving surface (*e.g.*, a conveyor belt), requires further investigation. There is no doubt that the problem will be more challenging than what has been studied in this work, which is under a much better controlled environment in a lab. In addition, this work only demonstrates the success of classification-based woodchip MC estimation, while the preliminary results have shown that the regression-based MC estimation is much more challenging for this application. This is due to the fact that, although MDCSs of CSI amplitude enables linear separation of different MC levels, the functional relationship between CSI data and woodchip MC values is actually much more complicated and research in this area is a potential suggested area worth investigating.

5.2.2. VM for semiconductor manufacturing

It is worth noting that there are other nonlinear VM approaches as discussed in this work such as kernel-based or ANN-based approaches. Although FVM uses features, they are different from features extracted using kernel-based methods because they have clear physical and/or statistical meanings. Similarly, although ANN-based methods can extract the nonlinear relationship between independent and dependent variables, the interpretation of the relationship is challenging. In other words, it is difficult if not impossible to find which variable/feature contributes how much to the output. In addition, the author suggests the use certain feature selection approaches to further explore each of the proposed features and their contribution to superior prediction performance as a part of future work.

These are also areas of FVM and RFVM that are worth further investigation, such as different recursive or adaptive schemes to further improve their performances and are a potential direction of research.

5.2.3. Machine learning techniques for process modeling and condition monitoring using non-invasive IIoT vibration sensors

The major goal and contribution of this work are to predict the flow rate and RPM by investigating a set of feature engineering approaches to improve the modeling of the relationship between vibration data collected from IIoT sensors and flow rate and RPM. In contrast to a hierarchical approach proposed in the previous work to predict the flow rate from each RPM stage, a single model is proposed in this work to predict the RPM with minimal error using a fusion of time domain and frequency domain-based features selected using domain knowledge. While the relationship between vibration data and RPM is linear and can be modeled with methods such as

PLS, modeling the relationship with flow rate is challenging, and complex models such as k-neighbors regression are used to achieve the required prediction performance.

This work demonstrates that feature engineering, once again, plays a key role in developing successful data-driven machine learning models for predicting key process information, and rote application of ML algorithms without considering domain knowledge leads to poor predictions. The performance is compared at different levels or extent of feature engineering. It is seen that statistical approaches combined with feature engineering that involve extensive human learning showed superior performance. However, Deep Neural Networks (DNN) or Deep Learning (DL) approaches are known to model complex relationships relatively well and have been researched extensively in recent years. However, these models require high computation and relatively more data to model the relationship efficiently. Although the interpretability of DNN or DL approaches remains an issue, this area of research is a potential direction to investigate further. The author believes that there is undeniable potential for DL in process systems engineering applications and is worth exploring.

Systematic feature selection on the systems engineering enhanced features proposed in this work is another suggested research direction for the future. Different feature selection approaches can be used to select a subset of features and can potentially improve the process modeling and monitoring performance further.

6. Bibliography

[1] K. Ashton, “That Internet of Things Thing,” *RFID J.*, 2009.

[2] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Comput. Networks*, 2010.

[3] P. Deshpande, “Predictive and prescriptive analytics in big data Era,” in *Advances in Intelligent Systems and Computing*, vol. 810, 2018.

[4] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, “The industrial internet of things (IIoT): An analysis framework,” *Comput. Ind.*, 2018.

[5] J. Conway, “The Industrial Internet of Things: An Evolution to a Smart Manufacturing Enterprise,” *Schneider Electr.*, 2016.

[6] Y. Chen, “Integrated and Intelligent Manufacturing: Perspectives and Enablers,” *Engineering*, 2017.

[7] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, “Intelligent Manufacturing in the Context of Industry 4.0: A Review,” *Engineering*, vol. 3, no. 5, 2017.

[8] C. Arnold, D. Kiel, and K. I. Voigt, “How the industrial internet of things changes business models in different manufacturing industries,” in *International Journal of Innovation Management*, 2016.

[9] J. Cheng, W. Chen, F. Tao, and C. L. Lin, “Industrial IoT in 5G environment towards smart manufacturing,” *J. Ind. Inf. Integr.*, vol. 10, 2018.

[10] A. Kusiak, “Smart manufacturing,” *Int. J. Prod. Res.*, 2018.

[11] F. Tao, Q. Qi, A. Liu, and A. Kusiak, “Data-driven smart manufacturing,” *J. Manuf. Syst.*, 2018.

[12] H. S. Kang *et al.*, “Smart manufacturing: Past research, present findings, and future directions,” *Int. J. Precis. Eng. Manuf. - Green Technol.*, 2016.

[13] P. Zheng *et al.*, “Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives,” *Frontiers of Mechanical Engineering*. 2018.

[14] K. J. Kramer, E. Masanet, T. Xu, and E. Worrell, “Energy efficiency improvement and cost saving opportunities for the pulp and paper industry,” in *Improving Energy Efficiency and Greenhouse Gas Reduction in the Pulp and Paper Industry*, Nova Science Publishers, Inc., 2011.

[15] N. Martin, N. Anglani, D. Einstein, M. Khrushch, E. Worrell, and L. K. Price, “Opportunities to improve energy efficiency and reduce greenhouse gas emissions in the US pulp and paper industry,” *Lawrence Berkeley Natl. Lab.*, 2000.

[16] S. Brueske, C. Kramer, and A. Fisher, “Bandwidth Study on Energy Use and Potential Energy Saving Opportunities in US Pulp and Paper Manufacturing,” Energetics, 2015.

[17] M. Rahman, A. Avelin, and K. Kyprianidis, “A review on the modeling, control and diagnostics of continuous pulp digesters,” *Processes*, vol. 8, no. 10, pp. 1–26, 2020.

[18] H.-K. Choi and J. S.-I. Kwon, “Modeling and control of cell wall thickness in batch delignification,” *Comput. Chem. Eng.*, vol. 128, pp. 512–523, 2019.

[19] H. Choi and J. S. Kwon, “Multiscale modeling and control of Kappa number and porosity in a batch-type pulp digester,” *AIChE J.*, vol. 65, no. 6, p. e16589, 2019.

[20] H.-K. Choi, S. H. Son, and J. Sang-Il Kwon, "Inferential Model Predictive Control of Continuous Pulping under Grade Transition," *Ind. Eng. Chem. Res.*, vol. 60, no. 9, pp. 3699–3710, 2021.

[21] ASTM, "Standard test methods for moisture content of wood ASTM D4442," *1983 Annu. B. ASTM Stand.*, no. November, pp. 431–445, 2016.

[22] J. Reeb and M. Milota, "Moisture Content by the Oven-Dry Method for Industrial Testing," *WDKA*, pp. 66–74, 1999.

[23] H. Daassi-Gnaba, Y. Oussar, M. Merlan, T. Ditchi, E. Géron, and S. Holé, "Moisture content recognition for wood chips in pile using supervised classification," *Wood Sci. Technol.*, vol. 52, no. 5, pp. 1195–1211, 2018.

[24] H. Daassi-Gnaba, Y. Oussar, M. Merlan, T. Ditchi, E. Géron, and S. Holé, "Wood moisture content prediction using feature selection techniques and a kernel method," *Neurocomputing*, vol. 237, pp. 79–91, 2017.

[25] P. Pan, T. McDonald, J. Fulton, B. Via, and J. Hung, "Simultaneous moisture content and mass flow measurements in wood chip flows using coupled dielectric and impact sensors," *Sensors*, vol. 17, no. 1, p. 20, 2017.

[26] L. Fridh, L. Eliasson, and D. Bergström, "Precision and accuracy in moisture content determination of wood fuel chips using a handheld electric capacitance moisture meter," *Silva Fenn.*, vol. 52, no. 5, 2018.

[27] P. Pan, T. P. McDonald, B. K. Via, J. P. Fulton, and J. Y. Hung, "Predicting moisture content of chipped pine samples with a multi-electrode capacitance sensor," *Biosyst. Eng.*,

2016.

[28] M. Merlan, T. Ditchi, Y. Oussar, S. Holé, E. Géron, and J. Lucas, “Resonant half-wave antenna for moisture content assessment in wood chips,” *Meas. Sci. Technol.*, 2019.

[29] E. A. Amaral, L. M. Santos, E. V. S. Costa, P. F. Trugilho, and P. R. G. Hein, “Estimation of moisture in wood chips by near infrared spectroscopy,” *Maderas Cienc. y Tecnol.*, 2020.

[30] L. Liang, G. Fang, Y. Deng, Z. Xiong, and T. Wu, “Determination of Moisture Content and Basic Density of Poplar Wood Chips under Various Moisture Conditions by Near-Infrared Spectroscopy,” *For. Sci.*, 2019.

[31] M. Hultnäs and V. Fernandez-Cano, “Determination of the moisture content in wood chips of Scots pine and Norway spruce using Mantex Desktop Scanner based on dual energy X-ray absorptiometry,” *J. Wood Sci.*, 2012.

[32] J. Couceiro, O. Lindgren, L. Hansson, O. Söderström, and D. Sandberg, “Real-time wood moisture-content determination using dual-energy X-ray computed tomography scanning,” *Wood Mater. Sci. Eng.*, 2019.

[33] P. Hu, W. Yang, X. Wang, and S. Mao, “MiFi: Device-free wheat mildew detection using off-the-shelf wifi devices,” in *2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings*, 2019.

[34] W. Yang, X. Wang, S. Cao, H. Wang, and S. Mao, “Multi-class wheat moisture detection with 5GHz Wi-Fi: A deep LSTM approach,” in *Proceedings - International Conference on Computer Communications and Networks, ICCCN*, 2018.

[35] W. Yang, X. Wang, A. Song, and S. Mao, “Wi-Wheat: Contact-Free Wheat Moisture

Detection with Commodity WiFi," in *IEEE International Conference on Communications*, 2018.

[36] D. Shah, J. Wang, and Q. P. He, "An Internet-of-things Enabled Smart Manufacturing Testbed," in *IFAC-PapersOnLine*, 2019, vol. 52, no. 1, pp. 562–567.

[37] D. Shah, A. Hancock, A. Skjellum, J. Wang, and Q. P. He, "Challenges and opportunities for IoT-enabled cybermanufacturing: what we learned from an IoT-enabled manufacturing technology testbed," in *Proceedings of Foundations of Computer Aided Process Operations / Chemical Process Control*, 2017, p. 66.

[38] J. Wang and Q. P. He, "Multivariate Statistical Process Monitoring Based on Statistics Pattern Analysis," *Ind. Eng. Chem. Res.*, 2010.

[39] K. Suthar, D. Shah, J. Wang, and Q. P. He, "Next-generation virtual metrology for semiconductor manufacturing: A feature-based framework," *Comput. Chem. Eng.*, vol. 127, pp. 140–149, 2019.

[40] Q. P. P. He and J. Wang, "Statistical process monitoring as a big data analytics tool for smart manufacturing," *J. Process Control*, vol. 67, pp. 35–43, 2018.

[41] Q. Peter He and J. Wang, "Statistics Pattern Analysis: A Statistical Process Monitoring Tool for Smart Manufacturing," in *Computer Aided Chemical Engineering*, 2018.

[42] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool Release: Gathering 802.11n Traces with Channel State Information," *ACM SIGCOMM Comput. Commun. Rev.*, 2011.

[43] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity Wi-Fi," *IEEE Trans. Mob. Comput.*, vol. 18, no. 6, pp. 1342–1355, 2018.

[44] I. Ahamed and M. Vijay, "Comparison of different diversity techniques in MIMO antennas," in *2017 2nd International Conference on Communication and Electronics Systems (ICCES)*, 2017, pp. 47–50.

[45] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "802.11 with multiple antennas for dummies," *ACM SIGCOMM Comput. Commun. Rev.*, 2012.

[46] J. Hu, H. Peng, T. Liu, X. Yao, H. Wu, and P. Lu, "A flow sensing method of power spectrum based on piezoelectric effect and vortex-induced vibrations," *Measurement*, vol. 131, pp. 473–481, 2019.

[47] Q. P. He and J. Wang, "Statistics pattern analysis: A new process monitoring framework and its application to semiconductor batch processes," *AIChE J.*, 2011.

[48] Q. P. He, J. Wang, and D. Shah, "Feature Space Monitoring for Smart Manufacturing via Statistics Pattern Analysis," *Comput. Chem. Eng.*, vol. 126, pp. 321–331, 2019.

[49] Q. P. He, J. Wang, H. E. Gilicia, J. D. Stuber, and B. S. Gill, "Statistics Pattern Analysis based Virtual Metrology for Plasma Etch Processes," no. Vm.

[50] D. Shah, J. Wang, and Q. P. He, "Feature Engineering in Big Data Analytics for IoT-Enabled Smart Manufacturing–Comparison between Deep Learning and Statistical Learning," *Comput. Chem. Eng.*, p. 106970, 2020.

[51] D. Shah, J. Wang, and Q. P. He, "A feature-based soft sensor for spectroscopic data analysis," *J. Process Control*, vol. 78, pp. 98–107, 2019.

[52] K. Suthar, D. Shah, J. Wang, and Q. Peter He, "Feature-based Virtual Metrology for Semiconductor Manufacturing," in *Computer Aided Chemical Engineering*, 2018.

[53] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, 2011.

[54] J. Friedman, T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*. Springer series in statistics New York, 2001.

[55] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, 1995.

[56] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Min. Knowl. Discov.*, 1998.

[57] C. M. Bishop, *Machine Learning and Pattern Recoginiton*. 2006.

[58] S. Theodoridis, “Neural Networks and Deep Learning,” in *Machine Learning*, 2015.

[59] C. C. Aggarwal, *Neural Networks and Deep Learning - A Textbook*. 2018.

[60] M. Nielsen, “Improving the way neural networks learn,” *Neural Networks Deep Learn.*, 2016.

[61] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, 1996.

[62] L. Breiman, “Pasting small votes for classification in large databases and on-line,” *Mach. Learn.*, 1999.

[63] L. Breiman, “Bagging predictors,” *Mach. Learn.*, 1996.

[64] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 1998.

[65] G. Louppe and P. Geurts, “Ensembles on random patches,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012.

[66] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[67] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, 2012.

[68] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, 2011.

[69] J. Bergstra, D. Yamins, and D. Cox, “Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms,” in *Proceedings of the 12th Python in Science Conference*, 2013.

[70] B. Komer, J. Bergstra, and C. Eliasmith, “Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn,” in *Proceedings of the 13th Python in Science Conference*, 2014.

[71] K. Lensing and B. Stirton, “Perspectives on integrated metrology and wafer-level control,” *IEEE Int. Symp. Semicond. Manuf. Conf. Proc.*, pp. 315–319, 2007.

[72] B. S. Gill, T. F. Edgar, and J. D. Stuber, “A novel approach to virtual metrology using Kalman filtering,” *Futur. Fab Int.*, no. 35, pp. 86–91, 2010.

[73] T. Lin, M. Hung, R. Lin, and F. Cheng, “A Virtual Metrology Scheme for Predicting CVD Thickness in,” *IEEE Int. Conf. Robot. Autom.*, vol. 12, no. May, pp. 1054–1059, 2006.

[74] R. Herschel and V. M. Miori, “Ethics & Big Data,” *Technol. Soc.*, vol. 49, 2017.

[75] C. Park and S. B. Kim, “Virtual metrology modeling of time-dependent spectroscopic signals by a fused lasso algorithm,” *J. Process Control*, vol. 42, 2016.

[76] S. Wold, N. Kettaneh-Wold, J. F. MacGregor, and K. G. Dunn, “Batch Process Modeling and MSPC,” in *Comprehensive Chemometrics*, 2010.

[77] R. Rendall, B. Lu, I. Castillo, S. T. Chin, L. H. Chiang, and M. S. Reis, “A Unifying and Integrated Framework for Feature Oriented Analysis of Batch Processes,” *Ind. Eng. Chem. Res.*, 2017.

[78] R. C. Wang, T. F. Edgar, M. Baldea, M. Nixon, W. Wojsznis, and R. Dunia, “Process fault detection using time-explicit Kiviat diagrams,” *AIChE J.*, 2015.

[79] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting & Control*. 2015.

[80] F. Haugen, *200 Nok*, no. August. 2012.

[81] M. R E Kalman (Reserach Institute for Advanced Study, Baltimore, “A New Approach to Linear Filtering and Prediciton Problems,” *J. Basic Eng.*, 1960.

[82] S. Wold, K. Esbensen, and P. Geladi, “Priciple component analysis,” in *Chemometrics and Intelligent Laboratory Systems*, 1987.

[83] I. T. Jolliffe, *Principal Component Analysis. Second Edition*. 2002.

[84] I. T. Jolliffe, “A Note on the Use of Principal Components in Regression,” *Appl. Stat.*, 1982.

[85] P. Geladi and B. R. Kowalski, “Partial least-squares regression: a tutorial,” *Anal. Chim. Acta*, 1986.

[86] M. H. Hung, T. H. Lin, F. T. Cheng, and R. C. Lin, "A novel virtual metrology scheme for predicting CVD thickness in semiconductor manufacturing," *IEEE/ASME Trans. Mechatronics*, 2007.

[87] P. Kang, D. Kim, and S. Cho, "Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing," *Expert Syst. Appl.*, 2016.

[88] Y. Zhang, B. Lu, and T. F. Edgar, "Batch trajectory synchronization with robust derivative dynamic time warping," *Ind. Eng. Chem. Res.*, 2013.

[89] H. Galiciaa, Q. Heb, and J. Wanga, "Statistics Pattern Analysis based fault detection and diagnosis," *CPC VIII Conf.*, pp. 1–16, 2012.

[90] Q. P. He, Jin Wang, H. E. Gilicia, J. D. Stuber, and B. S. Gill, "Statistics pattern analysis based virtual metrology for plasma etch processes," *2012 Am. Control Conf.*, pp. 4897–4902, 2014.

[91] Q. P. He, J. Wang, and D. Shah, "Feature space monitoring for smart manufacturing via statistics pattern analysis," *Comput. Chem. Eng.*, 2019.

[92] P. C. Zikopoulos, D. DeRoos, K. Parasuraman, T. Deutsch, D. Corrigan, and J. Giles, *Harness the Power of Big Data: The IBM Big Data Platform*. 2012.

[93] "Surface Metrology for In-Situ Pad Monitoring in Chemical Mechanical Planarization (CMP)."

[94] E. A. Baisie, Z. Li, and S. Technology, "Pad conditioning in chemical mechanical polishing : a conditioning density distribution model to predict pad surface shape Xiaohong

Zhang," vol. 8, no. 1, pp. 103–119, 2013.

[95] B. J. Hooper, G. Byrne, and S. Galligan, "Pad conditioning in chemical mechanical polishing," *J. Mater. Process. Technol.*, 2002.

[96] D. Boning *et al.*, "Run by Run Control of Chemical-Mechanical Polishing," vol. 19, no. 4, pp. 307–314, 1996.

[97] A. I. Khuri, "Multiresponse surface methodology," *Handbook of Statistics*. 1996.

[98] A. M. (2000). R. control in semiconductor manufacturing. C. press. Moyne, J., Del Castillo, E., & Hurwitz, *Run-to-Run Control in Semiconductor Manufacturing*. .

[99] J. Wang, Q. Peter He, and T. F. Edgar, "State estimation in high-mix semiconductor manufacturing," *J. Process Control*, 2009.

[100] P. Kadlec, B. Gabrys, and S. Strandt, "Data-driven Soft Sensors in the process industry," *Computers and Chemical Engineering*. 2009.

[101] 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*, 2011.

[102] M. L. Adams, *Rotating machinery vibration: From analysis to troubleshooting, second edition*. 2009.

[103] E. P. Carden and P. Fanning, "Vibration based condition monitoring: A review," *Structural Health Monitoring*. 2004.

[104] N. Tandon and A. Choudhury, "Review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings," *Tribol. Int.*, 1999.

[105] Analog Devices, “ADXL345 Product Specification,” *TRANSDUCERS 2009 - 15th Int. Conf. Solid-State Sensors, Actuators Microsystems*, 2009.

[106] M. Maksimović, V. Vujović, N. Davidović, V. Milošević, and B. Perišić, “Raspberry Pi as Internet of Things hardware : Performances and Constraints,” *Des. Issues*, 2014.

[107] D. Albright, “Arduino Vs Raspberry Pi: A Detailed Comparison,” *Digital Trends*. 2015.

[108] B. Bourque, “Arduino Vs Raspberry Pi,” *Digit. Trends*, 2015.

[109] D. Shah, J. Wang, and Q. P. He, “Feature engineering in big data analytics for IoT-enabled smart manufacturing – Comparison between deep learning and statistical learning,” *Comput. Chem. Eng.*, 2020.

[110] H. Wold, “Soft Modelling by Latent Variables: The Non-Linear Iterative Partial Least Squares (NIPALS) Approach,” *J. Appl. Probab.*, 1975.

[111] 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.

[112] D. M. Pirouz, “An Overview of Partial Least Squares,” *SSRN 1631359*, 2006.

[113] 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.

[114] S. Wold, M. Sjostrom, and L. Eriksson, “{PLS}-regression: a basic tool of chemometrics,” *Chemom. Intell. Lab. Syst.*, vol. 58, pp. 109–130, 2001.

[115] A. Höskuldsson, “PLS regression methods,” *J. Chemom.*, vol. 2, no. 3, pp. 211–228, 1988.

[116] M. M. Breuniq, H. P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying density-based

local outliers," *SIGMOD Rec. (ACM Spec. Interes. Gr. Manag. Data)*, 2000.

[117] E. O. Brigham and R. E. Morrow, "The fast Fourier transform," *IEEE Spectr.*, 1967.

[118] N. R. Lomb, "Least-squares frequency analysis of unequally spaced data," *Astrophys. Space Sci.*, 1976.

[119] W. H. Press and G. B. Rybicki, "Fast algorithm for spectral analysis of unevenly sampled data," *Astrophys. J.*, 1989.

[120] J. D. Scargle, "Studies in astronomical time series analysis. II - Statistical aspects of spectral analysis of unevenly spaced data," *Astrophys. J.*, 1982.