Hybrid Machine Learning Techniques for Manufacturing and Beyond

by

Jangwon Lee

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: Soft Sensor, Variable Selection, Process Monitoring,
Domain Knowledge, Machine Learning, Feature Engineering

Approved by

Q. Peter He, Chair, Associate Professor, Chemical Engineering
Jin Wang, Co-chair, Walt and Virginia Woltosz Endowed Professor, Chemical Engineering
Mario Eden, Joe&Billie McMillan Professor & Chair, Chemical Engineering
Nedret Billor, Professor, Mathematics and Statistics
Jesus Flores-Cerrillo, Associate Director, R&D and Digital, Linde

Abstract

This dissertation presents research performed to develop a novel soft sensor, feature space process monitoring, and domain knowledge-based path analysis for manufacturing and healthcare industries.

In recent years, as the Internet of Things (IoT) and data storage techniques (*i.e.*, cloud service) have been evolved, large-scale data are available to various industries such as retail, healthcare, and manufacturing. With notable demonstrations from the world's largest companies, such as Google, Amazon, Facebook and Microsoft, that insights can be obtained from big data, many businesses and institutions have been utilizing their own big data for potentially making new inferences and solving challenging problems in data-driven ways. However, it is sometimes difficult to extract valuable information and gain insights from big data with rote application of machine learning (ML) since data collected from various sources may not be relevant and often contain noises. Without domain knowledge, the results from ML approaches can be incomplete, or even lead to misleading conclusions. Therefore, in this research I aim to demonstrate the limitations of pure data-driven ML techniques in several case studies that are relevant to manufacturing and healthcare, and then to address the limitations by developing solutions that systematically integrate domain knowledge with ML techniques.

In the first part of this work (Chapter 2), I introduce a novel spectroscopy-based soft sensor which was developed by integrating a *feature engineering* approach – Statistics Pattern Analysis (SPA) – with a new *feature selection* approach – Consistency Enhanced Evolution for Variable Selection (CEEVS) – referred to as SPA-CEEVS. Based on the understanding of the spectral dataset that not all features contribute equally to the sample properties, a novel feature

selection method, CEEVS, is proposed to identify truly relevant features that are associated with chemical functional group regions, leading to improved soft sensor performance and easier interpretation of results compared to the soft sensor based on the original spectroscopy data. SPA, one of the feature engineering methods, is embedded in the CEEVS algorithm to better capture the characteristics of spectra such as nonlinearity. SPA can also reduce the influence of spectral disturbance and background noise by extracting statistics and shape features from spectral data. To demonstrate the effectiveness of the proposed SPA-CEEVS method, comparison study of various variable selection methods and nonlinear models are conducted on several industrial near-infrared (NIR) spectral datasets.

In the second part of this work (Chapter 3), I propose a *data-driven* feature space monitoring (FSM) approach that monitors periodic operations of pressure swing adsorption (PSA) processes. In FSM framework, features extracted from process variables are used to monitor the process operation, instead of raw process variables themselves. Domain knowledge of the PSA process helps to understand which features need to be generated and selected. In this work, I suggest a way of selecting features based on this domain knowledge. In addition, the FSM based fault detection method addresses challenges in monitoring periodic processes, such as unequal step and/or cycle time that requires trajectory alignment or synchronization for the traditional statistical process monitoring (SPM) methods. In this study, the k-nearest neighbor-based FSM (FSM-kNN) is developed for fault detection. The basic idea of FSM-kNN is that the distance between a faulty cycle and its neighboring training cycles (consisting of normal operation cycles) is greater than that between a normal cycle and its neighboring training cycles. In addition, a step-wise fault diagnosis is proposed to identify the root cause of faults when faults are detected. The proposed method not only shows superior fault detection performance

compared to the conventional SPM methods for both simulated faults and real faults from an industrial PSA process, but also correctly identifies the root causes of the faults.

In the third part of this work (Chapter 4), *path analysis* based on domain knowledge is proposed to examine if the hospitals specialized in certain diseases achieve better results in terms of costs and patient outcomes. With domain knowledge in healthcare industry, I formulate some hypotheses and construct paths. Pure data-driven ML approaches without hypotheses such as multiple linear regression and partial least square regression can lead to incomplete conclusion because they consider only one path among all the possible paths. However, the path analysis consists of all the possible paths where hospital specialization can affect the hospital performance so that the model can reveal full effects of hospital specialization. The comparison between the path analysis and the pure data-driven ML approaches suggests that domain knowledge can play a critical role in machine learning applications and should be incorporated whenever possible.

The contribution of this work and potential future work are summarized in Chapter 5. As demonstrated in this work, pure data-driven ML techniques have many limitations. For example, the results of pure data-driven ML tend to be sensitive to training datasets, possibly leading to irrational conclusions. The new feature selection and feature engineering techniques proposed in this work can improve the robustness and reliability of ML by reducing the influence of noises and disturbances, and capturing better process characteristics. In addition, the pure data-driven ML may fail to reveal the comprehensive effect of explanatory variables on a response variable. This work investigates the role of domain knowledge and how it enables us to establish knowledge-guided model structure. This structure would entail all logical paths to explain

various influences of explanatory variables on a response variable, leading to complete and mechanistically interpretable results.

Acknowledgements

I was able to complete this dissertation with the help of many people. They have contributed to my research in various ways. First of all, I would like to express my gratitude to Dr. Peter He. He has always stood by me in the journey of being a researcher and growing as an independent researcher. He has inspired me to think more and come up with better ideas when I encountered challenges. Without his support and guidance, I would not have completed my journey. My co-advisor, Dr. Jin Wang, also has provided me with great support and advice to conduct my research. She taught me how to work effectively and efficiently. In addition, she has shown her trust in me, encouraging me to be a better researcher.

I also would like to thank my other committee members. Dr. Mario Eden, who is also the department chair, has contributed to making a great environment that has allowed me to concentrate on my research and he has provided great supports when I needed his help. Dr. Nedret Billor has been an excellent guide for me in the world of statistics. I got more passionate about statistics and made my foundation for statistical knowledge by taking her courses, which was a critical component of my research. Last but not least, Dr. Jesus Flores-Cerrillo provides me great opportunities to learn how to handle actual data from industrial applications and has provided many valuable insights in monthly meetings, making my research encompass wider perspectives and more successful in solving real industrial problems.

Few things would have been left in my Ph.D. life if I missed my wonderful friend, Kerul Suthar. He has provided me with his full support whenever I needed his advice, help, and comfort. He was an amazing companion in my journey.

Lastly, I would like to thank my wife, Kyungjin Kim, for her tremendous support and endless patience. She has helped me to completely focus on my research and encouraged me to continue to do research. I also want to thank my son, Joseph, for being patient with an unkind father and growing well.

Jangwon Lee

Auburn, Alabama

June 8, 2021

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AFD	Average of First Derivative
AHRQ	Agency for Healthcare Research and Quality
ANN	Artificial Neural Network
API	Active Pharmaceutical Ingredient
ARD	Automatic Relevance Determination
ARS	Adaptive Reweighted Sampling
ASD	Average of Second Derivative
BETA	Regression Coefficient
CARS	Competitive Adaptive Reweighted Sampling
CEEVS	Consistency Enhanced Evolution for Variable Selection
CUSUM	Cumulative Sum
DPCA	Dynamic Principal Component Analysis
DR	Discriminant Ratio
DRG	Diagnosis Related Group
DTW	Dynamic Time Warping
EDF	Exponentially Decreasing Function
EWMA	Exponentially Weighted Moving Average
FDA	Fisher Discriminant Analysis
FSM	Feature Space Monitoring
GA	Genetic Algorithm
GPR	Gaussian Process Regression

HCUP	Healthcare Cost and Utilization Project
IoT	Internet of Things
IQI	Inpatient Quality Indicator
ITI	Information Theory Index
KDE	Kernel Density Estimation
KLD	Kullback-Leiber Divergence
kNN	k-Nearest Neighbors
KPCA	Kernel principal Component Analysis
LASSO	Least Absolute Shrinkage and Selection Operator
LOS	Length of Stay
LR	Logistic Regression
MCVT	Monte Carlo Validation and Testing
MICA	Multiway Independent Component Analysis
ML	Machine Learning
MLR	Multiple Linear Regression
MPCA	Multiway Principal Component Analysis
MSPM	Multivariate Statistical Process Monitoring
NDX	Number of Diagnosis
NIPALS	Nonlinear-Iterative Partial Least Square
NIS	National Inpatient Sample
NPR	Number of Procedure
NRMSECV	Normalized Root Mean Squared Error Cross Validation
NRMSEP	Normalized Root Mean Squared Error Prediction

OLS	Ordinary Least Squares
OR	Ordinal Regression
PA	Path Analysis
PCA	Principal Component Analysis
PCR	Principal Component Regression
PLSR	Partial Least Square Regression
PSA	Pressure Swing Adsorption
PSE	Process Systems Engineering
RBF	Radial Basis Function
RMSECV	Root Mean Squared Error Cross Validation
SLL	Slope of Linear Regression Line
SMOTE	Synthetic Minority Over-sampling Technique
SPA	Statistics Pattern Analysis
SPC	Statistical Process Control
SPM	Statistical Process Monitoring
SSL	Coefficient of Squared Term for Second Order Regression Line
SSR	Sum of Squared Residuals
SVP	Stability and Variable Permutation
SVR	Support Vector Regression
TOTCHG	Total Charge
UVE	Uninformative Variable Elimination
VIP	Variable Importance in Projection
WWTP	Waste Water Treatment Process

Chapter 1. Introduction

With the evolution of the sensing (*e.g.*, Internet-of-Things or IoT sensors) and data storage techniques, many industries such as automobile, chemical, petrochemical, and healthcare can easily record and store massive amount of data [1]–[3]. At the same time, advance in computational power enables the industries to analyze large quantities of data and extract valuable information from the data. As a result, tremendous amount of research has been conducted in the past few decades in the broad area of data-driven modeling to address complex industrial problems and to improve operation efficiency [1]–[10]. Among various types of data-driven methods, *process monitoring* and *soft sensor* are two of the most frequently used tools to improve the reliability of operation and to make better quality products. *Process monitoring* aims to detect abnormal conditions (*i.e.*, faults) in a system or process, and to quickly identify its root cause after a fault is detected. Process monitoring enables process engineers to detect and correct the process problems and return to normal operation in a timely manner. On the other hand, s*oft sensor* addresses challenges in obtaining some difficult-to-measure variables in real time, such as quality of product and yield in industrial processes, using other easy-to-measure variables. In other words, soft sensor predicts difficult-to-measure variables online based on the relationship between easy-to-measure variables and difficult-to-measure variables, which reduces time delay and enables quality monitoring of product in real time.

The process monitoring and soft sensor techniques can be implemented via either first-principles or data-driven approaches. First-principles approaches aim to build mathematical models to describe the physics of a system and failure mechanism for process monitoring and to quantify the effect of explanatory variables on response variables for soft sensing based on

physics and chemistry principles (*i.e.*, domain knowledge) [11]–[13]. One of the major advantages of first-principles approaches is that they can accurately describe nonlinear and transient process behaviors, resulting in superior prediction performance. First-principles models have been used to support process operations in the manufacturing industries for over 40 years due to this advantage [12]. Especially, they have been actively studied in the fields such as prognostics of equipment, simulation of industrial processes, and equipment modeling [13]–[17]. However, it is often difficult to apply first-principles models to industrial processes because the process behaviors are too complex and stochastic in nature to understand underlying physical/chemical phenomena. In contrast, data-driven approaches aim to describe process behaviors from historical data by utilizing statistics, machine learning or deep learning [11]. Although data-driven models require more historical data than first-principles models, they can approximate complex phenomena with limited understanding of system physics. Therefore, data-driven models are well suited for complex systems where underlying engineering and physical principles are not well known. Due to this nature of data-driven approaches, they are satisfactory for industrial applications and have been expanding to various industries. The first data-driven approach for process monitoring is statistical process control (SPC), which was pioneered by Shewhart in early 1920s [18]. SPC is used for quality control by utilizing the statistical distribution of quality variables. Since SPC is based on univariate Gaussian distribution, there is a limitation to improve model performance on multivariate processes. In the 1980 – 1990s, data-driven process monitoring applied multivariate statistical models such as principal component analysis (PCA) and partial least squares (PLS) to fault detection and diagnosis for reliable process operations [19], [20].

In recent years, data-driven approaches have become preferred to first-principles models because multi-level control systems make operation more complex in modern industries. In addition, data-driven approaches can deal with a gigantic size of historical data with the advance of computing power and storage techniques, thereby revealing a more accurate relationship between variables, leading to good model performance. Therefore, the most commonly applied process monitoring and soft sensor techniques employ data-based machine learning (ML) approaches, of which the ultimate objective is to improve operational efficiency through cost saving and reducing process downtime. In this study, I present how data-driven methods address manufacturing issues and improve decision making through accurate understanding of relationship between explanatory and response variables in both manufacturing and healthcare industries. However, results from pure data-based ML approaches sometimes can be incomplete and lead us to misleading conclusions. Thus, I aim to develop strategies that integrate domain knowledge with data-driven ML approaches (*i.e.,* hybrid ML) to obtain more reliable and interpretable results, leading to better decision making in manufacturing and healthcare industries. The ability of hybrid ML techniques to capture process characteristics is determined by hybrid architecture (*e.g.*, algorithms, input features, etc.). Therefore, the domain knowledge-guided feature selection and feature engineering frameworks are proposed to make better hybrid model's structures, helping to improve model performance. I believe these frameworks are applicable to process monitoring, soft sensor, and prognostics as well as any prediction cases in various industries.

The dissertation is structured as follows.

In Chapter 2, I propose a novel spectroscopy-based soft sensor that was developed by integrating a *feature engineering* approach – Statistics Pattern Analysis (SPA) – with a new

feature selection approach – Consistency Enhanced Evolution for Variable Selection (CEEVS). In this chapter, I discuss background on spectroscopy-based soft sensors and necessity of variable selection. Then, I review the different data-driven techniques for soft sensors compared in this study. These methods are classified into 1) variable selection methods based on linear partial least square regression (PLSR) and 2) nonlinear approaches including support vector regression (SVR) and Gaussian process regression (GPR). In this chapter, a new variable selection method (CEEVS) is proposed. CEEVS aims to improve the consistency of variable selection regardless of different training datasets. This method is based on Darwin's evolution theory, i.e., "survival of the fittest". The characteristics and algorithm of CEEVS are discussed in detail in this chapter. Next, I briefly review SPA feature-based soft sensor. SPA, one of the feature engineering methods, is integrated with CEEVS to further improve predictive power and interpretability. This SPA feature-based soft sensor integrated with CEEVS (SPA-CEEVS) is described in detail. Lastly, a new consistency index metric is proposed to measure the consistency of the variable selection. Detailed comparison of SPA-CEEVS with the other variable selection methods as well as nonlinear models is conducted using near infrared (NIR) datasets from different fields of industries.

In Chapter 3, I propose a new feature space monitoring (FSM) fault detection and diagnosis method for pressure swing adsorption (PSA) processes. The proposed method, k-nearest neighbor-based FSM (FSM-kNN), is completely different from many variants of the FSM method. First, I discuss the importance of PSA processes in various industries and the necessity of fault detection for reliable process operation. Then, a PSA process is briefly summarized to understand how different this process is from batch and continuous processes. I discuss the unique characteristics of the PSA process and how these characteristics pose

challenges to process monitoring; (1) it is operated in a periodic fashion, which leads to unsteady-state, or highly dynamic process behavior, (2) one cycle consists of many different steps with different operation conditions, which leads to highly complex nonlinear behavior, and (3) cycle time is frequently adjusted to meet demand fluctuations, which leads to multimodal operations of the process. Many traditional fault detection methods have been studied to address these challenges of the PSA process monitoring. In this chapter, I review the conventional fault detection methods – a PCA-based fault detection method and kNN rule-based fault detection methods. Since these fault detection methods have some limitations for successful process monitoring, I develop FSM-kNN method to improve the fault detection rate and to reduce the false alarm rate. In this approach, the process condition is monitored using statistics and various features extracted from pressure profiles of each PSA step instead of utilizing the raw pressure profiles of the PSA process. I focus on how to extract the statistics and shape features from the original variables of the PSA process. Then, the algorithm of FSM-kNN method is described in detail. In this chapter, I next propose a step-wise fault diagnosis to identify the root cause of the faulty step(s). To demonstrate the effectiveness of the proposed fault detection and diagnosis method, I compare the FSM-kNN with three different conventional fault detection methods using both simulated fault data and real fault data from an industrial PSA process. Based on the comparison results, I discuss how the FSM-kNN method overcomes the difficulties of PSA process monitoring.

In Chapter 4, I examine whether the focused factory theory (*i.e.*, factories that concentrate on narrow range of services or operations produce better products at low costs) is applicable to hospital operations. Since hospital operations and manufacturing processes share some similarities at systems level, I believe some of the theories and techniques developed for

manufacturing processes are also applicable to hospital operations. Specifically, I examine whether the hospitals that are specialized in certain diseases achieve better results in terms of costs and patient outcomes using a large national healthcare cost and utilization project (HCUP) dataset. I first introduce the HCUP dataset used in this work and share the challenges in analyzing this dataset. In this chapter, a specialization index is proposed to quantify hospital specialization. With the specialization index, pure data-driven machine learning (ML) approaches are used to investigate the effect of hospital specialization on hospital performance in terms of cost (measured by total charge) and patient outcome (measured by death of patient during hospitalization). The various ML approaches are briefly reviewed, and the limitations of the pure ML approaches are discussed; they can only uncover the partial effect of hospital specialization on hospital performance. To address the limitations of the pure data-driven ML methods, I propose a knowledge-guided path analysis to comprehensively understand how the hospital specialization influences the hospital performance. The path analysis consists of all the possible paths where the hospital specialization can affect the hospital performance. Through comparison between the pure data-driven ML methods and the knowledge-guided path analysis, I demonstrate that, without domain knowledge, the results from naive ML approaches are incomplete and misleading. The full effects of specialization are revealed only when ML is applied to a model structure that is defined based on domain knowledge. The comparison results suggest that domain knowledge can play a significant role in machine learning applications and should be incorporated whenever possible.

In Chapter 5, contributions of this work are summarized and the potential directions for future work are proposed. Specifically, I aim to advance industrial process monitoring and soft sensor techniques with domain knowledge. In addition, I develop a better structure of path

analysis with the aid of domain knowledge to understand the effect of hospital specialization. I believe that suggestions in future research directions make further enhancements to the proposed methods in this work.

Chapter 2. Novel spectroscopy-based soft sensor

2.1 Background

With the advancements of spectroscopic technologies including near infrared (NIR), Ramon Spectroscopy, and UV/Vis spectroscopies, various properties could be inferred from a sample's spectrum profile. Correspondingly, multivariate modeling approaches (i.e., soft sensor models), which correlate the spectroscopic reading of a sample to its properties of interest, have drawn increased research interest. These soft sensor models offer a non-invasive, fast, and cheap way to estimate the sample properties of interest and have been applied in many different fields. For example, spectra-based soft sensors have been developed to determine properties such as octane number of gasoline, moisture or protein content of corn, active pharmaceutical ingredient (API) in drug, and microorganism concentration in a mixed culture [21]–[26]. The most commonly used modeling approach for soft sensor is partial least squares (PLS) due to its simplicity, robustness and the inherent capability in addressing collinearity among predictor variables.

However, since PLS is a linear model, its performance can be unsatisfactory for datasets with nonlinear relationship between predictors and response variables. The nonlinear soft sensors such as support vector regression (SVR), artificial neural network (ANN) and Gaussian process regression (GPR) have been studied in the literature [27]–[31]. Although the nonlinear methods can successfully quantify the sample properties of interest for nonlinear datasets, they have a couple of limitations that are originated from the following characteristics of NIR spectra: (1) multicollinearity, (2) spectra noise, and (3) high dimensionality. These characteristics could

---

This chapter was excerpted from "Consistency-enhanced evolution for variable selection can identify key chemical information from spectroscopic data" published in Industrial & Engineering Chemistry Research [21] and from "Improving featured-based soft sensing through feature selection" published in IFAC-PapersOnLine [22]. The author is the first author of these papers.

increase the risk of over-fitting for high dimensional data. In addition, it is difficult to understand underlying predictor variables associated with chemical functional groups due to the complexity of the nonlinear methods.

Variable selection can be one of the solutions to address these challenges. It has been well-recognized that the performance of a soft sensor can be significantly improved only when relevant variables are included as a predictor [32]–[36]. This is particularly important for spectrum-based soft sensors because readings at different wavelengths are highly correlated. In addition, although many multivariate statistical methods including PLS require much larger number of samples than the number of variables to perform well, most spectral datasets have relatively small sample size (less than 100) but a large number of variables (several hundreds of wavelengths). Therefore, eliminating irrelevant wavelengths could help circumvent the difficulty that arises from many variables. This has led to the development of many variable selection methods in the past few decades. Most of the existing variable selection methods focus on selecting the variables (i.e., wavelengths or wavelength segments) that are strongly correlated with a response variable to improve prediction performance. These variable selection approaches include direct methods that rank variable contributions such as variable selection based on variable importance in projection (VIP) [37] or regression coefficient (BETA) [34], and iterative methods such as uninformative variable elimination (UVE) [38], least absolute shrinkage and selection operator (LASSO) [39] and Elastic Net [40]. Among iterative approaches, a group of variable selection methods based on the principle of "survival of the fittest" have shown superior performance. The representative methods of this group are the genetic algorithm (GA) [41]–[43], the competitive adaptive reweighted sampling method (CARS) [25] and the method based on stability and variable permutation (SVP) [44]. By employing the principle of "survival of the

fittest", these methods rely on random sampling in the variable space and/or sample space to identify the most relevant predictor variables to improve prediction performance.

Despite many successful applications, existing variable selection methods also have limitations. It has been recognized that a soft sensor model with good fitness performance may not guarantee good variable selection performance [32], [34]. Specifically, for spectrum-based soft sensors, the selected wavelengths sometimes show little connection to the chemical bounds or functional groups presenting in the sample. In addition, the selected variables can be quite sensitive to the choice of the training and validation data. In particular, the variables selected from different Monte Carlo (MC) runs using randomly selected training and validation data often show low consistency with each other. The inconsistency among different MC runs suggests that the selected variables (wavelengths) may not contain the truly relevant predictors that are the wavelengths associated with the underlying chemical bonds or functional groups that determine the sample properties.

To address this limitation of variable selection, a feature-based soft sensor was developed, which originated from SPA [24], [45]. In the SPA feature-based soft sensor, whole spectrum is split into equally spaced segments and then the statistics/features are extracted along the variable (wavelength) dimension in each segment. Instead of spectral reading of samples, the statistics/features are used as predictors to build the PLS model. This approach provides four benefits: (1) the features better capture spectral characteristics such as nonlinearity and peak shift; (2) features reduce the number of predictors (wavelengths); (3) extraction of features could filter out the spectral noise and disturbance; (4) the whole information in the spectrum could be used for the soft sensor.

However, although the number of features extracted from whole spectrum is significantly reduced compared to the number of wavelengths, SPA feature-based method still does not solve the curse of dimensionality, *i.e.,* the number of features is larger than the number of calibration samples. In addition, considering all the features do not equally contribute to sample properties, variable selection would be desirable. Thus, a novel variable selection method CEEVS is developed, which aims to improving the consistency of variable selection [21], [46]. The basic idea of the method is that if the selected variables are consistent regardless of the calibration sets, they are likely to be truly relevant ones and would contribute to improve the prediction power. Details of CEEVS is discussed in Section 2.3.

In this work, to further improve soft sensor's performance, a novel spectroscopy-based soft sensor, SPA-CEEVS is proposed by integrating the feature engineering – SPA – with the feature selection – CEEVS to utilize the advantages of both methods. The proposed method not only improves the predictive accuracy (driven by SPA framework) and consistency of feature selection (driven by CEEVS) for NIR data sets, but also delivers easier interpretation of results. The effectiveness of SPA-CEEVS is demonstrated by using five NIR data sets. The performance of SPA-CEEVS is compared with five variable selection methods, *i.e.,* CARS, SVP, GA, Elastic Net and CEEVS and two nonlinear methods, *i.e.,* SVR and GPR. The full PLS model is used as the basis for comparison of all the methods.

2.2 Review of variable selection algorithms and machine learning methods

2.2.1 Partial least square regression (PLSR)

PLSR is one of multivariate statistical techniques to find the relationship between predictor variables and response variables. PLSR 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 [47] is a popular algorithm to implement PLS. More information on the algorithm and its properties are discussed in [48]–[51].

$\boldsymbol{X}_{n\times m}$ 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:

$$\boldsymbol{X}_{n\times m} = \boldsymbol{T}_{n\times p}\boldsymbol{P}^{T}_{m\times p} + \boldsymbol{E}_{n\times m} \quad (2.1)$$

$$\boldsymbol{Y}_{n\times l} = \boldsymbol{U}_{n\times p}\boldsymbol{Q}^{T}_{l\times p} + \boldsymbol{F}_{n\times l} \quad (2.2)$$

where p is the number of principal components; $\boldsymbol{T}_{n\times p}$ and $\boldsymbol{Q}^{T}_{l\times p}$ 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 $\boldsymbol{U}$.

2.2.2 Genetic algorithm (GA)

Inspired by Darwin's evolution theory of "survival of the fittest", GA is one of the most commonly applied variable selection methods [29], [42], [43]. According to the evolution theory, the individuals who are well adapted to the environment will be more likely to survive and produce the next generation [41]. Therefore, in GA, parent chromosomes (*i.e.*, subsets of selected variables) are determined based on its "fitness to the environment", such as prediction performance. Then crossover and mutation are applied to produce offspring, *i.e.*, new sets of selected variables. Through crossover, portions of two parent chromosomes are crossed and combined to make two offspring which have new combinations of genes (*i.e.*, variables or wavelengths); through mutation, new genes not included in the chromosomes population could

have a chance to be included, which may improve the offspring's fitness to the environment. This reproduction step is repeated until a termination criterion is satisfied [52].

2.2.3 Competitive adaptive reweighted sampling (CARS)

In CARS, the importance of a variable is determined based on its absolute regression coefficient (BETA) obtained through PLSR. The variables with large absolute regression coefficients are considered as the important variables. CARS employs the iterative sampling runs to determine the optimal subset of variables. In each sampling run, two variable reduction procedures, namely exponentially decreasing function (EDF) and adaptive reweighted sampling (ARS), are applied to reduce the number of variables. The root mean square error of cross-validation ($RMSE_{CV}$) is calculated using the variables retained in the sampling run. After n_S times sampling runs, CARS obtains n_S models consisting of the different subsets of variables and the model with the lowest $RMSE_{CV}$ is selected as the optimal model [25]. CARS has been applied to develop soft sensors in many different applications, including spectroscopic data collected from GC-MS, NIR, and UV/Vis [53]–[56].

2.2.4 Stability and variable permutation (SVP)

Recently, SVP was proposed based on the evolutionary principles of 'intraspecific competition' and 'survival of the fittest'. In SVP, the importance of each variable is determined through variable stability and variable permutation analysis. Variable stability is evaluated through random sampling of the sample space, while variable permutation analysis is performed through random sampling of the variable space. After computing the variable stability and performing variable permutation analysis, SVP divide all the variables into the elite variable set and normal variable set by adaptive reweighted sampling (ARS). The elite variable set consists of variables with high stability, while the normal set contains variables with relatively low

stability. To eliminate the uninformative variables, SVP employs exponentially decreasing function (EDF), which remove variables with small difference from the normal variable set. In each sampling run, the procedures described above are performed. After n_S sampling runs, SVP obtains n_S models with different variable subsets; then the variable subset that results in minimum mean and relatively low standard deviation of the $RMSE_{CV}$'s is selected as the optimal subset of the selected variables [44].

2.2.5 Support vector regression (SVR)

Consider training data set $\{(x_1, y_1), \cdots, (x_n, y_n)\} \subset \chi$, where $\chi \in \mathbb{R}^m$ is the space of predictor features, $y_i \in \mathbb{R}$ is the response variable, and n is the number of the training data set. The goal of SVR is to find $f(x)$ that has the deviation no larger than ε for all training data y_i and at the same time is flat as possible. The linear function $f(x)$ has the form:

$$f(\mathrm{x}) = <\mathrm{w}, \mathrm{x}> + \mathrm{b} \text{ with } \mathrm{w} \in \chi, \mathrm{b} \in \mathbb{R} \qquad (2.3)$$

where, $<\cdot, \cdot>$ represents the dot product in χ. The flat function $f(x)$ can be obtained through minimizing the norm of w, *i.e.*, $\|w\|^2 = <\mathrm{w}, \mathrm{w}>$. By introducing the slack variables ξ_i, ξ_i^*, we can obtain the feasible solution. The optimization problem is defined as following:

$$\text{minimize } \frac{1}{2}\|\mathrm{w}\|^2 + \mathrm{C}\sum_{i=1}^{n}(\xi_i + \xi_i^*) \qquad (2.4)$$

$$\text{subject to } \begin{cases} y_i - <w, \mathrm{x}_i> - b \leq \varepsilon + \xi_i \\ <w, \mathrm{x}_i> + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \qquad (2.5)$$

where constant $\mathrm{C} > 0$ is a regularization term which determines the trade-off between the flatness of $f(x)$ and the toleration of deviations larger than ε. The objective function shown in equation 2.4 can be reformulated into a dual problem by introducing a dual set of variables (i.e., Lagrange multipliers). The solution can be obtained by satisfying Karush-Kuhn-Tucker (KKT) conditions and can be defined as:

$$f(\mathrm{x}) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)k(\mathrm{x}_i, \mathrm{x}) \quad (2.6)$$

where α_i, α_i^* are Lagrange multipliers and $k(\mathrm{x}_i, \mathrm{x})$ represents kernel mapping. The Gaussian kernel function is used for nonlinear regression, while the dot product between two sample feature vectors is used for linear regression. The Gaussian kernel (*i.e.*, radial basis function (RBF)) is defined as:

$$k(\mathrm{x}_i, \mathrm{x}_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (2.7)$$

where σ is the kernel parameter. More details about SVR can be found in [57].

2.2.6 Gaussian process regression (GPR)

Gaussian process regression model is a nonparametric kernel-based probabilistic model. Consider training data set $\{(x_1, y_1), \cdots, (x_n, y_n)\} \subset \chi$, where $\chi \in \mathbb{R}^m$ is the space of predictor features, $y_i \in \mathbb{R}$ is the response variable, and n is the number of the training data set. The objective of a regression model is to find the relationship $y = f(\mathbf{x}|\theta) + \varepsilon$ between predictor and response variables. Gaussian process regression is established as the regression function with zero-mean Gaussian prior distribution, shown as follows:

$$y = [f(\mathbf{x}_1), f(\mathbf{x}_2), \cdots, f(\mathbf{x}_1)] \sim \mathrm{GP}(0, \mathbf{C}) \quad (2.8)$$

where $\mathbf{C}$ is the covariance matrix, and in this work, automatic relevance determination (ARD) squared-exponential covariance function is used, which is defined as:

$$\mathbf{C}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left\{-\frac{1}{2}\sum_{k=1}^{m}\frac{(\mathrm{x}_{ik} - \mathrm{x}_{jk})^2}{\sigma_k^2}\right\} + \delta_{ij}\sigma_n^2 \quad (2.9)$$

where σ_f^2 and σ_n^2 are signal variance and noise variance, respectively; σ_k^2 is the length scale for each predictor; $\delta_{ij} = 1$ when $i = j$, otherwise $\delta_{ij} = 0$. Let $\theta = (\sigma_f^2, \sigma_n^2, \sigma_1^2, \sigma_2^2, \cdots, \sigma_m^2)$ stand for hyperparameter set. The hyperparameters can be optimized by maximizing the marginal likelihood. Once optimizing the hyperparameters, I can obtain the posterior distribution of the

y_{new} for corresponding $\mathbf{x}_{new}$, which follow a Gaussian distribution of which the mean and variance can be defined as follows:

$$\bar{y}_{new} = \mathbf{K^T}(\mathbf{x}_{\text{new}})\mathbf{C}^{-1}\mathbf{y} \quad (2.10)$$

$$\sigma^2_{new} = \mathbf{C}(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{new}}) - \mathbf{K^T}(\mathbf{x}_{\text{new}})\mathbf{C}^{-1}\boldsymbol{K}(\mathbf{x}_{\text{new}}) \quad (2.11)$$

where $\boldsymbol{K}(\mathbf{x}_{\text{new}}) = [\boldsymbol{C}(\mathbf{x}_{\text{new}}, \mathbf{x}_1), \boldsymbol{C}(\mathbf{x}_{\text{new}}, \mathbf{x}_2), \cdots, \boldsymbol{C}(\mathbf{x}_{\text{new}}, \mathbf{x}_{\text{n}})]^T$. More details about GPR can be found in [58], [59].

2.2.7 Elastic Net

Elastic net is a linear regression model trained by shrinking the regression coefficient with both L1 norm penalty (lasso) and L2 norm penalty (ridge). Elastic net was developed to encourage grouping effect during variable selection, which enables the method to select groups of correlated variables. The objective function can be defined as follows:

$$\min_{\beta_0,\beta}\left(\frac{1}{2n}\sum_{i=1}^{n}(y_i - \beta_0 - x_i^T\beta)^T + \lambda P_\alpha(\beta)\right) \quad (2.12)$$

where $P_\alpha(\beta) = \frac{(1-\alpha)}{2}\|\beta\|_2^2 + \alpha\|\beta\|_1$ and α is the mixing parameter between ridge ($\alpha = 0$) and lasso ($\alpha = 1$). More details about Elastic net can be found in [40].

2.3 Introduction to Consistency Enhanced Evolution for Variable Selection (CEEVS)

The variables selected by many variable selection methods such as GA, CARS and SVP are not necessarily the truly relevant variables, *i.e.*, the ones corresponding to the key chemical bonds or functional groups that determine the sample properties of interest. Another evidence that the selected variables may not be the informative variables is that variable selection results are very sensitive to training samples. In order to understand the reason for the low consistency of variable selection, first, it is important to figure out the difference between the evolution theory based variable selection and biological evolution. In biological evolution, it usually takes millions of years for natural selection to converge to an optimal solution; however, in variable

selection the limited sample space and the limited evolution process may cause the variable selection to be stuck in a local optimum and miss the global one. Therefore, I believe that the limited sample space and limited evolution may be one of the underlying reasons for the inconsistency among different MC runs. However, without knowing what the global optimum is (i.e., the ground truth of the truly relevant variables), it is difficult to devise approaches to directly address this limitation.

I address this difficulty based on the following rationale: if a variable selection algorithm can identify the truly relevant input variables, it should consistently identify the same subset of the variables regardless of the choice of the training samples. In other words, the variable selection results among different MC runs should be relatively consistent to identify the truly relevant predictors. Therefore, I hypothesize that if a variable selection method delivers better consistency in terms of selected variables among different MC runs, it is more likely that it selects the truly relevant variables and as a result would deliver better prediction performance. Based on this hypothesis, the CEEVS algorithm aims to improve the consistency in variable selection.

2.3.1 Notation

In this work, $\boldsymbol{X}_{n\times m}$ denotes the spectral data, which consists of n samples and spectral absorbances of m wavelengths for each sample; $y_{n\times 1}$ denotes a response variable vector (*i.e.*, property of interest) with dimension of $n \times 1$. Both $\boldsymbol{X}_{n\times m}$ and $y_{n\times 1}$ are autoscaled to zero mean and unit variance before model development through PLS. The equations of PLS model can be found in equations 2.1 and 2.2 where l equals to 1 in this study.

2.3.1.1 Gene, chromosome, and fitness

The CEEVS method is based on the "survival of the fittest" principle, and follows the same terminologies as GA. A gene refers to an individual variable (wavelength), and a chromosome ($C_{m\times 1}$) refers to a set of selected variables: the *i*-th element (c_i) of the chromosome is either "1" or "0", indicating whether the *i*-th variable is included in the chromosome or not, respectively. The fitness of a chromosome is determined through prediction error, *i.e.*, normalized root mean squared error from cross-validation ($NRMSE_{CV}$).

$$NRMSE_{CV} = \frac{\sqrt{\frac{1}{n_V}\sum_{i=1}^{n_V}(y_i-\hat{y}_i)^2}}{(y_{max}-y_{min})} \times 100\% \qquad (2.13)$$

where, n_V is the number of samples of the validation dataset. In this work, 10-fold cross validation is employed for all methods. Therefore, the average of the ten $NRMSE_{CV}$'s is used.

2.3.1.2 Variable Stability and Probability

In existing literature [25], [44], [60], variable stability is determined through random sampling of the training data and evaluating how consistently the variable contributes to the soft sensor model. Specifically, to compute the stability, MC sampling is applied in which certain percentage (denoted as γ) of the n samples are randomly selected to build a PLS model, and this random selection is iterated for n_S times. A full PLS model that include all wavelengths as predictor variables is established for each subset of data to compute regression coefficients. As regression coefficient (BETA) determines how much a variable contribute to the prediction of the response variable, it has been used to evaluate the stability of each variable.

$$S_{BETA-j} = \frac{|\bar{b}_j|}{\sqrt{\frac{1}{n_S-1}\sum_{i=1}^{n_S}(b_{ij}-\bar{b}_j)^2}} \qquad (2.14)$$

where, S_{BETA-j} is the stability of the j-th variable based on regression coefficients, $\bar{b}_j$ is the average value of regression coefficients of j-th variable from n_S full PLS models using samples

randomly selected from the training dataset based on the pre-determined sampling ratio γ, and b_{ij} is the regression coefficient of j-th variable in i-th PLS model.

Besides regression coefficient BETA, variable importance in projection (VIP) also indicates how much a variable contributes to the response variable. Unlike BETA, VIP scores estimate the importance of each variable in the projection used in a PLS model. It has been reported that when each predictor contributes differently to the response variable (which is the case for most, if not all, practical applications), BETA-based variable selection may not work as well as VIP-based variable selection [32], [34]. In fact, Wold et al. [37] recommended a combination of VIP and BETA for variable selection. To improve the consistency of variable selection, in this work I propose using the combination of VIP and BETA to compute variable stability. To do so, I first define variable stability based on VIP.

$$S_{VIP-j} = \frac{|\bar{v}_j|}{\sqrt{\frac{1}{n_S-1}\sum_{i=1}^{n_S}(v_{ij}-\bar{v}_j)^2}} \tag{2.15}$$

where S_{VIP-j} is the stability of the j-th variable based on VIP scores, $\overline{v_j}$ is the average value of the VIP scores of j-th variable among n_S models, and v_{ij} is the VIP score of j-th variable in the i-th model. To combine S_{BETA} and S_{VIP} for determining the stability for each variable, S_{BETA} and S_{VIP} are first standardized since they have different scales.

$$Z_{BETA-j} = \frac{S_{BETA-j} - \overline{S_{BETA}}}{std(S_{BETA})} \tag{2.16}$$

$$Z_{VIP-j} = \frac{S_{VIP-j} - \overline{S_{VIP}}}{std(S_{VIP})} \tag{2.17}$$

where $\overline{S_{BETA}}$ and $\overline{S_{VIP}}$ are the average stability of all variables based on BETA and VIP scores, respectively; $std(S_{BETA})$ and $std(S_{BETA})$ are the corresponding standard deviations, respectively. Then the average of Z_{BETA-j} and Z_{VIP-j}, denoted as Z_j, is used to determine the stability of the j-th variable.

$$Z_j = \frac{1}{2}(Z_{BETA-j} + Z_{VIP-j}) \quad (2.18)$$

Note that in this work, Z_{BETA} and Z_{VIP} were assigned the same weight, which can be adjusted for different applications.

To remove any potential bias, I first convert the variable stability into a probability; then each variable is randomly selected according to its probability to generate the initial population of chromosomes. The probability of the j-th variable is defined as followings:

$$p_j = \lambda_1 + (\lambda_2 - \lambda_1)\left(\frac{Z_j - Z_{min}}{Z_{max} - Z_{min}}\right) \quad (2.19)$$

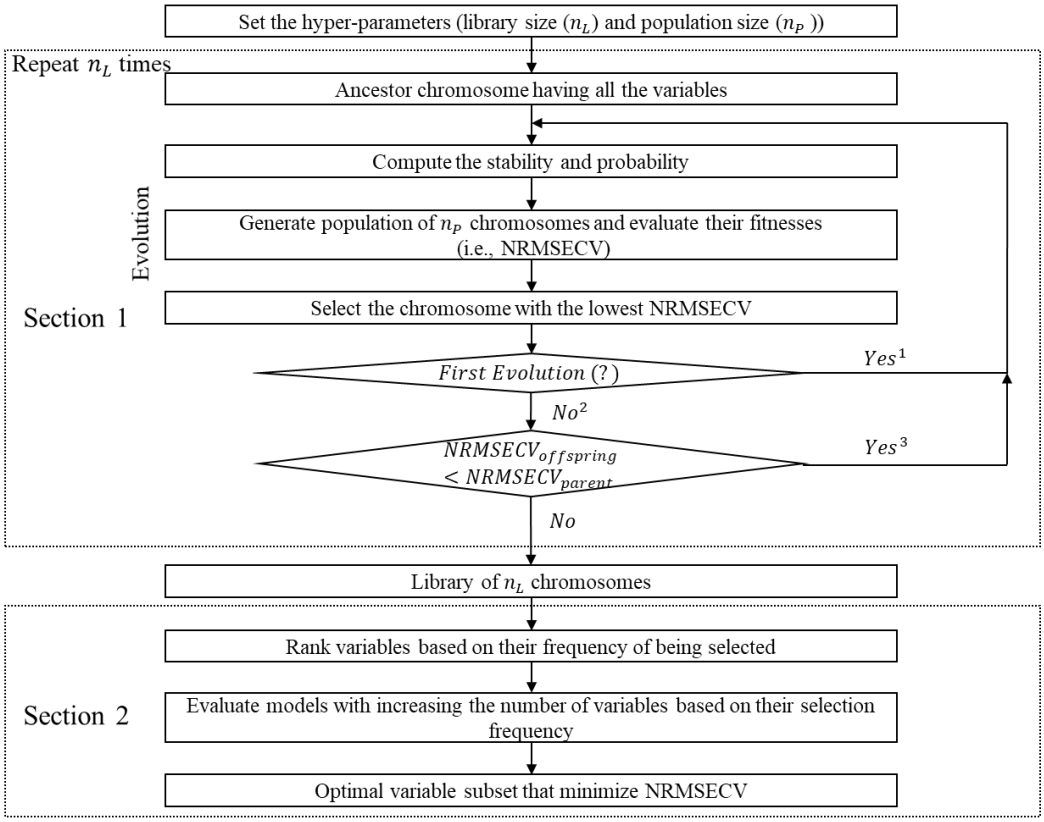
where, λ_1 is a small probability (10^{-5} in this work) to ensure that even the variable of the minimum stability has a chance to be selected and evaluated; λ_2 is 1; Z_{max} and Z_{min} are the maximum and minimum stabilities among all variables, and Z_j is the stability of the j-th variable. When $Z_j = Z_{min}$, $p_j = \lambda_1$; when $Z_j = Z_{max}$, $p_j = \lambda_2 = 1$.

2.3.2 CEEVS Algorithm

As shown in Figure 2.1, CEEVS consists of two main sections: Section I is to construct a library with optimal chromosomes, and Section II is to select the optimal subset of variables from the library to build the soft sensor.

For Section I, CEEVS takes a consistency enhanced evolution process in order to obtain an optimal chromosome with limited iterations. In GA, the chromosomes of the initial population are generated randomly where each variable has the same probability to be selected. In CEEVS, starting with the complete variable set, the initial chromosome population is generated randomly based on each variable's probability of selection as defined in equation 2.19. As shown in Section 2.3.1.2, the probability of selection is simply a scaled variable stability; in other words, variables with higher stability will be selected with higher probability. In this way, the evolution process will start with a better initial population, as more important variables will more likely be

selected for the initial population. Once the initial population of n_P chromosomes are obtained, each chromosome is evaluated for its fitness value. I use the selected variables (i.e., the variables that have "1" in the chromosome) to build a PLS model, and the model's $NRMSE_{CV}$ value is used as the fitness value for the chromosome. The optimal chromosome, i.e., the one with the minimal $NRMSE_{CV}$ within the initial population, is considered as a parent to generate offspring for the evolution process. The objective of the evolution process is to further eliminate the uninformative variables in the parent chromosome before it is stored into the library. Again, 10-fold cross validation is employed in this work for all methods. Therefore, $NRMSE_{CV}$ is actually referring to the average of the 10 $NRMSE_{CV}$ values.



1. The selected chromosome to be parent chromosome.
2. The selected chromosome in previous evolution to be parent chromosome.
   The selected chromosome in current evolution to be offspring chromosome.
3. The offspring to be parent chromosome.

Figure 2.1 Flow diagram of CEEVS algorithm

The evolution process of CEEVS is completely different from GA. Instead of cross-over and mutate, in CEEVS, I simply use the variables selected by the parent chromosome as the new complete variable set, and repeat the whole process to generate the next best chromosome which is denoted as an offspring. For each additional run of evolution, the offspring from the previous run is considered the parent chromosome, and the variable selected by the parent chromosome is considered as the new "full" variable set; Next, the variable stability and probability are re-computed for this new "full" set; then, a population of n_P offspring are generated randomly based on the variable's probability for selection, and evaluated for their fitness value. In this way, all the offspring are guaranteed to contain fewer variables than the parent and may have a better fitness value. This evolution process is repeated until the fitness of the offspring is worse than that of the parent, meaning the parent can no longer produce better offspring. Then the parent of the final evolution run, i.e., the best chromosome generated from the evolution process, is stored into the library. This evolution process will repeat n_L times with different random seeds, which is the pre-determined library size, i.e., the number of the optimal chromosomes to be stored in the library. Each time the process starts with the complete set of variables. At the end of n_L repetitions, the library will contain n_L optimally evolved chromosomes, i.e., subsets of selected variables that deliver the lowest $NRMSE_{CV}$ during each repeated evolution process.

For Section II, starting with the library that contains n_L best chromosomes generated in Section I, I first rank all the variables based on their frequency of presence in the library. Next, I build a series of PLS models with increasing the number of variables based on their selection frequency. In other words, the first PLS model is built with the most frequently selected variables in the library and the second model adds the next frequently selected variable. This process is repeated until the number of variables included in the model reaches a pre-defined

upper limit. This upper limit can be adjusted to reduce the risk of overfitting. In this work, I set the upper limit as 300 variables. Finally, all models are evaluated for their fitness ($NRMSE_{CV}$), and the variable subset that produce lowest $NRMSE_{CV}$ value is considered the final result of the selected variables.

It is worth noting that all Monte Carlo (MC) repetitions involved in the CEEVS and other variable selection methods are carried out on the training samples only. Specifically, the procedures of CEEVS shown in Figure 2.1 were all performed using the training samples only, with n_L MC repetitions of different random seeds to generate library of n_L chromosomes.

2.3.3 Choice of Tuning Parameters

One of the advantages of CEEVS is simpler tuning compared to GA. First, there are only four parameters in CEEVS, which include the library size (n_L), the population size (n_P), the ratio of samples (γ) and the number of sampling runs (n_S). n_L determines the number of the chromosomes to be stored in the library, which is also the number of repetition (or evolution) in Section I of the algorithm. n_P is the number of chromosomes present in each population. γ and n_S are related to evaluating variable stability: γ is the ratio or percentage of samples to be randomly selected and n_S is the number of the randomly selected sample subsets, i.e., the number of PLS models to be built in order to evaluate the variable stability. Second, CEEVS is not sensitive to these parameters. As detailed later in Section 2.4.3, sensitivity analysis shows that when n_L, n_P, γ and n_S are large enough, their effect on the final soft sensor performance becomes negligible. Therefore, in this work I decide to keep all 4 parameters fixed instead of changing them from dataset to dataset. Table 2.1 lists the parameter setting used in this work, and the recommended range if one chooses to fine-tune the parameter.

Table 2.1 Parameters used in this work and recommended range of tuning parameters

	Parameter (this work)	Recommended range
n_L	200	100 – 500
n_P	400	200 – 500
γ	0.9	0.8 – 0.9
n_S	400	300 – 800

2.4 Case studies, Performance metrics, Results & Discussion

2.4.1 Case studies

Five published NIR datasets are used to evaluate the performance of different variable selection methods. Table 2.2 summarizes the five datasets, including the number of samples and variables, the partition of the dataset into training and testing, as well as relevant references. Figure 2.2 plots the sample spectra for each dataset.

Table 2.2 Summary of the Five NIR Datasets

	# of samples in calibration set	# of samples in test set	# of samples in total	# of variables	Property of interest	Ref.
Corn[a]	64 (80%)	16 (20%)	80	700	Protein content	[24], [61]
Diesel	180 (70%)	76 (30%)	256	401	Aromatic content	[44], [62]
Pharma	459 (70%)	196 (30%)	655	650	Active pharmaceutical ingredients (API)	[24], [26], [63]
Wheat	121 (80%)	30 (20%)	151	150	Protein concentration	[44], [64]
Beer	48 (80%)	12 (20%)	60	926	Extract concentration	[65], [66]

[a]NIR spectra measured on mp5 spectrometer was used.

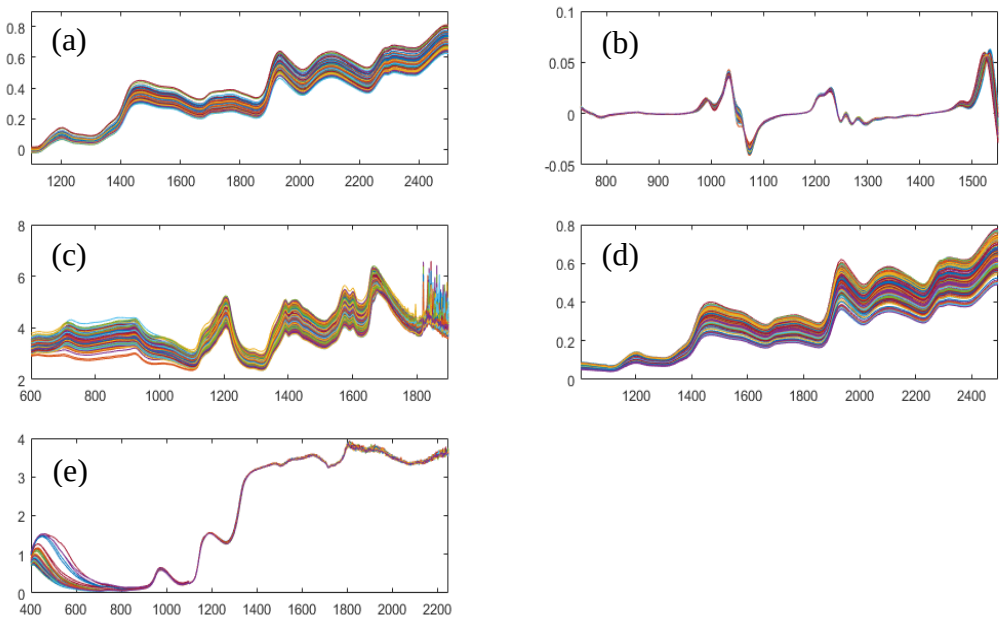


Figure 2.2 The spectra of five datasets. (a) corn dataset; (b) disel fuel dataset; (c) pharmaceutical tablet dataset; (d) wheat dataset; (e) beer dataset. For all subplots, x-axis is wavelength (nm) and y-axis is absorbance.

2.4.2 Performance metrics

To eliminate the potential bias caused by a specific partition of the whole dataset into calibration and testing subsets, a Monte Carlo validation and testing (MCVT) procedure that Shah et al. proposed previously is followed [24]. Specifically, I conduct 100 MC runs and use the results from all MC runs to evaluate the performance of each variable selection method. For each MC run, the calibration and testing subsets are randomly selected according to the percentage listed in Table 2.2.

The performance of different variable selection methods is assessed through three metrics. The first two are based on the soft sensor prediction performance, while the third directly evaluates the performance of variable selection through a consistency index.

I choose normalized root mean square error in prediction ($NRMSE_P$) to evaluate the prediction performance of different soft sensor models. The definition of $NRMSE_P$ is given in equation 2.20, where n_T is the number of samples of the test dataset in each MC runs. As shown in equation 2.20, the normalization in $NRMSE_P$ facilitates the comparison of different methods across different datasets.

$$NRMSE_P = \frac{\sqrt{\frac{1}{n_T}\sum_{i=1}^{n_T}(y_i - \hat{y}_i)^2}}{(y_{max} - y_{min})} \times 100\% \qquad (2.20)$$

In this work, the mean and the standard deviation of $NRMSE_P$ obtained from the 100 MC runs are used as the two metrics to evaluate the performance of different methods. The mean ($\overline{NRMSE_P}$) evaluates the accuracy of each method while the standard deviation (σ_{NRMSE_P}) assesses the robustness of the method [24].

To evaluate the consistency of the variable selection among different MC runs, I define a consistency index (I_c) as the following:

$$I_c = \frac{\sum_{i=1}^{m} prob(x_i)}{m'} \qquad (2.21)$$

where m' is the number of the variables (among all m variables) being selected at least once among all MC runs; $prob(x_i)$ is the probability of the i-th variable being selected, which is quantified by how frequently a variable is selected among all the MC runs. Clearly, a higher I_c indicates that the informative variables are more consistently selected regardless of calibration datasets.

It is also worth noting that different MC runs will result in different variables being selected due to different training samples being used and the stochastic nature of all "survival of the fittest" based variable selection methods. When these selected variables are used to build PLS models, the principal components (PC's) will be different for different MC runs. It is also

possible that the number of PC's will be different as it is determined through 10-fold cross-validation. The goal of MCVT is to compare different variable selection methods through the accuracy (*i.e.*, the average of the 100 $NRMSE_P$'s) and precision or robustness (*i.e.*, the standard deviation of the 100 $NRMSE_P$'s) of each method. A similar approach has been reported in the literature [67].

2.4.3 Results of CEEVS

To ensure a fair comparison, all methods being compared were optimized through 10-fold cross-validation. The tuning parameters for each method are listed in Table 2.3. For each method, the optimal tuning parameters were determined through exhaustive search within a specified range for the parameter.

Table 2.3 Tuning parameters that were optimized for each method

Methods	**Tuning parameters**
Full PLS	# of PC's
CARS	# of PC's, # of Monte Carlo sampling runs
SVP	# of PC's, # of iterations, sampling ratio of MCS-S[a] and MCS-P[b], # of sampling in MCS-S[a] and MCS-P[b]
GA	# of PC's, population size, # of iterations, crossover scheme, mutation rate, initial population, termination criterion
CEEVS[c]	# of PC's

[a]Monte Carlo sampling in sample space; [b]Monte Carlo sampling in variable space; [c]Other parameters are fixed as shown in Table 2.1.

Performance comparison

For each dataset, the variable selection and soft sensor prediction results from each method are tabulated in Table 2.4 – 2.8. The best performance corresponding to each metric is shown in boldface. In these tables, Improvement rate (%) refers to the improvement of $\overline{NRMSE_P}$ over that of the full PLS model, n_{PC} is the "mean ± std" of the number of principal components of the final soft sensor among 100 MC runs, n_{VAR} is the "mean ± std" of the number of selected variables among 100 MC runs, except full PLS where all variables are used.

Table 2.4 The performance comparison using the corn dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	9.197	2.390	-	-	11.6 ± 1.7	700
CARS	9.263	2.760	0.063	-0.72	12.3 ± 1.7	21.4 ± 8.2
SVP	9.569	2.602	0.062	-4.05	14.0 ± 0.9	25.9 ± 10.0
GA	8.730	2.337	0.119	5.07	9.0 ± 2.4	73.6 ± 27.2
CEEVS	**8.335**	**2.051**	**0.212**	9.37	9.1 ± 2.3	100.9 ± 39.2

Table 2.5 The performance comparison using the diesel fuel dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	2.38	**0.30**	-	-	12.3 ± 1.7	401
CARS	2.94	0.65	0.136	-23.54	13.1 ± 1.6	54.7 ± 55.1
SVP	2.32	0.43	0.150	2.71	13.6 ± 1.4	47.0 ± 13.9

GA	2.24	**0.30**	0.240	6.12	11.8 ± 1.7	92.0 ± 41.5
CEEVS	**2.20**	**0.30**	**0.432**	7.56	11.2 ± 1.8	123.4 ± 37.5

Table 2.6 The performance comparison using the pharmaceutical tablets dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	5.05	**0.76**	-	-	14.3 ± 2.5	650
CARS	4.72	0.84	0.064	6.50	15.1 ± 3.1	30.2 ± 15.0
SVP	4.85	0.83	0.104	3.85	18.5 ± 1.5	50.1 ± 25.8
GA	4.46	0.90	0.138	11.69	10.8 ± 3.0	69.1 ± 44.1
CEEVS	**4.45**	0.89	**0.231**	11.86	13.3 ± 2.4	91.9 ± 56.1

Table 2.7 The performance comparison using the wheat dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	3.614	**0.587**	-	-	15.9 ± 1.5	150
CARS	3.687	0.669	0.243	-2.02	15.2 ± 2.1	36.3 ± 13.0
SVP	4.011	0.685	0.151	-11.00	18.0 ± 1.7	21.8 ± 2.5
GA	3.502	0.595	0.286	3.08	10.7 ± 1.7	40.4 ± 13.6
CEEVS	**3.497**	0.624	**0.289**	3.22	11.2 ± 2.4	35.5 ± 11.4

Table 2.8 The performance comparison using the beer dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	6.57	6.46	-	-	9.1 ± 2.6	926
CARS	3.24	2.76	**0.192**	50.64	9.1 ± 2.6	86.8 ± 38.2
SVP	4.18	5.20	0.166	36.28	13.4 ± 2.1	113.0 ± 12.6
GA	2.37	1.85	0.142	63.91	7.8 ± 2.6	94.1 ± 58.0
CEEVS	**2.36**	**1.45**	0.182	64.11	8.1 ± 2.6	130.2 ± 85.9

As shown in the tables, across different datasets, CEEVS performs the best in almost all performance metrics. Specifically, among all 15 comparison instances (5 datasets × 3 performance metrics). In terms of $\overline{NRMSE_P}$, CEEVS performs the best for all 5 datasets; in terms of I_C, CEEVS performs the best for 4 of the 5 datasets and the 2nd best for the rest one; in terms σ_{NRMSE_P}, CEEVS performs the best for 3 of the 5 datasets, while slightly larger σ_{NRMSE_P} for the rest 2 datasets. These results indicate that by enhancing the consistency of variable selection, we can achieve better prediction performance.

Besides the quantitative metrics given in the tables, Figure 2.3 (a) and (b) compare the predicted vs measured quality variable for the diesel and beer datasets. From these two figures, it can be seen that the predictions of CEEVS stay the closest to the diagonal line, further indicating

the superior prediction accuracy and robustness.

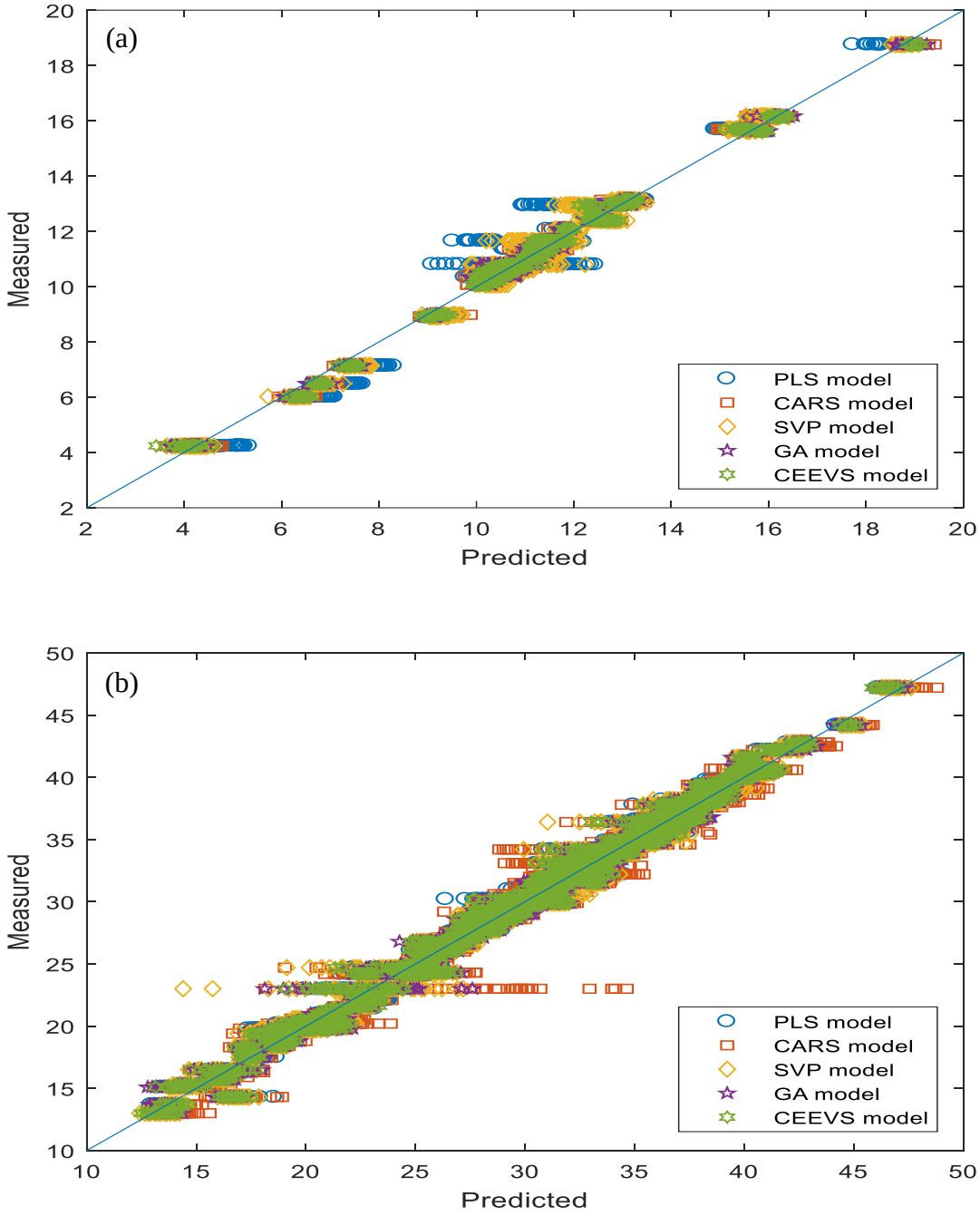


Figure 2.3 Plot of predicted vs. measured properties from five methods. (a) beer dataset; (b) diesel dataset

CEEVS can extract the underlying chemical information

One of the limitations of the existing variable selection methods based on "survival of fittest" is that the selected variables (wavelengths) for the soft sensor model may not have clear relationship with the chemical bounds or functional groups presenting in the sample. By enhancing the consistency of variable selection, I expect that CEEVS could identify the truly relevant variables that reveal the underlying chemical information. Further examination of the variable selection results from different methods confirmed our hypothesis.

Figures 2.4 and 2.5 plot the frequency of each variable being selected (denoted by the vertical thin bars) among all 100 MC runs for the corn dataset and the pharmaceutical tablets dataset for all four variable selection methods. The sample spectra (denoted by the red curves) are plotted on the same figures to visualize the portions of the spectra that are selected at high frequencies by different variable selection methods. These figures clearly show that CEEVS delivers the best consistency in terms of variable selection, as the variables that were selected from different runs are clustered together around spectrum peaks/valleys at high frequency, indicating high consistency. More importantly, further analysis show that the selected variables (corresponding to peaks or valleys) are associated with different chemical bonds/groups, which are labelled on the plot for the CEEVS method. The underlying chemical information revealed by the selected variables further support our claim that the selected variables with high consistency are likely the truly relevant ones.

In terms of variable selection frequency, GA performs similar to CEEVS, while the clustering of the selected variables may not be as clear and distinct as that from CEEVS. For CARS and SVP, although the number of variables being selected by these two methods are usually much smaller than those from GA and CEEVS, the consistency of variable selection is

much worse and as a result, the selected variables could reveal little underlying chemical information.

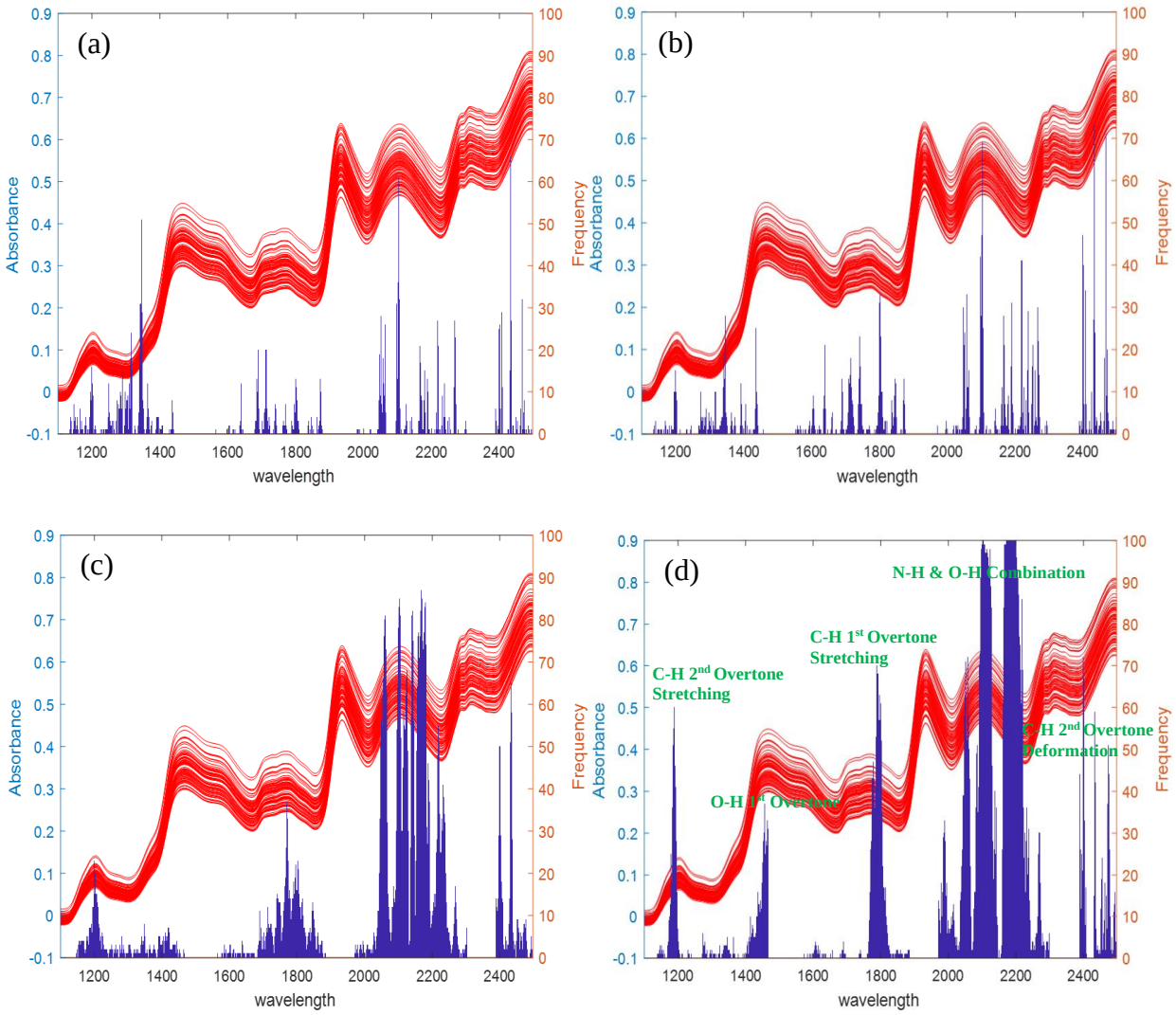


Figure 2.4 Plot of spectra (red curves) and histogram of selected wavelengths (blue vertical bars) over 100 MC runs for the corn dataset. (a) CARS; (b) SVP; (c) GA; (d) CEEVS

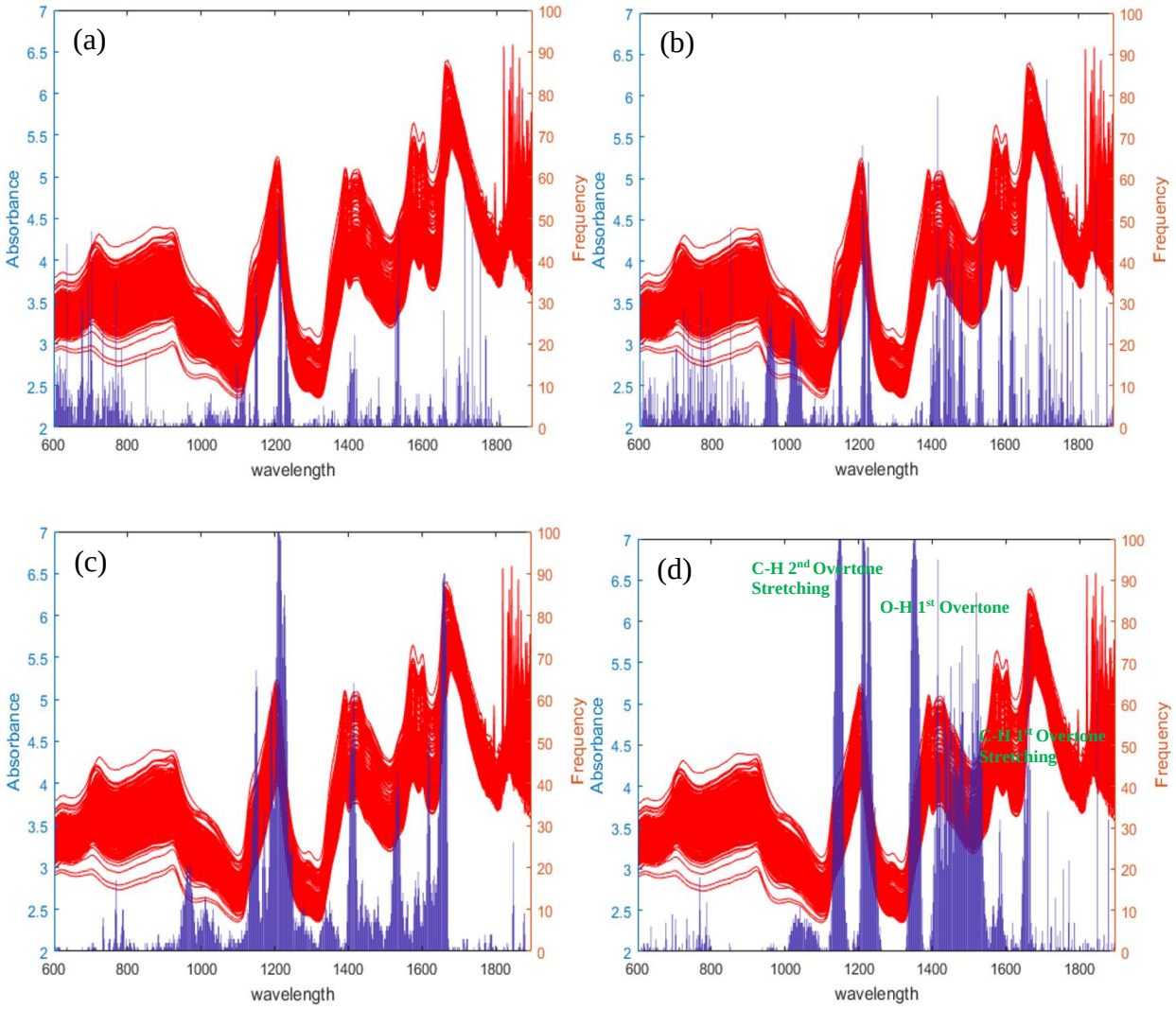


Figure 2.5 Plot of spectra (red curves) and histogram of selected wavelengths (blue vertical bars) over 100 MC runs for the pharmaceutical tablets dataset. (a) CARS; (b) SVP; (c) GA; (d) CEEVS

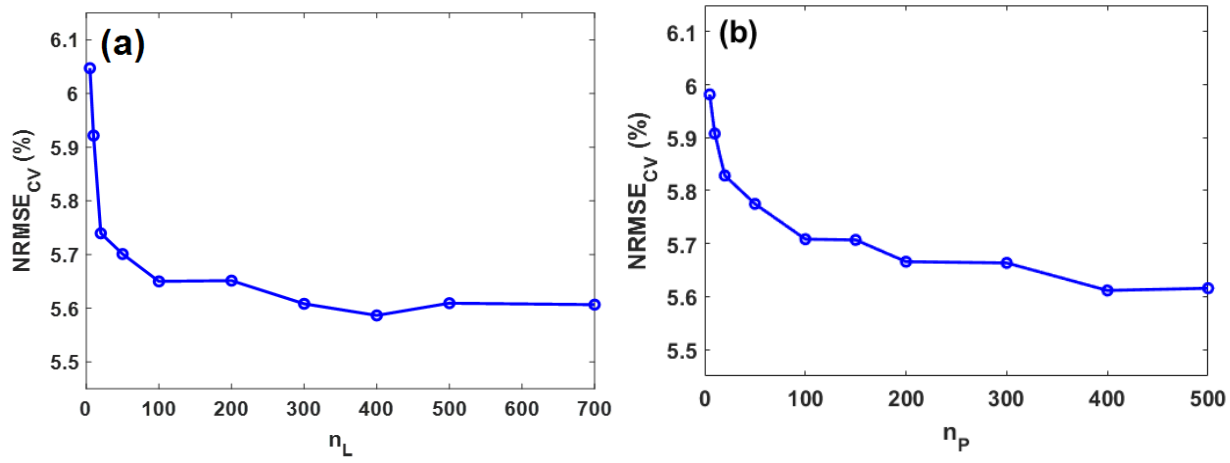
Robustness of CEEVS

CEEVS has four tuning parameters, the library size (n_L), the population size (n_P), the sampling ratio (γ) and the number of sampling runs (n_S). To examine the robustness of the method with respect to its tuning parameters, I test 10 different levels for each tuning parameter.

For the number of chromosomes in the library (n_L), the ten levels I tested were [5, 10, 20, 50, 100, 200, 300, 400, 500, 700]. The cross-validation results corresponding to the tested levels for the corn dataset is plotted in Figure 2.6 (a). The results for other datasets are very similar to

the corn dataset. Figure 2.6 (a) shows that as n_L increase, $NRMSE_{CV}$ initially decreases sharply; and then it stabilizes when n_L is sufficiently large. Because n_L determines the number of best performing chromosomes to be stored in the library, the initial increase in n_L allows more relevant variables to be stored in the library; however, as n_L increasing, the enhanced variable selection consistency delivered by CEEVS allows all truly relevant variables being selected, therefore, further increasing the number of repetitions does not result in further improvement in the model performance. Based on the testing of all datasets, in this work, I fix n_L at 200 for all the case studies.

For the size of population (n_P), the ten levels I tested were [5, 10, 20, 50, 100, 150, 200, 300, 400, 500]. The cross-validation results for the corn dataset is plotted in Figure 2.6 (b) and other dataset show very similar behavior. Similar to the case of n_L, as n_P increases, the cross-validation performance saw significant improvement initially, then levels off as n_P keep increasing. This is because the initial increase in n_P allows more chromosomes to be evaluated, thereby increasing the probability of producing superior offspring. However, after sufficient number of chromosomes have been evaluated, this effect diminishes. Based on the effect of n_P for all the datasets, I set n_P to 400 for all the case studies in this work.



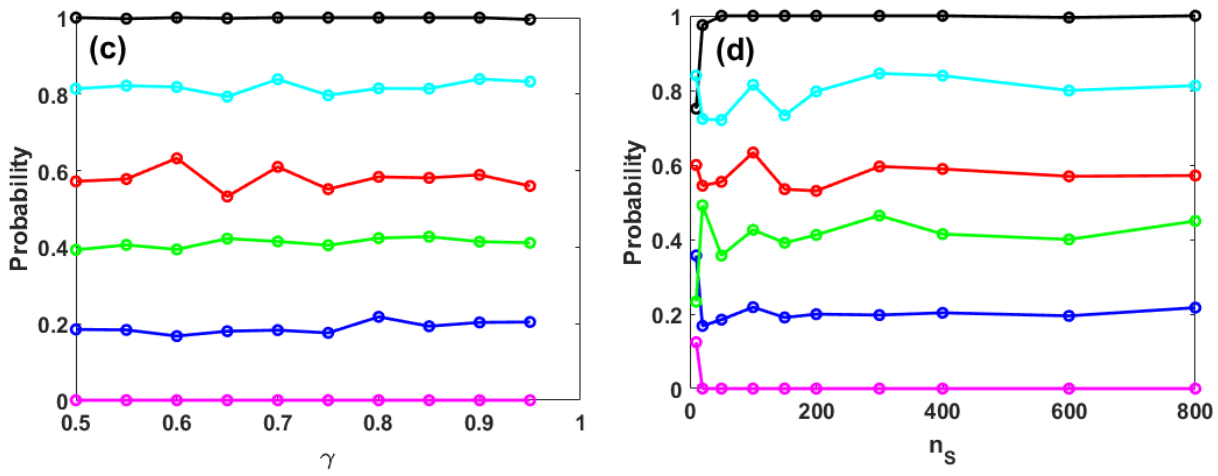


Figure 2.6 (a) The effect of n_L on performance for the corn dataset. (b) The effect of n_P on performance for the corn dataset. (c) The effect of γ on the initial selection probability of five representative variables (denoted by different lines) that have different levels of probability of selection. (d) The effect of n_S the initial selection probability of five representative variables (denoted by different lines) that have different levels of probability of selection.

The sampling ratio (γ) and the number of sampling runs (n_S) are involved in evaluating variable stability and probability for selection, so here I examine their effect on variable's probability for selection. I selected 5 representative variables that have different levels of probability for selectin. For γ, the 10 levels examined are [0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95], and for n_S, the 10 levels examined are [10, 25, 50, 100, 150, 200, 300, 400, 600, 800]. As shown in Figure 2.6 (c) and (d), similar to n_L and n_P, when γ and n_S are large enough, the probability for selection become quite insensitive to the tuning parameters. In this work, I choose γ = 0.9 and n_S = 400 for all case studies.

2.4.4 Discussion

It has been well documented that variable selection can help address several challenges associated with soft sensor development for spectroscopic datasets, namely: (1) variable multicollinearity, *i.e.*, variables are highly correlated; (2) highly noisy data; (3) curse of dimensionality, *i.e.,* the number of variables is larger than the number of samples. In addition, variable selection could improve model predictive accuracy by eliminating irrelevant predictor variables and provide a better understanding of the chemically important wavelength regions by reducing model complexity. However, variable selection methods can be sensitive to calibration data and their performance may be unstable. As shown in Tables 2.4 – 2.8, PLS soft sensors using variables selected by CARS and SVP delivered worse prediction performance compared to the full PLS soft sensor without variable selection for 3 out of the 5 datasets. More importantly, the low consistency of selected variables among different MC runs suggests that their performances are sensitive to the choice of the training samples. There are two possible reasons to explain such sensitivity. First, both CARS and SVP use the regression coefficients to define the stability of variables, which introduces significant variability in variable selection as regression coefficients are sensitive to the choice of the training samples. Second, both methods adopt EDF to remove the less important variables. Once the variables are eliminated based on their stability (which depends heavily on the training samples), they will not be re-evaluated. However, some previously eliminated variables could contribute significantly to prediction when variable combination changes.

To address these limitations, in CEEVS both regression coefficients and VIP scores are used to define the variable stability; and by using the frequency of a variable being stored in the library to rank the variables instead of using variable stability, CEEVS allows less important

variables to be evaluated in different combinations. In addition, unlike GA where the initial population is generated completely randomly, CEEVS uses variable stability to guide the generation of the initial population which favors the more important variables. Moreover, the evolution process in CEEVS is also guided by variable stability, which enables CEEVS to deliver much enhanced consistency in variable selection. I believe such enhanced consistency in variable selection suggests truly relevant variables are selected, as the underlying relationship between sample spectrum and sample property does not change across different training samples. As expected, the enhanced consistency in variable selection not only resulted in the improved soft sensor prediction performance, but also revealed key chemical information in the spectra. Finally, compared to GA, CEEVS significantly reduces the number of tuning parameters and deliver highly robust performance over a wide range of turning parameters. This is highly desirable as it makes the implementation of CEEVS significantly easier for practitioners and could be adopted easily for different applications.

2.5 Extension of CEEVS

2.5.1 Introduction to Statistics Pattern Analysis (SPA) feature-based soft sensor

SPA is originally developed for a process monitoring that He et al. proposed previously [45], [68], [69], in which the statistics/features of process variables are extracted along the time dimension. Instead of the process variables themselves, features of process variables are used to monitor process operation condition. SPA can address many process challenges such as dynamics, nonlinearity, non-Gaussianity and non-synchronized trajectories. In addition, since features can explain process characteristics much better than process variables themselves, the effectiveness of SPA in process monitoring has been demonstrated in many case studies [45], [68], [69]. In the SPA feature-based soft sensor, the statistics/features are extracted along the

variable (i.e., wavelength) dimension. Then, PLS is established to correlate statistics/features to response variable(s) (i.e., sample properties), instead of the raw spectroscopic reading of samples. Notice that sample-wise statistics/features are calculated for SPA based process monitoring approach, while variable-wise ones are calculated for SPA feature-based soft sensor. Figure 2.7 shows the schematic diagram of the SPA feature-based soft sensor approach. First, the whole spectrum is split into roughly equally spaced s non-overlapping segments. Then, f different features are extracted from each spectrum segment. In this work, feature pool consists of eight different statistics as following: mean (μ), standard deviation (σ), skewness (γ), kurtosis (κ), average of first derivative of spectrum over an interval (AFD), average of second derivative of spectrum over an interval (ASD), slope of linear regression line (SLL), and coefficient of squared term for second order regression line (SSL). μ and σ describe the center and dispersion of spectroscopic readings (i.e., absorbance) in each segment, respectively. γ, κ, SLL and SSL are the features associated with the shape of the spectrum in each segment. AFD and ASD are used to measure the rate of change of absorbance in each segment. The soft sensor model is established using the extracted features (total $s \times f$ features for each sample) and response variables. The predictor variables have $X_{n\times(s\times f)}$ where n is number of samples, and $(s \times f)$ is number of predictors. $y_{n\times 1}$ denotes a response variable vector (*i.e.*, property of interest) with dimension of $n \times 1$. The major advantages of SPA feature-based soft sensor are as follows: (1) it can utilize information from the whole spectrum with much smaller number of the features compared to wavelengths; (2) the features not only better capture spectral characteristics such as nonlinearity and peak shift, but also reduce the influence of the spectra noises and disturbances, which would improve the soft sensor's predictive power. More details about SPA feature-based soft sensor can be found in [24].

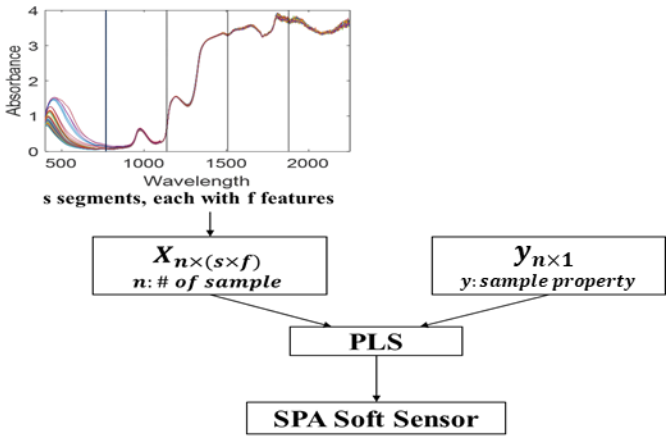


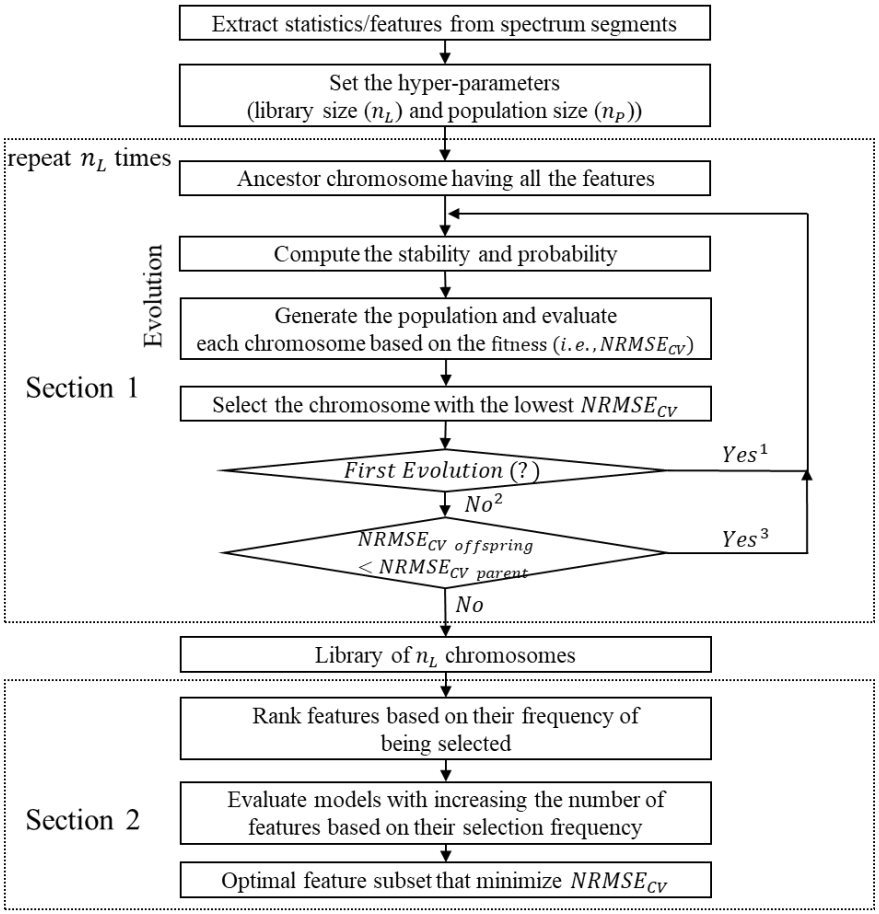
Figure 2.7 Schematic of SPA feature-based soft sensor

2.5.2 SPA feature-based soft sensor integrated with CEEVS (SPA-CEEVS)

In Section 2.4.3, I found that CEEVS usually select the largest number of wavelengths, and the selected wavelengths consistently cluster around spectrum peak or valleys, which is how the underlying chemical information is identified. This makes sense, because the general features of molecular spectra are of continuous bands, and the shape of the peak or valley, in addition to peak height, could contain important information about the underlying molecular structure. As the shape of the peak cannot be captured by a single wavelength, this is why a segment of wavelengths around a peak or valley were consistently selected by CEEVS. However, the wavelengths within the peak/valley segment are highly correlated and do contain many redundant information. If such information could be captured by different features, I do not have to include the whole segment of the wavelengths, reducing the number of predictors (wavelengths) without scarifying prediction performance. In this work, I propose to integrate SPA feature-based soft sensor with CEEVS to further simplify the soft sensor model through the feature selection.

In SPA-CEEVS, rooted in SPA feature-based soft sensing, I apply CEEVS to select relevant features, which are then used to build the soft sensor model. In this way, I could obtain a

significantly simplified model because CEEVS starts with smaller predictor pool (*i.e.*, statistics/features) compared to wavelengths. In addition, the proposed method could further enhance the prediction performance, as irrelevant features are removed through feature selection. Finally, the selected features could reveal chemically important information, leading to easier interpretation of results. As shown in Figure 2.8, SPA-CEEVS follows the CEEVS algorithm but instead of wavelengths, the statistics/features extracted from spectrum segments are used for feature selection.



1. The selected chromosome to be parent chromosome.
2. The selected chromosome in previous evolution to be parent chromosome.
   The selected chromosome in current evolution to be offspring chromosome.
3. The offspring to be parent chromosome.

Figure 2.8 Flow diagram of the SPA-CEEVS algorithm

2.5.3 Results of SPA-CEEVS

Table 2.9 summarizes the tuning parameters for SPA related methods and nonlinear methods. For a fair comparison, all methods except for GPR are optimized through 10-fold cross-validation. For GPR method, the optimal tuning parameters are obtained by maximizing the marginal likelihood. For other methods, exhaustive search is used to find the optimal tuning parameters.

Table 2.9 Tuning parameters for comparison methods

Methods	**Tuning parameters**
Elastic Net	α, regularization parameter (λ)
SVR	regularization parameter, kernel scale parameter, ε, kernel function[a]
GPR[b]	Length scale for each predictor, signal standard deviation, noise standard deviation
SPA	# of segments, # of PC's
SPA-CEEVS[c]	# of segments, # of features/Statistics, # of PC's

[a]linear and Gaussian; [b]Automatic Relevance Determination (ARD) squared exponential kernel is used for covariance function; [c]Other parameters are fixed as specified in Table 2.1.

Performance comparison

In this work, the comparison results of SPA-CEEVS, feature engineering integrated with feature selection, with other variable selection methods based on the raw predictors (*i.e.*, wavelengths) are listed in Table 2.10 – 2.14 to show that feature engineering can improve the predictive power. In addition, the comparison results of SPA-CEEVS rooted in linear PLS method with the nonlinear methods are presented in Table 2.15 – 2.19 to show that feature engineering and selection can explain nonlinear relationship between predictor and response

variables. Then, I demonstrate how the features can capture the spectral characteristics such as nonlinearity through investigating the relationship between the features and the sample properties. The best performance for each metric is denoted in boldface. Details about each metric are explained in Section 2.4.3. The SPA and SPA-CEEVS utilize all the features extracted from whole spectrum as described in Section 2.5.1, while the other methods use the raw predictor variables (*i.e.*, wavelengths).

Table 2.10 Performance comparison of feature selection between with feature engineering and without feature engineering using the corn dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	9.197	2.390	-	-	12 ± 2	700
CARS	9.263	2.760	0.063	-0.72	12 ± 2	21 ± 8
SVP	9.569	2.602	0.062	-4.05	14 ± 1	26 ± 10
GA	8.730	2.337	0.119	5.07	9 ± 2	74 ± 27
CEEVS	8.335	2.051	0.212	9.37	9 ± 2	101 ± 39
Elastic Net	12.151	2.288	**0.824**	-32.12	-	563 ± 13
SPA-CEEVS	**8.025**	**2.008**	0.282	**12.74**	10 ± 3	46 ± 20

Table 2.11 Performance comparison of feature selection between with feature engineering and without feature engineering using the diesel dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	2.38	0.30	-	-	12 ± 2	401
CARS	2.94	0.65	0.136	-23.54	13 ± 2	55 ± 55
SVP	2.32	0.43	0.150	2.71	14 ± 1	47 ± 14
GA	2.24	0.30	0.240	6.12	12 ± 2	92 ± 42
CEEVS	**2.20**	0.30	0.432	7.56	11 ± 2	123 ± 38

Elastic Net	2.21	**0.29**	0.419	7.22	-	139 ± 10
SPA-CEEVS	2.21	0.34	**0.480**	7.45	13 ± 2	110 ± 36

Table 2.12 Performance comparison of feature selection between with feature engineering and without feature engineering using the pharmaceutical tablets dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	5.05	**0.76**	-	-	14 ± 3	650
CARS	4.72	0.84	0.064	6.50	15 ± 3	30 ± 15
SVP	4.85	0.83	0.104	3.85	19 ± 2	50 ± 26
GA	4.46	0.90	0.138	11.69	11 ± 3	69 ± 44
CEEVS	4.45	0.89	0.231	11.86	13 ± 2	92 ± 56
Elastic Net	4.87	0.80	0.289	3.49	-	91 ± 23
SPA-CEEVS	**4.40**	0.92	**0.321**	**12.88**	9 ± 2	39 ± 15

Table 2.13 Performance comparison of feature selection between with feature engineering and without feature engineering using the wheat dataset

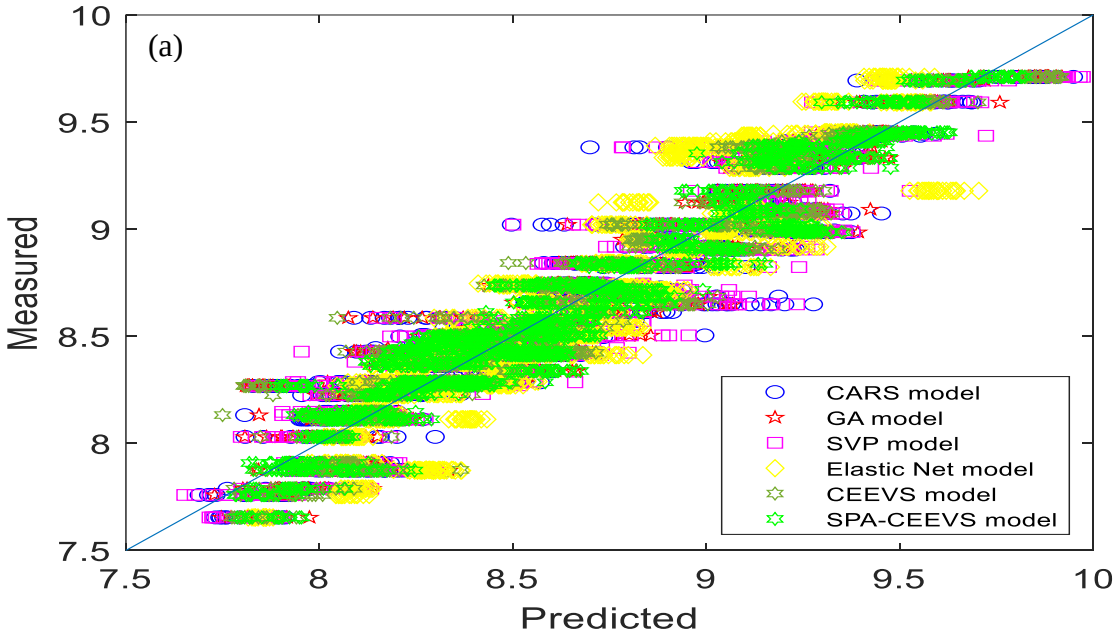
Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	3.614	0.587	-	-	16 ± 2	150
CARS	3.687	0.669	0.243	-2.02	15 ± 2	36 ± 13
SVP	4.011	0.685	0.151	-11.00	18 ± 2	22 ± 3
GA	3.502	0.595	0.286	3.08	11 ± 2	40 ± 14
CEEVS	3.497	0.624	0.289	3.22	11 ± 2	36 ± 11
Elastic Net	5.806	0.846	**0.862**	-60.69	-	124 ± 3
SPA-CEEVS	**2.755**	**0.502**	0.357	**23.77**	12 ± 2	50 ± 14

Table 2.14 Performance comparison of feature selection between with feature engineering and without feature engineering using the beer dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	I_C	Improvement rate (%)	n_{PC}	n_{VAR}
Full PLS	6.57	6.46	-	-	9 ± 3	926
CARS	3.24	2.76	0.192	50.64	9 ± 3	87 ± 38
SVP	4.18	5.20	0.166	36.28	13 ± 2	113 ± 13
GA	2.37	1.85	0.142	63.91	8 ± 3	94 ± 58
CEEVS	2.36	1.45	0.182	64.11	8 ± 3	130 ± 86
Elastic Net	3.38	3.12	0.197	48.58	-	64 ± 14
SPA-CEEVS	**1.65**	**1.15**	**0.288**	**74.87**	9 ± 3	24 ± 13

As shown in Tables 2.10 – 2.14, across different datasets, SPA-CEEVS offers the best in almost all performance metrics. In terms of $\overline{NRMSE_P}$, SPA-CEEVS performs the best for 4 datasets and the 2nd best for the rest one; in terms of I_C, SPA-CEEVS performs the best for 3 of the 5 datasets and the 2nd best for the rest 2 datasets. For corn and wheat dataset, Elastic Net has the largest I_C but it has poor ability to identify truly relevant variables associated with chemical functional groups, resulting in worse predictive accuracy than Full PLS model. In terms of σ_{NRMSE_P}, SPA-CEEVS performs the best for 3 of the 5 datasets, while slightly larger σ_{NRMSE_P} for the rest 2 datasets. Further investigation is conducted to understand why SPA-CEEVS has larger σ_{NRMSE_P} compared to full PLS method for pharmaceutical tablets dataset. All the $NRMSE_P's$ of SPA-CEEVS in 100 MC runs are lower than those of full PLS method, which enables SPA-CEEVS to have the lowest $\overline{NRMSE_P}$. However, for some MC runs, SPA-CEEVS has small improvement rate of the predictive accuracy, which makes the proposed method have comparatively larger deviation from $\overline{NRMSE_P}$ than full PLS method does. Therefore, SPA-

CEEVS has larger σ_{NRMSE_P} than full PLS method. It is worth noting that compared to CEEVS, SPA-CEEVS further improves the predictive power for almost all the datasets. These results indicate that SPA-CEEVS, feature engineering integrated with feature selection can achieve better prediction performance than other variables selection methods without feature engineering. Besides the quantitative metrics given in Tables 2.10 – 2.14, Figure 2.9 provides the detailed comparison of the prediction performance from the variable selection methods for 100 MC runs. The predicted value by SPA-CEEVS clustered the closest to the diagonal line given different training data, demonstrating superior prediction accuracy and robustness than other variable selection methods without feature engineering.



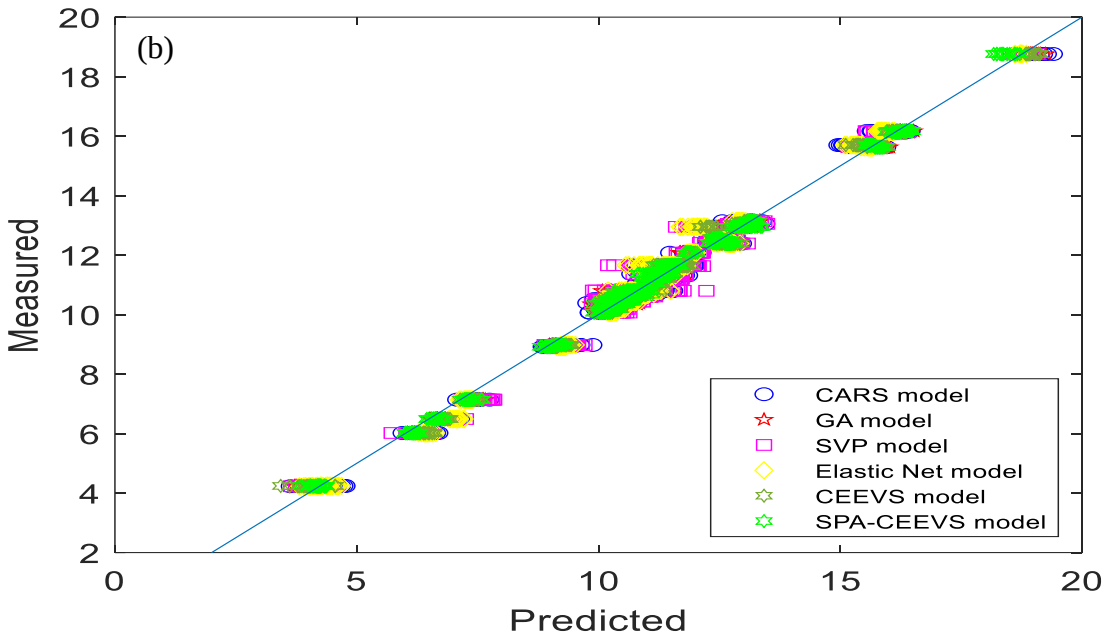


Figure 2.9 Plot of predicted vs. measured properties from variable selection methods. (a) Corn dataset; (b) Beer dataset.

Table 2.15 Performance comparison between linear and nonlinear methods using the corn dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	9.197	2.390	-	12 ± 2	700
SVR(linear)	9.258	2.495	-0.67	-	700
SVR(Gaussian)	10.657	2.466	-15.88	-	700
GPR	8.093	**1.741**	12.00	-	700
SPA	9.413	2.186	-2.35	11 ± 3	184
SPA-CEEVS	**8.025**	2.008	**12.74**	10 ± 3	46 ± 20

Table 2.16 Performance comparison between linear and nonlinear methods using the diesel dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	2.38	**0.30**	-	12 ± 2	401

SVR(linear)	2.30	0.54	3.44	-	401
SVR(Gaussian)	2.38	0.62	-0.01	-	401
GPR	2.37	1.30	0.59	-	401
SPA	**2.11**	**0.30**	**11.27**	14 ± 2	232
SPA-CEEVS	2.21	0.34	7.45	13 ± 2	110 ± 36

Table 2.17 Performance comparison between linear and nonlinear methods using the pharmaceutical tablets dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	5.05	**0.76**	-	14 ± 3	650
SVR(linear)	4.68	0.89	7.18	-	650
SVR(Gaussian)	4.61	0.91	8.59	-	650
GPR	4.69	0.91	7.13	-	650
SPA	4.61	0.82	8.59	12 ± 3	168
SPA-CEEVS	**4.40**	0.92	**12.88**	9 ± 2	39 ± 15

Table 2.18 Performance comparison between linear and nonlinear methods using the wheat dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	**Improvement rate (%)**	n_{PC}	n_{VAR}
Full PLS	3.614	0.587	-	16 ± 2	150
SVR(linear)	3.734	0.69	-3.32	-	150
SVR(Gaussian)	2.643	0.519	26.85	-	150
GPR	**2.602**	0.491	**28.00**	-	150
SPA	2.755	**0.480**	23.76	15 ± 2	184
SPA-CEEVS	2.755	0.502	23.77	12 ± 2	50 ± 14

Table 2.19 Performance comparison between linear and nonlinear methods using the beer dataset

Method	$\overline{NRMSE_P}$	σ_{NRMSE_P}	Improvement rate (%)	n_{PC}	n_{VAR}
Full PLS	6.57	6.46	-	9 ± 3	926
SVR(linear)	6.58	6.47	-0.20	-	926
SVR(Gaussian)	6.58	6.48	-0.23	-	926
GPR	2.60	1.61	60.36	-	926
SPA	3.69	2.70	43.85	11 ± 3	216
SPA-CEEVS	**1.65**	**1.15**	**74.87**	9 ± 3	24 ± 13

As shown in Tables 2.15 – 2.19, SPA-CEEVS has better prediction performance compared with other linear methods, Full PLS and SPA for almost all the datasets. In addition, SPA-CEEVS is much simpler model than SPA, which approximately include 10% – 50% of features SPA utilizes, resulting in easier interpretation of results. More interestingly, although SPA-CEEVS employs the linear PLS model, it can deliver comparable performance with the nonlinear methods, SVR and GPR. Figures 2.10 and 2.11 answer how linear model-based SPA-CEEVS can explain the nonlinear relationship between predictor variables and response variable. Figure 2.10 shows that for almost all the variables, their correlation coefficients are less or equal to 0.5, indicating that each variable is not strongly correlated with the response variable. However, As shown in Figure 2.11, the features can have large linear relationship with the response variable. Especially, higher-order statistics (HOS) such as $\gamma, \kappa,$ and SSL tend to have large correlation coefficients, indicating features can capture the spectral characteristics such as nonlinearity. In addition, during the process of the generation of the features, a large amount of the variability of the absorbance can be reduced via background and baseline shift correction [70]–[72], resulting in improving the relationship between the features and response variable.

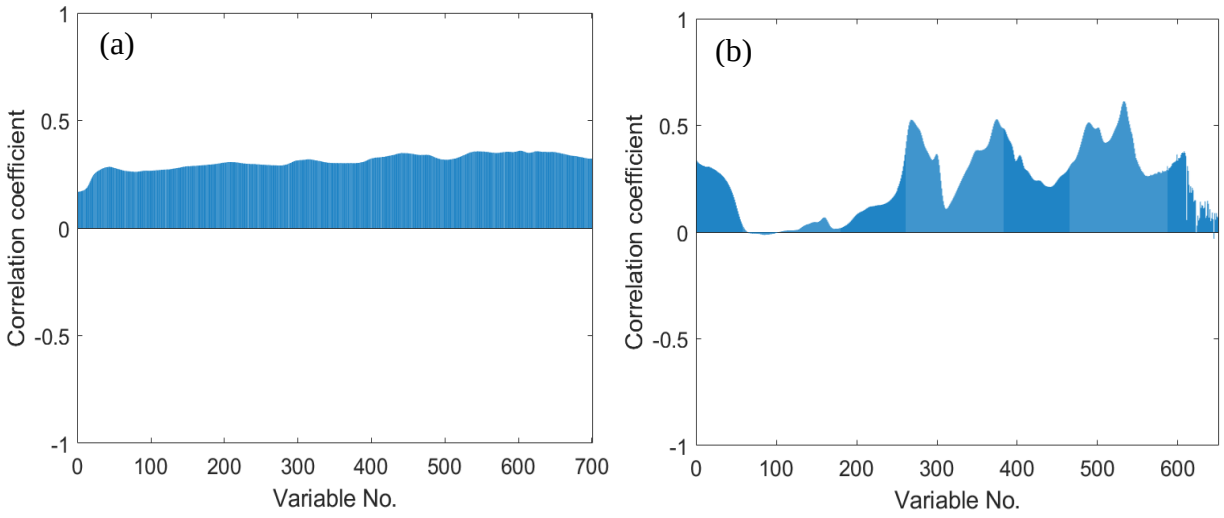
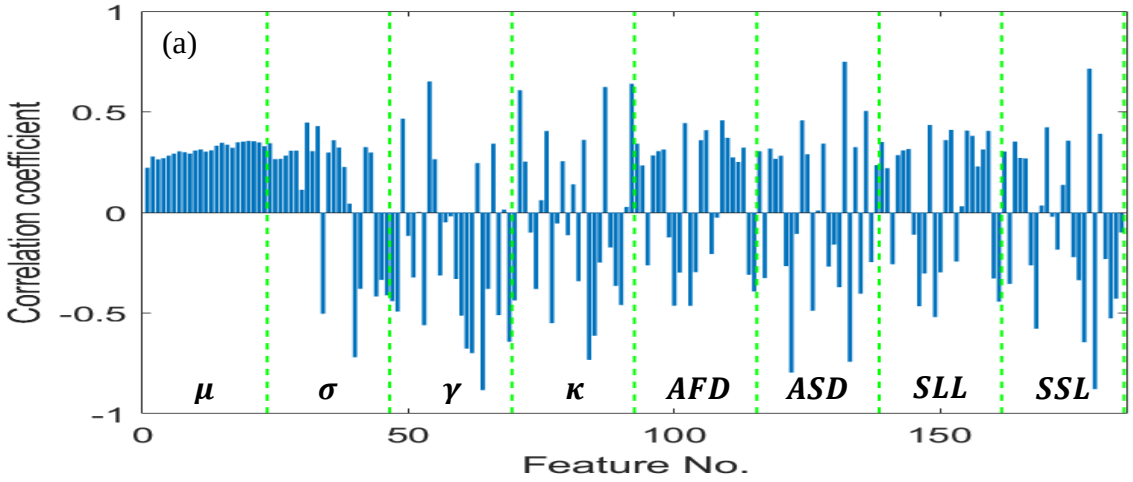


Figure 2.10 Correlation coefficients between predictor variables (wavelengths) and response variable (sample property). (a) corn dataset; (b) pharmaceutical tablets dataset.



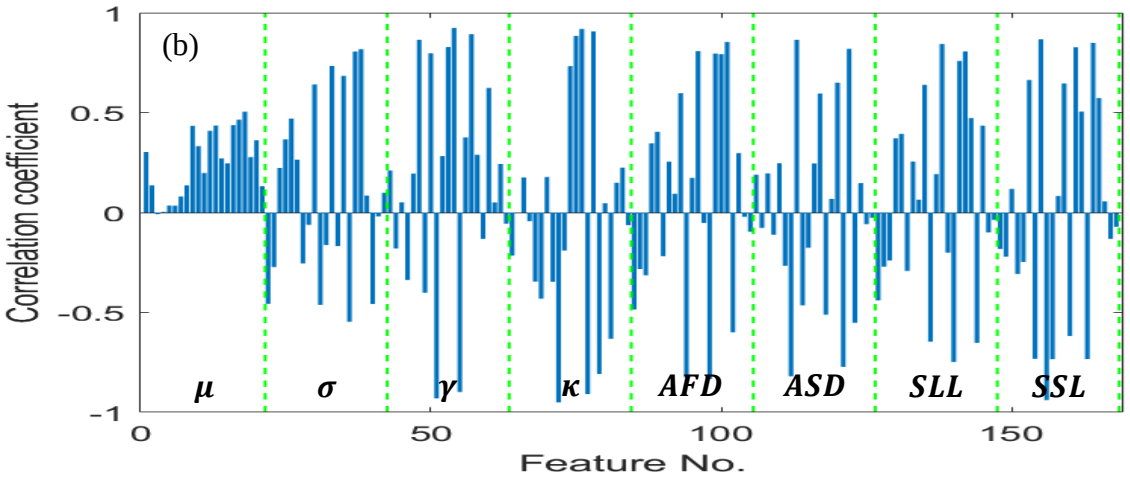
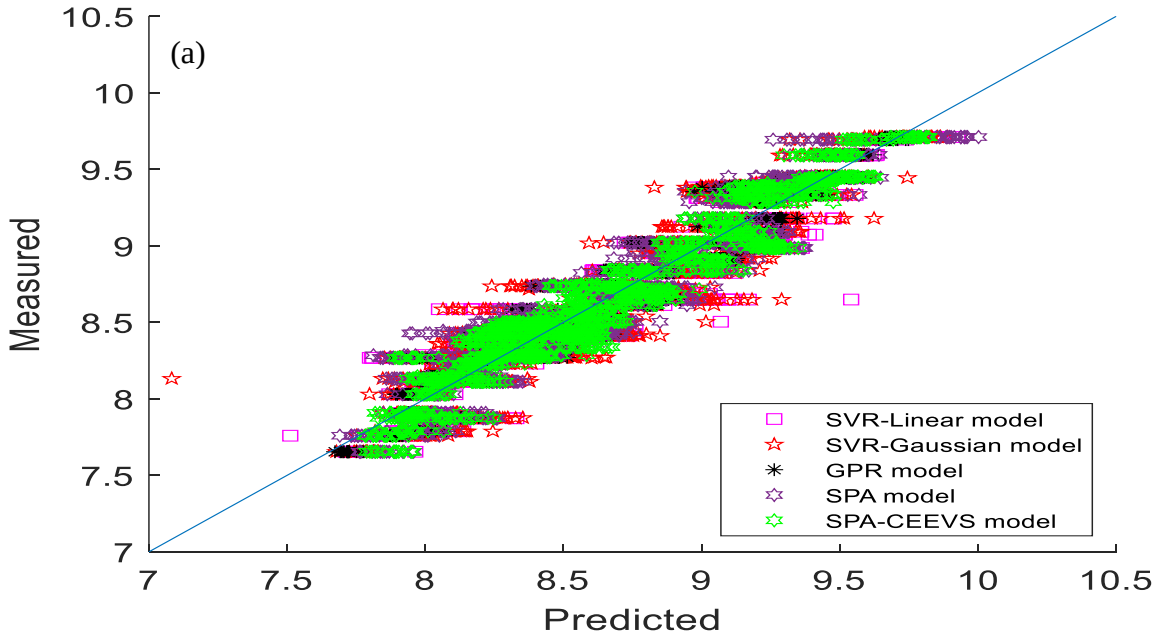


Figure 2.11 Correlation coefficients between features and response variable (sample property). (a) corn dataset; (b) pharmaceutical tablets dataset. The dotted lines split feature zones. Each bar represents correlation between a feature extracted from each segment and response variable.

Besides the quantitative metrics given in Tables 2.15 – 2.19, Figure 2.12 panels (a) and (b) compare the predicted vs measured quality variable for the corn and beer datasets. These figures show that the predictions of SPA-CEEVS stay the close to the diagonal line, further indicating the comparable prediction accuracy and robustness, compared to nonlinear methods.



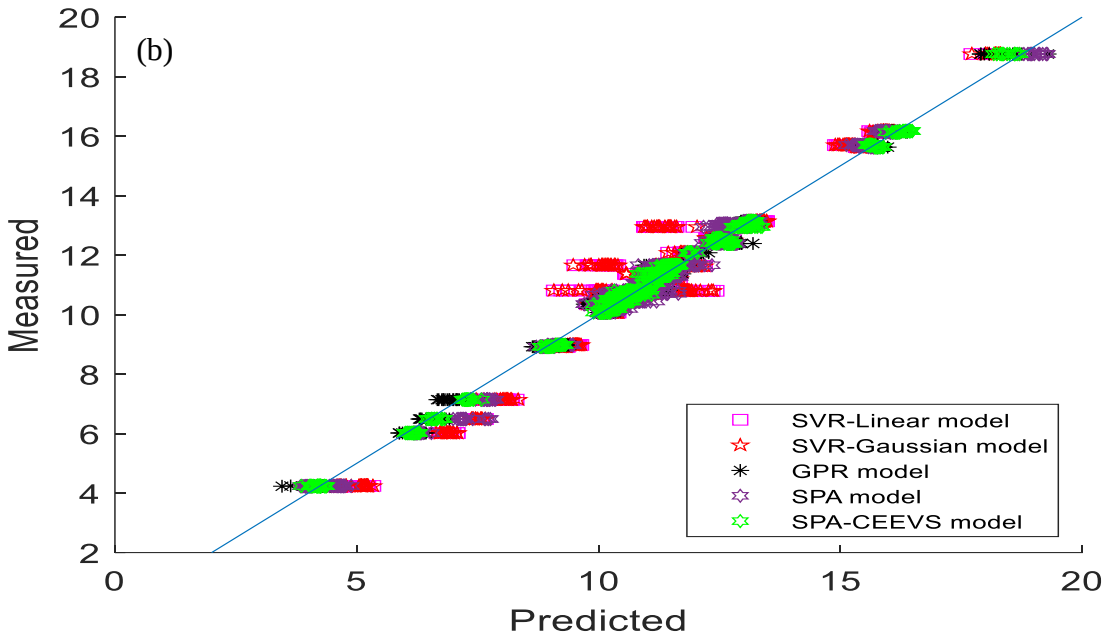


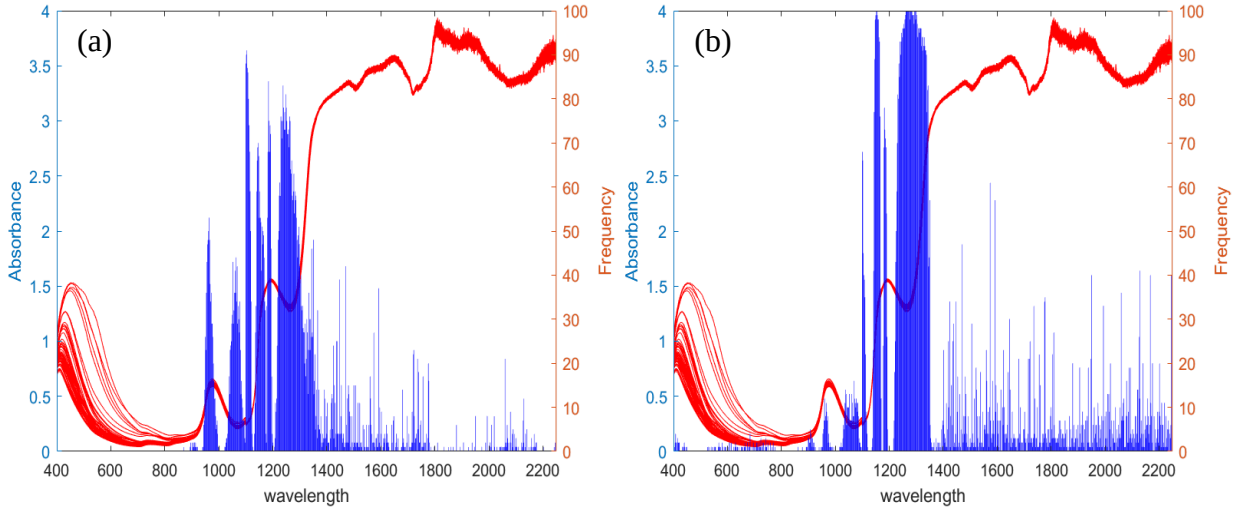
Figure 2.12 Plot of predicted vs. measured properties from linear-based and nonlinear methods. (a) corn dataset; (b) beer dataset.

In Section 2.4.3, I found that one major advantage of CEEVS is that it could identify the truly relevant variables associated with underlying chemical bond or functional groups through improving the consistency of variable selection. The variable selection frequency plots are used to demonstrate if this characteristic is conserved for SPA-CEEVS.

Figures 2.13 and 2.14 panels (a) – (e) show the frequency of selected variables (*i.e.*, wavelengths) represented by thin blue bars among all 100 MC runs for the beer dataset and the pharmaceutical tablets dataset for the five variable selection methods without feature engineering. On the other hand, Figures 2.13 and 2.14 (f) plot the frequency of selected features extracted from each segment and the dotted line represents each segment. These figures also include the sample spectra denoted by the red curves. Like CEEVS, SPA-CEEVS can deliver the high consistency in terms of feature selection. It is worth noting that the spectrum segments where SPA-CEEVS frequently selected the features are almost matched with the regions where

CEEVS consistently selected the variables. Considering that the selected variables by CEEVS are related to chemical functional groups, SPA-CEEVS also can reveal the underlying chemical information and the selected features with high frequency are more likely to be the informative predictor variables.

The spectral absorbances at all the wavelengths are used as predictor variables, instead of manually eliminating the noisy spectral regions before modeling because this study aims to demonstrate the usefulness of the proposed method by illustrating the proposed method does not select the noisy spectral regions. As shown in Figures 2.13 and 2.14, SPA-CEEVS rarely selects the features extracted from noisy spectral regions (1800 – 2250nm for beer dataset, 1800 - 1898nm for pharmaceutical tablets dataset). However, although Elastic Net provides the high consistency of variable selection, the approach selects many variables associated with noise, resulting in poor predictive accuracy. For GA, CARS and SVP methods, the clustering of the selected variables may not be as clear and distinct as that from SPA-CEEVS, even CEEVS, which result in the limitation of improvement of the predictive power.



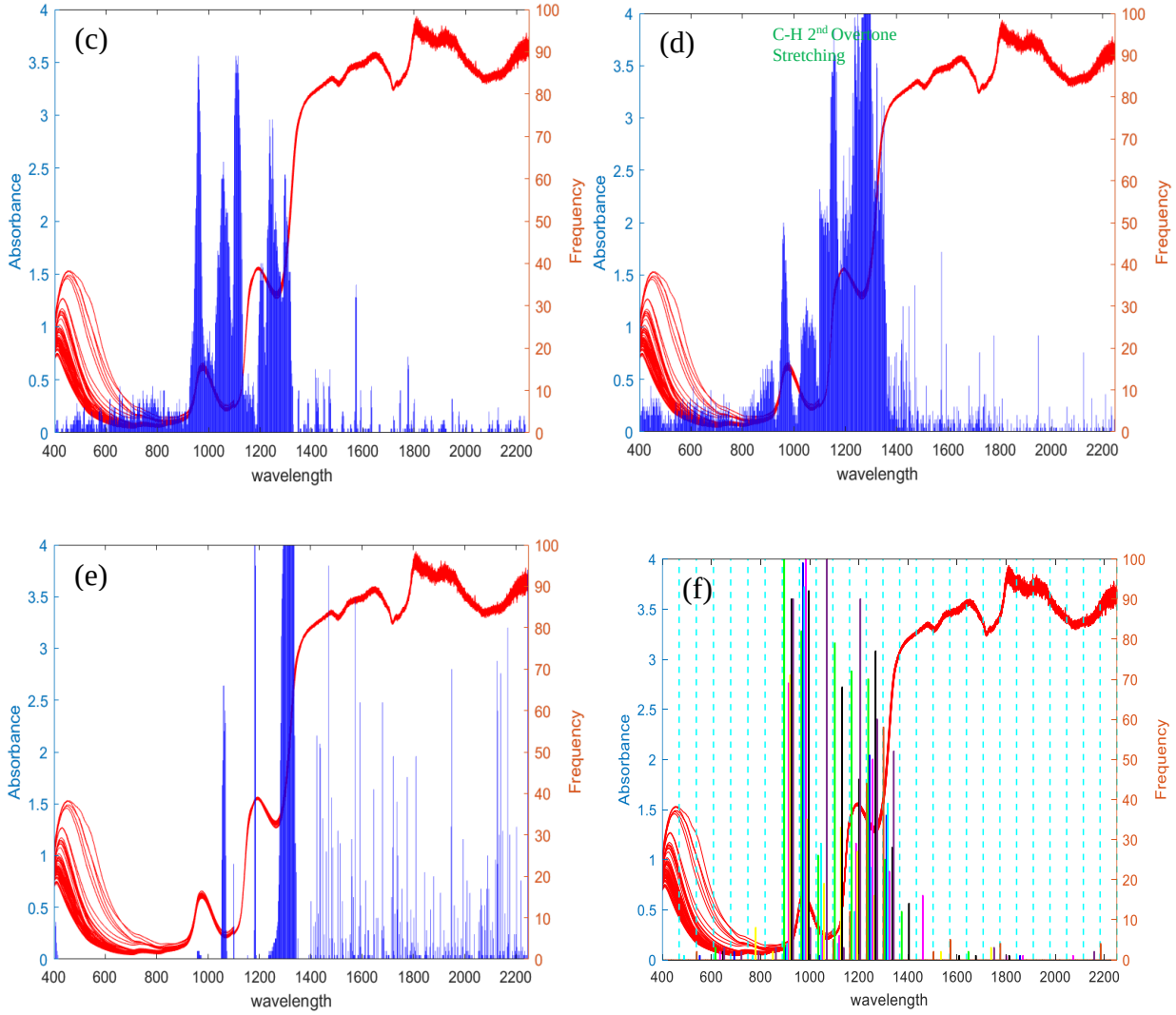
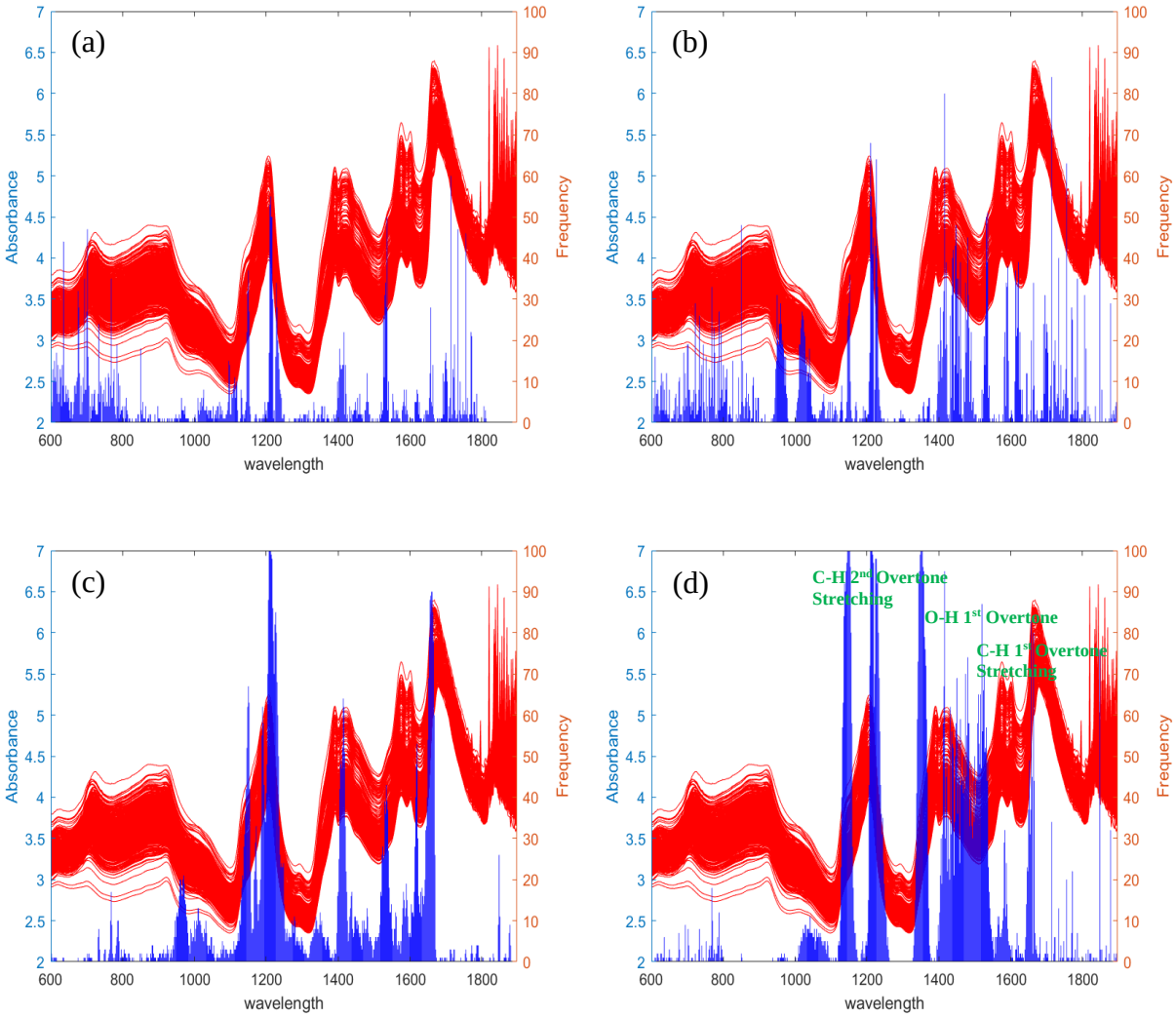


Figure 2.13 Plot of spectra (red curves) and selected wavelengths/features (vertical bars) over 100 MC runs for the beer dataset. (a) CARS; (b) SVP; (c) GA; (d) CEEVS; (e) Elastic Net; (f) SPA-CEEVS. In the SPA-CEEVS, the bars with different colors correspond to different features (brown: μ, green: σ, blue: γ, bright blue: κ, pink: AFD, yellow: ASD, black: SLL, purple: SSL). The dotted line denotes each segment.


(a)
(b)
(c)
(d)
Absorbance
Frequency
wavelength
C-H 2nd Overtone Stretching
O-H 1st Overtone
C-H 1st Overtone Stretching

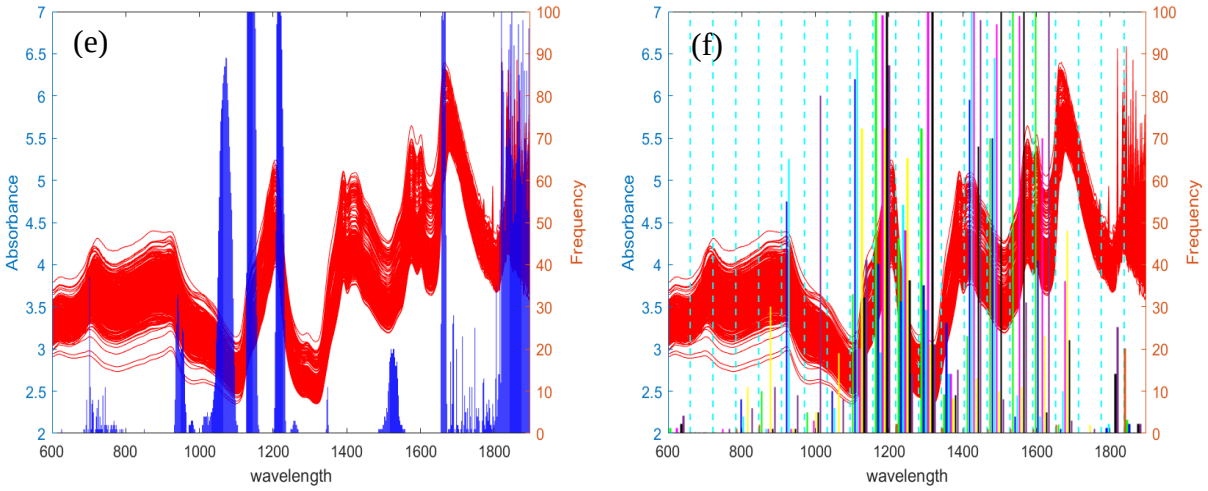


Figure 2.14 Plot of spectra (red curves) and selected wavelengths/features (vertical bars) over 100 MC runs for the pharmaceutical tablets dataset. (a) CARS; (b) SVP; (c) GA; (d) CEEVS; (e) Elastic Net; (f) SPA-CEEVS. In the SPA-CEEVS, the bars with different colors correspond to different features (brown: μ, green: σ, blue: γ, bright blue: κ, pink: AFD, yellow: ASD, black: SLL, purple: SSL). The dotted line denotes each segment.

2.5.4 Discussion

In this section, I discuss the potential reasons SPA-CEEVS can further improve the performance, compared to 1) existing variable selection methods with no feature engineering and 2) nonlinear methods. SPA feature-based soft sensor was developed to address the limitation of the existing variable selection methods; variable selection can be strongly influenced by the choice of the calibration set. They often fail to identify truly relevant variables, resulting in poor predictive accuracy for new samples. SPA feature-based soft sensor can overcome this limitation by using all the information contained in the whole spectrum.

Although the feature extraction significantly reduces the number of features compared to the number of predictor variables (*i.e.*, wavelengths), SPA method still does not solve the curse of dimensionality (*i.e.*, the number of features is larger than the number of calibration samples). SPA-CEEVS removes irrelevant features based on the fact that all the features do not equally contribute to sample properties, thereby addressing the curse of dimensionality. Tables 2.10 –

2.14 show that the number of the selected features by SPA-CEEVS is less than the number of calibration samples for all the datasets.

The superior prediction performance of SPA-CEEVS in terms of both accuracy and robustness can be attributed to the following factors. First, the eight features extracted from each segment possibly account for more accurate relationship with sample properties than the raw predictor variables because they can reflect various aspects of spectra characteristics of the segment. According to Figures 2.10 and 2.11, the features could successfully capture nonlinear relationship between sample spectra and property of interest. Therefore, SPA-CEEVS provides comparable and/or better performance metrics compared to the nonlinear methods such as SVR and GPR though it utilizes a linear PLS model. Second, feature extraction plays a role of noise filtering, resulting in reducing the influence of noise or disturbance. For example, when I extract the features such as mean or standard deviation, all the spectral readings within each segment are used. Therefore, the effect of potential noise can be reduced, which makes the model less sensitive to noise, compared to the model using the spectral readings of samples. Lastly, the selection of truly relevant features results in more accurate and robust prediction performance.

In terms of consistency of variable selection, Elastic Net is also a good model for all the datasets, but it does not provide satisfactory predictive power especially for corn and wheat datasets. Tables 2.10 and 2.13 show Elastic Net has the largest I_C (over 0.8), indicating almost all the variables are consistently selected among 100 MC runs, while the predictive accuracy is the worst among the all the methods. This is because Elastic Net fails to select the truly relevant variables, leading to poor prediction for new samples. One of potential reasons Elastic Net does not work well for these datasets is that all the variables are extremely highly correlated with each

other (*i.e.,* all the variables have larger than 0.8 Pearson correlation coefficient), which lead to losing the capability of variable selection.

2.6 Summary and conclusions

In recent years, with an increase in demand for high product quality, spectral-based soft sensors have gained significant attention in various industries to predict the accurate properties of products with easy-to-measure process variables. Due to the characteristics of NIR spectra such as multicollinearity and nonlinearity, many PLS-based variable selection methods have been studied to deal with multicollinearity and to address the curse of dimensionality, and nonlinear methods such as ANN and SVR have been investigated to handle nonlinearity. However, these methods still have a couple of limitations. Specifically, PLS-based variable selection methods may result in unsatisfactory performance when spectra samples have a strong nonlinear relationship with properties of interest. Besides, the selected variables often show little connection to the chemical bonds or functional groups. CEEVS is developed to improve the accuracy and consistency of variable selection. I assume that if a variable selection method can consistently select relevant variables across different training samples, it would deliver better prediction performance. This is because truly relevant variables would not change depending on different training datasets. To enhance the consistency of variable selection, CEEVS uses both PLS regression coefficients (BETA) and variable importance in projection (VIP) to determine variable stability, which reduces the sensitivity to the training data while the probability of selection based on the variable stability ensures that even the variable of the minimum stability has a chance to be selected. The probability of selection based on the variable stability also ensures that the evolution process would start with a better initial population than GA where the initial population is completely randomly selected. This helps the evolution to converge to

optimum faster. In addition, the chromosome evolution process is also different from GA. By using the parent chromosome from previous evolution run as a new starting point to re-evaluate the variable stability and using the updated stability to determine the probability for offspring generation, the evolution process enhances the consistency of variable selection while eliminating non-informative variables. The choice of the final informative variable subset is based on the frequency of each variable being selected into the library of optimal chromosome. In this way, a variable with lower stability by itself yet still informative when combined with other variables would be included and evaluated.

CEEVS delivers the best consistency and predictive accuracy compared to other variable selection methods. However, predictive power could be further improved if features can capture the nonlinear relationship between spectral readings and properties of interest. Many studies show that nonlinear methods deliver better predictive accuracy than PLS-based variable selection methods. But the complexity of the nonlinear methods makes it difficult to understand key variables, which restrict a better understanding of the physical/chemical relationship between chemical functional groups and properties of interest. In addition, they may increase the risk of over-fitting for high dimensional data. Therefore, I present a novel soft sensor, SPA-CEEVS, that integrates SPA (*i.e.,* feature engineering) with CEEVS (*i.e.,* feature selection) to handle the nonlinearity and to improve the interpretation of results.

There are several benefits associated with SPA-CEEVS method. First, it can better explain the characteristics of spectra samples such as nonlinearity and peak shift through extraction of a variety of features from each segment. Second, there is an effect of noise filtering when features are calculated from each segment of whole spectrum. That is, the features can reduce the influence of the spectra noise and disturbance. Finally, it offers a simple model

through feature selection, which addresses the curse of dimensionality. In addition, the parsimony model makes it easy to understand key features for the prediction of the properties of interest.

The effectiveness of SPA-CEEVS is demonstrated by studying five different NIR datasets. These case studies show that SPA-CEEVS achieves better prediction performance than the absorbance-based variable selection methods. Besides, SPA-CEEVS delivers comparable and/or better prediction performance compared to the nonlinear methods. The results also show that SPA-CEEVS achieves high variable selection consistency regardless of datasets. In summary, SPA-CEEVS is a simple and reliable prediction model for all the datasets; it identifies key features associated with underlying chemical information and achieves better predictive accuracy.

Chapter 3. Feature space monitoring (FSM) for pressure swing adsorption (PSA) processes

3.1 Background

In the past few decades, pressure swing adsorption (PSA) processes have gained increasing commercial acceptance as an energy efficient separation technology [73]. PSA applications range from traditional bulk gas separation and drying, to CO_2 sequestration, trace contaminant removal, and other. With its extensive industrial applications, PSA has drawn significant research interests from the process systems engineering community recently. The research has focused mainly on PSA system modelling and simulation [74]–[76], design and optimization [73], [77], [78]. For process monitoring, traditional univariate approaches, such as the Shewhart [18], cumulative sum (CUSUM) [79], and exponentially weighted moving average (EWMA) [80] have been utilized in many industrial processes. The advantage of univariate statistical process control (SPC) charts is easy to implement. However, they have some limitations to improve the accuracy of fault detection on multivariate processes [81], [82]. With increasing product capability, modern PSA systems consist of 10 more adsorbers and hundreds of valves, from which tremendous process data is stored. It is not efficient to apply univariate approaches to PSA processes for process monitoring. To address the challenges, multivariate statistical process monitoring (MSPM) methods such as principal component analysis (PCA) and partial least square (PLS) have been studied and applied to traditional chemical batch processes, not PSA processes [83]–[86]. Although there is a significant need for effective MSPM methods to detect and diagnose PSA process abnormalities in real-time to avoid major production disruptions, research in this area has been scarce. This is mainly due to the non-stationary and

This chapter was excerpted from " Feature based fault detection for pressure swing adsorption processes " published in IFAC-PapersOnLine [94]. The author is the first author of these papers.

periodic nature of the process, which poses special challenges to monitoring such a process. For example, the application of the conventional MSPM methods, such as PCA and its variants, can lead to frequent false alarms and/or missed faults [87]. To address this challenge, Pan et al. (2004) proposed a process monitoring approach for continuous processes with periodic characteristics by identifying a stochastic state space model that captures the statistical behavior of changes occurring from period to period. The approach was validated using a waste water treatment process (WWTP). While there are similarities between WWTP and PSA processes, there are also differences. For example, the activated sludge process, which is a main part of a WWTP, is a natural periodic process with somewhat constant cycle time that is driven by the diurnal temperature and light fluctuations. In contrast, PSA is a forced periodic process with cycle time dynamically controlled to address many disturbances that affect the PSA operations (e.g., increased or decreased product throughput to meet customer demand or to minimize cost by scheduling based on electricity pricing, raw material feed composition variations), even weather conditions can affect the plant operations. As a result, the cycle time is heavily and frequently adjusted, which renders the approach proposed in [87] less effective for PSA processes. Another difference is that while the biological process in the WWTP is a very slow process, PSA is a very fast process. Recently, Wang et al. (2017) proposed a geometric framework for the monitoring and fault detection of periodic processes [88]. The proposed approach was applied to two simulated periodic processes with superior performance compared to the conventional dynamic PCA (DPCA) and multiway PCA (MPCA). For the simulated 2-bed PSA process, a total of 26 variables relating to the flow rate of the feed, as well as pressures and concentrations in and across both beds were used for observation. However, in industrial PSA processes, not all of these variables were measured, especially the concentrations in and across

the beds. In addition, pressure is the major process variable to be monitored, in this case the proposed method is not applicable as there is no centroid for a single variable. Another method specifically proposed for monitoring industrial PSA processes is a US patent [89]. In this method, a moving window discrete Fourier transform (DFT) was applied to process the data such as pressure profile. A number of "relevant" peaks were identified from the frequency spectra (i.e., their frequencies and amplitudes). Then calculate the logarithm of the amplitude ratio of peak k between beds i and j, which is defined as $\mathcal{R}$ in this work as the following.

$$\mathcal{R} = log\left(\frac{A_{i,k}}{A_{j,k}}\right) \tag{3.1}$$

where A is peak amplitude, i and j are the bed or vessel indices, k is the peak index. $\mathcal{R}$ is then monitored over time, where the control upper and lower limits were calculated based on normal operation data. However, there is no clear definition of "relevant" or how the "relevant" peaks are identified, which makes us hard to study the approach in this work.

For successful PSA processes monitoring, process fault detection methods should deal with multimodal operations of the process since cycle time is frequently adjusted to meet demand fluctuations. Many researchers have studied the multimode process fault detection methods, which can be classified into the method with multiple models and the method with a global model [90]–[94]. While the method with multiple models requires dividing modes and building a model for each mode, the method with a global model generates one universal model, which captures the characteristics of all the modes. In terms of algorithms, the multimode process fault detection methods can be divided into the Gaussian mixture based methods and the k-nearest neighbor (kNN) rule based methods [95]–[98].

In this work, I propose a novel process monitoring method, k-nearest neighbor-based feature space monitoring (FSM-kNN). The basic idea of the proposed approach is that instead of

monitoring the original pressure profile of a PSA process, I characterize the pressure profile of each PSA step by statistics and shape or morphology features. These features are then grouped by cycles and monitored by kNN rule for process monitoring (i.e., fault detection and diagnosis).

3.2 Introduction to PSA process

3.2.1 PSA process description

PSA is a well-known technique for the separation and purification of the mixture of process gases. There are wide industrial applications as follows: (1) gas drying, (2) solvent vapor recovery, (3) fractionation of air, (4) production of purified hydrogen, (5) separation of carbon dioxide and methane from landfill gas, (6) carbon monoxide – hydrogen separation, (7) normal isoparaffin separation, and (8) alcohol dehydration [99]. The PSA process mainly consists of two basic steps – adsorption and desorption – to separate and purify gas mixtures. During the adsorption step, the feed gas mixtures enter an adsorber, and impurities are selectively adsorbed on adsorbents at relatively high pressure, producing the purified product. In the desorption step following the adsorption step, the pressure of the adsorber is decreased so that impurities are desorbed from the absorbent pores, making the adsorbents be reused. In recent years, the PSA process includes two adsorbers to 10 or more adsorbers depending on the product capacity on the plant. These vessels are operated in periodic fashion, which indicates that some vessels are in the adsorption step to remove impurities at high pressure, the others are in the desorption step to release the trapped impurities at low pressure. Those steps are designed to optimize the product gas purity [99], [100].

3.2.2 PSA process characteristics

In this section, we discuss the unique characteristics of a PSA process and how these characteristics pose challenges to process monitoring. Figure 3.1 shows the typical pressure

profile of a multi-bed PSA process. Due to the sensitivity of the actual operation and production data of the process, all axis tick labels in this and other figures based on real operation data were omitted. To reduce clutter, only the pressure profiles from three beds are plotted. This type of pressure time series plot is good for visualizing and observing between-bed variations. However, only obvious deviations/faults can be observed from this type of plot and it can become very cluttered and difficult to read if all beds were plotted on the same figure. Figure 3.2 plots the overlapping of multiple cycles of a single bed, which can be used to visualize within-bed variations. Figure 3.3 plots the durations of the cycles over a period of time. A cycle consists of several steps for adsorption/desorption process. Figure 3.4 (a) and (b) show the specific step durations. About half of the steps follow similar trends as the cycle duration as shown in Figure 3.4 (a), while the other half were maintained at relatively constant as shown in Figure 3.4 (b).

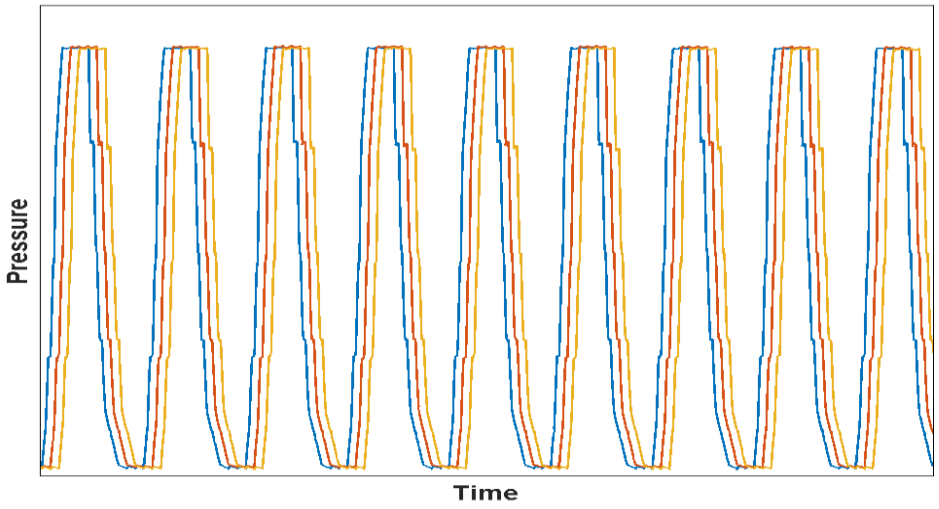


Figure 3.1 Typical pressure profiles of three beds in a multi-bed PSA process

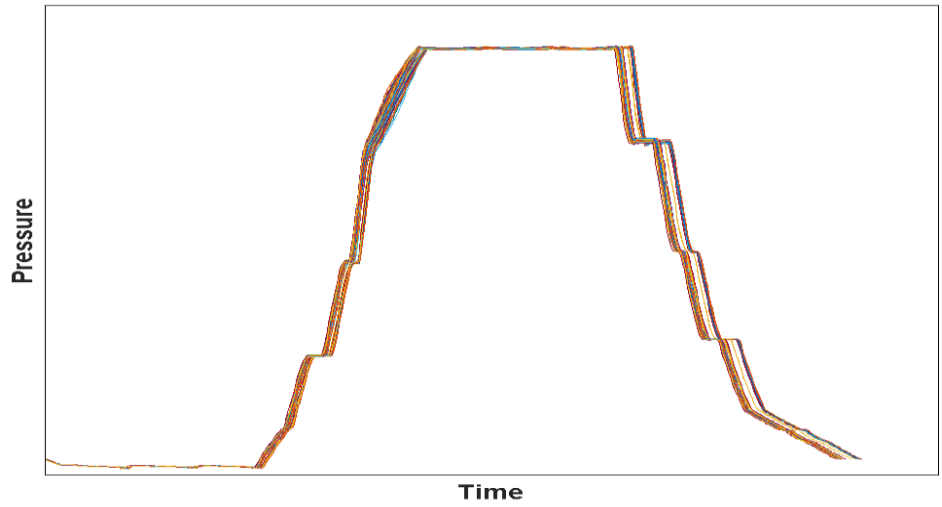


Figure 3.2 Overlapping pressure profiles of a single bed over multiple cycles

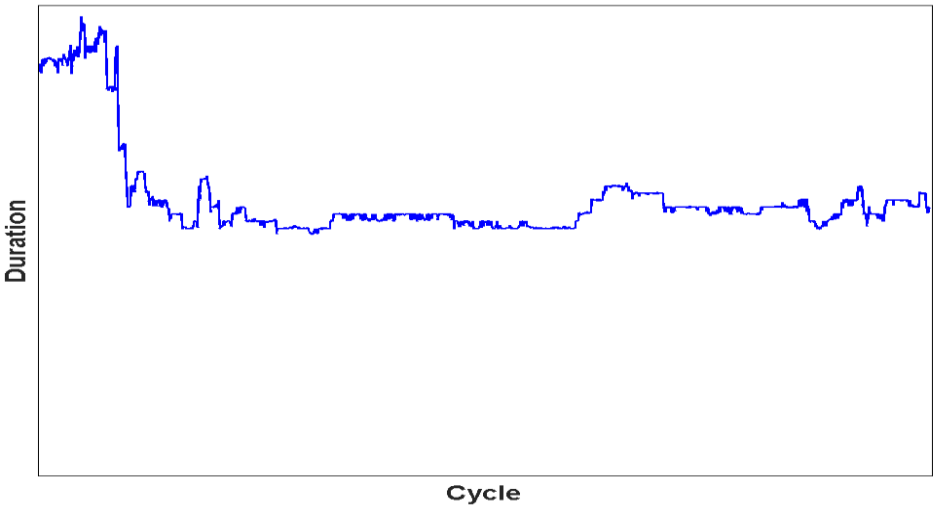
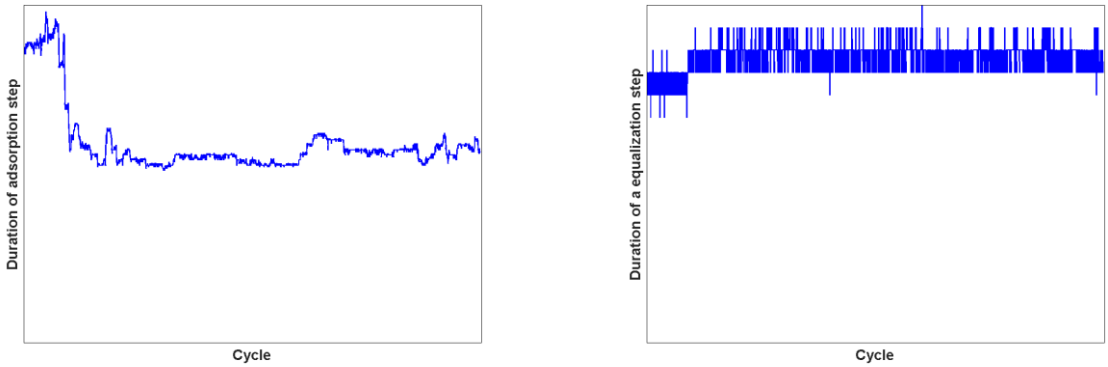


Figure 3.3 The cycle duration varies significantly from cycle to cycle



(a) (b)

Figure 3.4 (a) The step durations follow a similar trend as the cycle duration, or (b) are maintained at relatively constant

There are several points that can be made based on these three figures. First, the cycles are asynchronous across different beds as shown in Figure 3.1. They do not exactly overlap each other after unfolding for the same bed as shown in Figure 3.2, even for the onset, i.e., the start of the repressurization step, of the cycle. Second, the cycle duration, as well as the durations of the steps, vary from cycle to cycle as shown in Figures 3.3 and 3.4 (a). These durations are dynamically controlled to ensure product quality in response to dynamic scheduling, and/or disturbances such as demand change and weather conditions. Third, the process is highly nonlinear as shown in Figures 3.1 and 3.2. These characteristics pose significantly challenges to conventional MSPM methods such as MPCA, trilinear decomposition (TLD) and parallel factor analysis (PARAFAC) [101], or recently proposed methods such as multi-way independent component analysis (MICA) [102] and kernel PCA (KPCA) [103]. All these methods require the construction of a 2D or 3D array, which means that they all require synchronization of all steps of the entire cycle to equal step and cycle durations. This can be done through different ways, including simple cut, interpolation, dynamic time warping (DTW), etc. However, all these pre-

processing steps have their drawbacks, including trajectory distortion, information loss, etc. [19], [97]. It is particularly undesirable for PSA process because the step and cycle durations are dynamically controlled. As shown in Figures 3.3 and 3.4 (a), there are significant variabilities in step and cycle durations in a PSA process under normal operations. Therefore, the pre-processing steps mentioned above may result in unsatisfactory fault detection for the PSA process.

3.3 Review of preprocessing and process monitoring methods

In Section 3.2.2, I found that the cycle time is frequently adjusted to meet demand fluctuations. Therefore, the pre-processing is required to synchronize the different duration of the cycles and the steps. In this section, DTW is introduced as the pre-processing technique. For process monitoring, the traditional MPCA method and two kNN based fault detection methods are reviewed.

3.3.1 Dynamic Time Warping (DTW)

DTW warps the two asynchronous time series by compressing or expending them to make one resemble the other. It aims to find the optimal matching path between the two trajectories to achieve a minimum distance between them. DTW has been widely used in various fields such as speech recognition, process monitoring, prognostics, and imaging [104]–[107].

Let $X = (x_1, \cdots, x_N)$ and $Y = (y_1, \cdots, y_M)$ denote the two asynchronous time series. Assume i and j present the time index of the X and Y trajectories, respectively. DTW finds a warping path p of L points on a $N \times M$ grid.

$$p = [c(1), c(2), \cdots, c(k) \cdots c(L)],\ \max(N, M) \leq L \leq N + M \qquad (3.2)$$

$$c(k) = [i(k), j(k)] \qquad (3.3)$$

and each point $c(k)$ denotes an alignment between the two trajectories. There are several constraints that DTW should follow to find the best path – constraints on the endpoints of the path and local continuity constraints.

The endpoint constraints indicate that the end points of the two trajectories should be matched as following.

$$c(k) = (1, 1) \tag{3.4}$$

$$c(L) = (N, M) \tag{3.5}$$

The local continuity constraints consider physical considerations and do not allow excessive time compression and expansion. The monotonicity condition forces the path to be monotonous, which is expressed as

$$i(k+1) \geq i(k) \tag{3.6}$$

$$j(k+1) \geq j(k) \tag{3.7}$$

The requirement of avoiding extreme time compression and expansion is satisfied by constraining the local slope of the path to a specified range.

The goal of DTW is to find the best warping path through a grid of distances between two trajectories, which makes total distance minimum. The normalized total distance is defined as

$$D(N, M) = \frac{\sum_{k=1}^{L} d[i(k), j(k)] \cdot w(k)}{n(w)} \tag{3.8}$$

where $D(N, M)$ is a normalized total distance between the two asynchronous time series; $d[i(k), j(k)]$ is the weighted local distance between $i(k)$ and $j(k)$; $w(k)$ is a nonnegative weighting function; $n(w)$ is a normalization factor.

The best path is determined by solving the following equation.

$$D^*(N, M) = \min_{p}[D(N, M)] \tag{3.9}$$

$$p^* = \underset{p}{\operatorname{argmin}}[D(N,M)] \tag{3.10}$$

where $D^*(N,M)$ is the minimum normalized total distance; p^* is the best path. The $n(w)$ is used to make the normalized total distance independent of the number of path points L. More details about DTW can be found in [108].

3.3.2 Principal Component Analysis (PCA)

Let $\boldsymbol{X}_{n\times m}$ represent the data matrix with n samples and m variables. $\boldsymbol{X}$ is mean-centered to have zero mean for covariance-based PCA or is auto-scaled to have zero mean and unit variance for correlation-based PCA. The algorithms such as the NIPALS [109] or a singular value decomposition (SVD) can decompose the matrix $\boldsymbol{X}$ as follows:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^T + \widetilde{\boldsymbol{X}} = \boldsymbol{T}\boldsymbol{P}^T + \widetilde{\boldsymbol{T}}\widetilde{\boldsymbol{P}}^T = \left[\boldsymbol{T}\ \widetilde{\boldsymbol{T}}\right][\boldsymbol{P}\ \widetilde{\boldsymbol{P}}]^T \tag{3.11}$$

where $\boldsymbol{T}_{n\times l}$ and $\boldsymbol{P}_{m\times l}$ are the score and loading matrices, respectively. The PCA aims to find a set of l principal components (PCs), smaller than the m variables. The score matrix $\boldsymbol{T}$ is orthogonal. The loading matrix $\boldsymbol{P}$ consists of eigenvectors of the covariance or correlation matrix related to the l largest eigenvalues. The two metrics – the Hotelling's T^2 and the squared prediction error (SPE) – are usually used for process monitoring. The Hotelling's T^2 measures variations in principal component space (PCS):

$$T^2 = \mathbf{x}^T\boldsymbol{P}\boldsymbol{\Lambda}^{-1}\boldsymbol{P}^T\mathbf{x} \tag{3.12}$$

where $\boldsymbol{\Lambda}$ is the l largest eigenvalues of the covariance or correlation matrix. For a given significance level α, the process is considered normal if

$$T^2 \leq T_\alpha^2 \tag{3.13}$$

where the T_α^2 upper control limit is determined by using an F distribution or empirical way [45], [110]. Otherwise, the process is considered as an abnormal condition (i.e., $T^2 > T_\alpha^2$).

The SPE index reflects how much each sample deviates from normal process correlation and measures the projection of the sample vector on the residual space:

$$SPE = \|\tilde{\mathbf{x}}\|^2 = \|(\boldsymbol{I} - \boldsymbol{P}\boldsymbol{P}^T)\mathbf{x}\|^2 \quad (3.14)$$

For a given significance level α, the process is considered normal if

$$SPE \leq \delta_\alpha^2 \quad (3.15)$$

where the δ_α^2 upper control limit is determined by the method that Jackson and Mudholkar developed or empirical way [45], [111]. Otherwise, the process is considered as an abnormal condition (i.e., $SPE > \delta_\alpha^2$).

3.3.3 Multiway Principal Component Analysis (MPCA)

MPCA is a multivariate projection method to handle three-dimensional (3-D) array data. A batch process has 3-D array data that contains batch runs, variables, and time. As shown in Figure 3.5, the batch data can be configured as 3-D array $\underline{\boldsymbol{X}}(I \times J \times K)$, where I denotes batch runs, J is the number of variables, and K represents times throughout the batch. In MPCA, the 3-D array $\underline{\boldsymbol{X}}$ is unfolded into the 2-D array $\mathbf{X}$ to implement the PCA. In this approach, MPCA can identify variations from the normal operation trajectories.

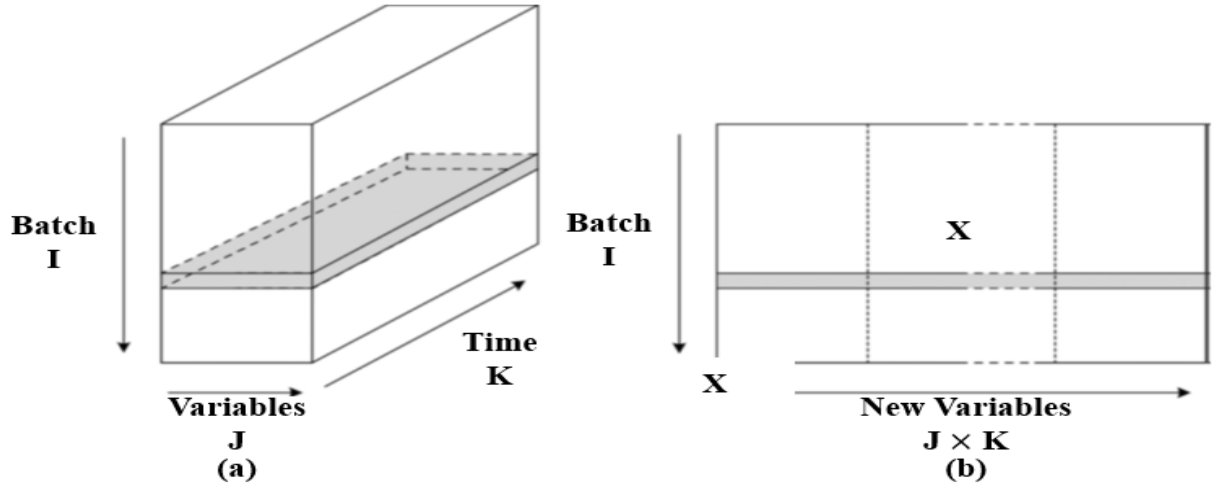


Figure 3.5 Data unfolding. (a) 3-D array data; (b) unfolded 2-D array data.

MPCA decomposes the matrix **X** as follows:

$$\boldsymbol{X}(I \times JK) = \boldsymbol{T}\boldsymbol{P}^{T} + \boldsymbol{E} \tag{3.16}$$

where $\boldsymbol{T}(I \times R)$ is the score matrix, $\boldsymbol{P}(JK \times R)$ is the loading matrix, $\boldsymbol{E}(I \times JK)$ is the residual matrix, and R is the number of principal components. The fault detection is performed the same as in Section 3.3.2. More details about MPCA can be found in [112]–[114].

3.3.4 k-Nearest Neighbor Rule-based Fault Detection (FD-kNN)

The FD-kNN is based on the hypothesis that the distance between a fault sample and its nearest neighboring normal samples in the training set is much larger than one between a normal sample and its nearest neighboring normal samples in the training set. The kNN squared distance of ith sample (D_i^2) is calculated as follows:

$$D_i^2 = \sum_{j=1}^{k} d_{i,j}^2 \tag{3.17}$$

where $d_{i,j}^2$ is the squared Euclidean distance between ith sample and jth nearest neighbor. For the fault detection, a threshold D_α^2 with a significance level α should be determined based on the distribution of the D_i^2, which can be estimated by a noncentral chi-square distribution. If the kNN squared distance of a new sample **x (**D_{x}^2) is smaller than or equal to the D_α^2 (i.e., $D_{\mathrm{x}}^2 \leq D_\alpha^2$), it is identified as a normal one. Otherwise, it is classified as an abnormal one. More details about FD-kNN can be found in [97].

3.3.5 Standardized k-Nearest Neighbor-based Fault Detection (SkNN)

SkNN is proposed to calculate the distance between a sample and its nearest training neighbors more accurate for multimode data by reducing the scale's difference of the variables within a mode and between the modes. As a first step, the multimode dataset $\boldsymbol{X}_{n\times m}$ (n: the number of samples, m: the number of variables) is auto-scaled to have zero mean and unit variance as following equation:

$$\boldsymbol{X}' = \frac{\boldsymbol{X} - \bar{\boldsymbol{X}}}{\sigma_{\boldsymbol{X}}} \tag{3.18}$$

where $\bar{\boldsymbol{X}}$ and $\sigma_{\boldsymbol{X}}$ represent the mean and standard deviation, respectively. A standardized distance is calculated as

$$SD\left(\boldsymbol{x}_i, \boldsymbol{x}_i^f\right) = \sqrt{\frac{[\boldsymbol{x}_i - \overline{N\left(\boldsymbol{x}_i^f\right)}]^T [\boldsymbol{x}_i - \overline{N\left(\boldsymbol{x}_i^f\right)}]}{\sigma_{N(\boldsymbol{x}_i)} \sigma_{N(\boldsymbol{x}_i^f)}}} \tag{3.19}$$

where $\boldsymbol{x}_i$ is the ith sample in $\boldsymbol{X}'$, $\boldsymbol{x}_i^f$ is fth neighbor of $\boldsymbol{x}_i$, $N(\boldsymbol{x}_i)$ is the k nearest neighbors of $\boldsymbol{x}_i$, $\overline{N(\boldsymbol{x}_i)}$ is the mean of $N(\boldsymbol{x}_i)$, $\sigma_{N(\boldsymbol{x}_i)}$ is the standard deviation of $N(\boldsymbol{x}_i)$. Once the standardized distance is computed, the accumulated distance is calculated as follows:

$$AD(\boldsymbol{x}_i) = \sum_{f=1}^{k} SD(x_i, x_i^f) \tag{3.20}$$

For the process monitoring, a threshold of AD with a significance level α can be calculated by the kernel density estimation (KDE). If the accumulated distance of a new sample $\mathbf{x}$ $AD(\mathbf{x})$ is smaller than or equal to the AD_α (i.e., $AD(\mathbf{x}) \leq AD_\alpha$), it is a normal one. Otherwise, it is detected as an abnormal one. More details about SkNN can be found in [98].

3.4 k-Nearest Neighbor-based Feature Space Monitoring (FSM-kNN)

The periodic processes have unique characteristics, such as a nonlinear and multimodal operation that make traditional MSPM methods challenging for fault detection of PSA processes. To address these challenges, I propose a k-nearest neighbor-based feature space monitoring (FSM-kNN) fault detection and diagnosis method for the PSA process. In the next section, I briefly review statistics pattern analysis (SPA), which is the predecessor of FSM-kNN, then introduce the FSM-kNN framework for PSA process monitoring.

3.4.1 Introduction to Statistics Pattern Analysis (SPA) for process monitoring

Statistics pattern analysis (SPA) was originally proposed for monitoring batch processes [45] and later extended to the monitoring of continuous processes and other applications such as soft sensor [24], [69]. Since then many variations and extensions have been proposed in the literature [115]–[118]. Because cyclic or periodic continuous processes share many similarities with batch processes (e.g., they are usually highly nonlinear processes with multiple steps or phases and their behaviors somewhat repeat from cycle to cycle or batch to batch), batch-based SPA is reviewed here.

Batch-based SPA hypothesizes that the batch behavior can be better characterized by the variance-covariance of batch statistics than by the variance-covariance of process variables. In SPA, a statistics pattern (SP) is a collection of various statistics calculated from a batch trajectory which 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). The basic idea of SPA is that the SPs of normal batches follow a similar pattern (i.e., normal pattern), while the SPs of abnormal or faulty batches must show some deviation from the normal pattern. More details on batch-based SPA can be found in [45].

3.4.2 Proposed fault detection framework (FSM-kNN)

As shown in Figure 3.6, the PSA processes consist of several steps whose functionality is different. Some steps are associated with adsorbing the impurities at high pressure, but the other steps are related to regenerating the adsorbents by decreasing the pressure and preparing for the next adsorption. The pressure in each step is a critical measurement to monitor the process

condition. Sometimes, the pressure profile is not sufficient to identify faults, which can lead to increasing false alarm rate and/or missing fault rate. To better explain the properties of process behavior in each step of cycles, I consider statistics as well as shape or morphological features. In this section, I will discuss the feature generation and the proposed fault detection and diagnosis. Figure 3.7 shows the flow diagram of FSM-kNN fault detection and diagnosis method.

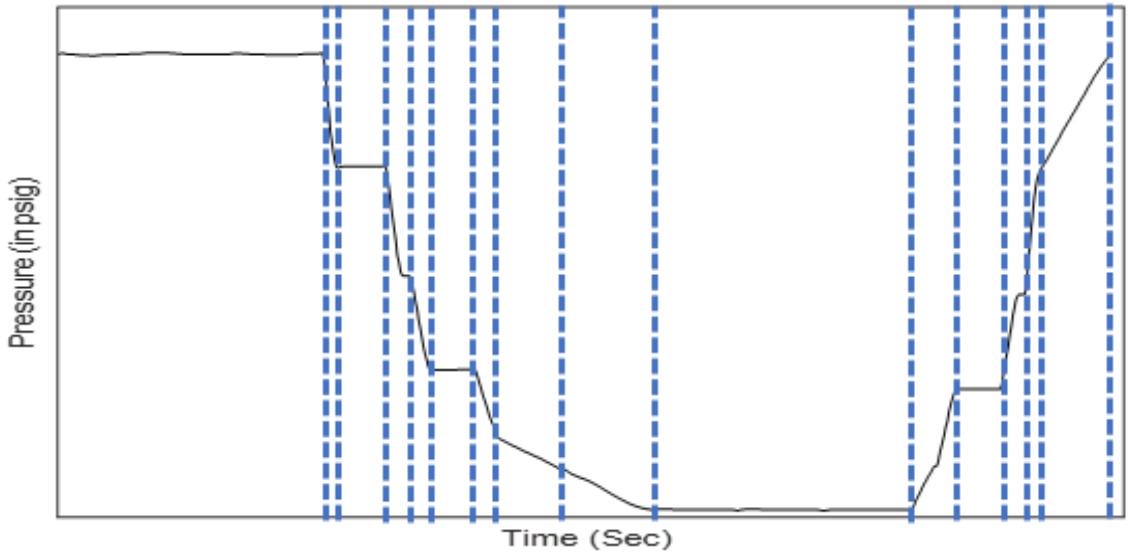


Figure 3.6 Pressure profile of PSA process. The dotted lines denote each step. (1) adsorption; (2) Equalization1 (3) Hold 1; (4) Equalization 2; (5) Equalization 3; (6) Hold 2; (7) Equalization 4; (8) Provide Purge; (9) Purge; (10) Blowdown; (11) Eqaulization 3-4; (12) Hold 3; (13) Equalization 2; (14) Equalization 1; (15) Repressurization.

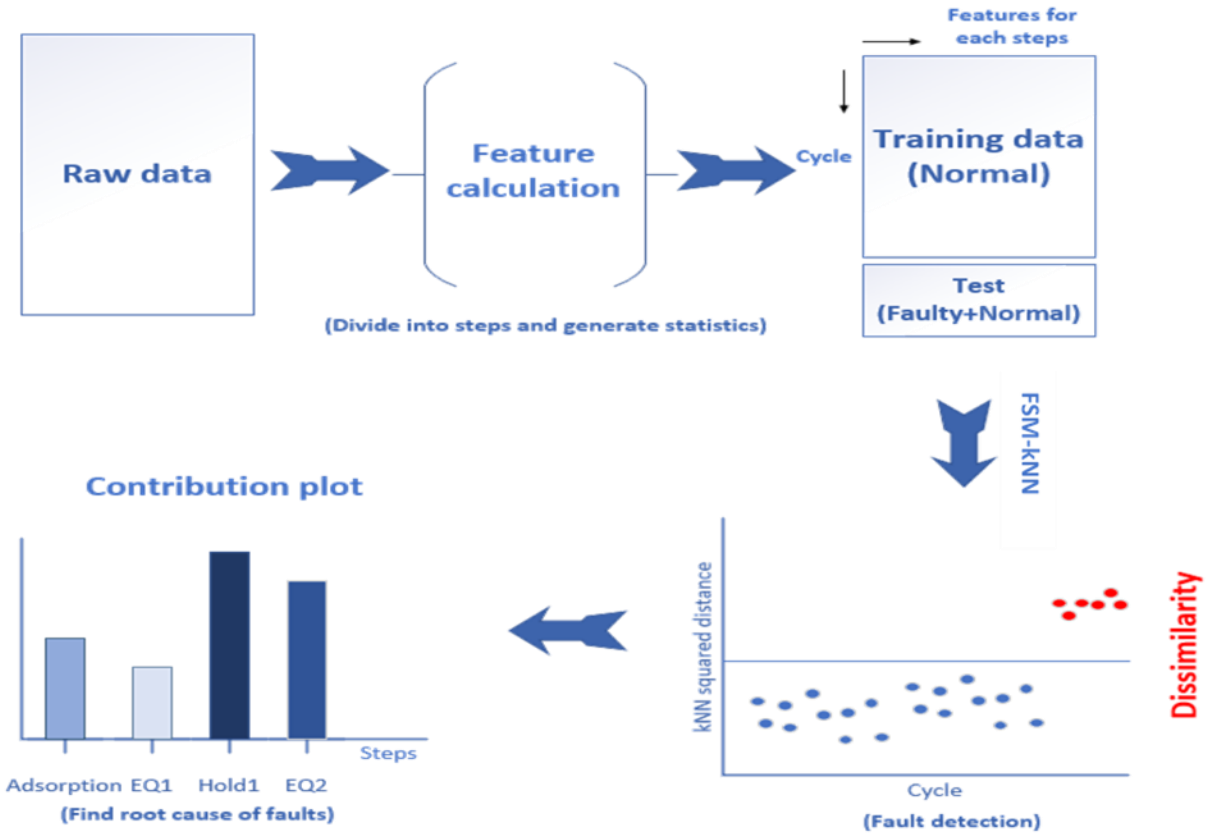


Figure 3.7 Flow diagram of FSM-kNN approach, which contains feature generation, fault detection, and fault diagnosis.

3.4.2.1 Notation

Let $\boldsymbol{x}_i \in \Re^{P_i}$ denote the P_i pressure measurements in ith cycle ($i = 1,2,\cdots,C$), where C is the number of cycles and P_i can vary depending on cycles. $\boldsymbol{x}_{i,j} \in \Re^{P_{i,j}}$ denotes the $P_{i,j}$ pressure measurements of the jth step ($j = 1,2,\cdots,T$) during the ith cycle, where T is the number of steps. In this work, there are total 15 steps (i.e., $T = 15$). $\boldsymbol{x}_{i,j}(t)$ denotes the tth pressure measurement during the jth step in the ith cycle ($t = 1,2,\cdots,N_{i,j}$). It is worth noting that $P_i = \sum_{j=1}^{T} P_{i,j}$ and $P = \sum_{i=1}^{C} P_i = \sum_{i=1}^{C}\sum_{j=1}^{T} P_{i,j}$. In this work, the following features are extracted from each step using original pressure measurements without any scaling or normalization.

3.4.2.2 Feature generation

Features

- Mean ($\mu_{i,j}$) is a measure of the central tendency of pressure distribution in the jth step of the ith cycle.

$$\mu_{i,j} = \frac{1}{P_{i,j}} \sum_{t=1}^{P_{i,j,}} x_{i,j}(t) \tag{3.21}$$

- Standard deviation ($\sigma_{i,j}$) is a measure of the dispersion of pressure distribution in the jth step of the ith cycle.

$$\sigma_{i,j} = \sqrt{\frac{1}{P_{i,j}} \sum_{t=1}^{P_{i,j}} (x_{i,j}(t) - \mu_{i,j})^2} \tag{3.22}$$

- Skewness ($\gamma_{i,j}$) is a measure of the asymmetry of pressure distribution in the jth step of the ith cycle.

$$\gamma_{i,j} = \frac{\frac{1}{P_{i,j}} \sum_{t=1}^{P_{i,j}} (x_{i,j}(t) - \mu_{i,j})^3}{\sigma_{i,j}^3} \tag{3.23}$$

- Kurtosis ($\kappa_{i,j}$) describes the shape of pressure distribution in the jth step of the ith cycle, which measures how heavily the tails of pressure distribution deviate from those of normal distribution.

$$\kappa_{i,j} = \frac{\frac{1}{P_{i,j}} \sum_{t=1}^{P_{i,j}} (x_{i,j}(t) - \mu_{i,j})^4}{\sigma_{i,j}^4} - 3 \tag{3.24}$$

- Root mean square ($RMS_{i,j}$) is a measure of the magnitude of pressure measurements in the jth step of the ith cycle.

$$RMS_{i,j} = \sqrt{\frac{1}{P_{i,j}} \sum_{t=1}^{P_{i,j,}} x_{i,j}^2\,(t)} \tag{3.25}$$

- Maximum ($max_{i,j}$) is the largest pressure measurement in the jth step of the ith cycle.

$$max_{i,j} = \max\left(\left|\boldsymbol{x}_{i,j}\right|\right) \tag{3.26}$$

- Crest factor ($CF_{i,j}$) is a measure of how extreme value is in the pressure measurements in the jth step of the ith cycle.

$$CF_{i,j} = \frac{max_{i,j}}{RMS_{i,j}} \tag{3.27}$$

- Mean absolute deviation ($D_{mean,i,j}$) is a measure of variability of pressure measurements in the jth step of the ith cycle by taking average of absolute deviation from the mean.

$$D_{mean,i,j} = \frac{1}{P_{i,j}} \sum_{t=1}^{P_{i,j}} \left| x_{i,j}(t) - \mu_{i,j} \right| \tag{3.28}$$

- Slope of linear regression line ($S_{LL,i,j}$) is a measure of the slope of change in pressure during the jth step of the ith cycle. $S_{LL,i,j}$ is determined through simple linear regression.

$$S_{LL,i,j} = \frac{P_{i,j} \cdot \sum_{t=1}^{P_{i,j}} \left(t \cdot x_{i,j}(t) \right) - \left(\sum_{t=1}^{P_{i,j}} t \right) \cdot \left(\sum_{t=1}^{P_{i,j}} x_{i,j}(t) \right)}{P_{i,j} \cdot \sum_{t=1}^{P_{i,j}} (t^2) - \left(\sum_{t=1}^{P_{i,j}} t \right)^2} \tag{3.29}$$

- Mean absolute error ($MAE_{i,j}$) is a measure of difference between the actual pressure measurements and the estimated pressure measurements for *the steps with the sloped pressure profiles* (e.g., equalization, provide purge, blowdown and repressurization steps) in the ith cycle ($j = 2,4,5,7,8,9,11,13,14, and\ 15$). The estimated pressure measurements in the jth step of the ith cycle, $\hat{x}_{i,j}(t)$ ($t = 1,2, \dots P_{i,j}$) are calculated by first order linear regression.

$$MAE_{i,j} = \frac{1}{P_{i,j}} \sum_{t=1}^{P_{i,j}} \left| x_{i,j}(t) - \hat{x}_{i,j}(t) \right| \ \textit{for steps with sloped pressure profile} \tag{3.30}$$

- Mean absolute errors ($MAE_{mean,i,j}\ and\ MAE_{RMS,i,j}$) measure the deviation of pressure measurements from the global mean and global RMS for *the steps with relatively flat pressure profiles* (e.g., adsorption, hold, and purge steps) in the ith cycle ($j = 1,3,6,10, and\ 12$). The global mean and global RMS of jth step are estimated based on all cycles under normal conditions (i.e., the training data), respectively.

$$\mu_{global,j} = \frac{1}{C}\sum_{i=1}^{C}\mu_{i,j} \quad (3.31)$$

$$RMS_{global,j} = \frac{1}{C}\sum_{i=1}^{C}RMS_{i,j} \quad (3.32)$$

where C is the number of cycles in the training set.

$$MAE_{mean,i,j} = \frac{1}{P_{i,j}}\sum_{t=1}^{P_{i,j}}\left|x_{i,j}(t) - \mu_{global,j}\right| \textit{ for steps with flat pressure profiles} \quad (3.33)$$

$$MAE_{RMS,i,j} = \frac{1}{P_{i,j}}\sum_{t=1}^{P_{i,j}}\left|x_{i,j}(t) - RMS_{global,j}\right| \textit{ for steps with flat pressure profiles} \quad (3.34)$$

Note that the above features except for three MAEs are also extracted from the residuals of each step which are obtained by subtracting the predicted pressure measurements from the raw pressure measurements. The linear regression is used to estimate the pressure measurements of each step. Since the residuals contain the unexplained variation, the features derived from the residual space can help distinguish the abnormal process behavior.

Feature matrix

Once all the features from a training data are calculated, I have the following training feature matrix.

$$\boldsymbol{a}_i = \left[\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i, \boldsymbol{\gamma}_i, \boldsymbol{\kappa}_i, \boldsymbol{RMS}_i, \boldsymbol{max}_i, \boldsymbol{CF}_i, \boldsymbol{D}_{mean,i}, \boldsymbol{MAE}_i, \boldsymbol{MAE}_{mean,i}, \boldsymbol{MAE}_{RMS,i}\right] \quad (3.35)$$

$$\boldsymbol{b}_i = \left[\widetilde{\boldsymbol{\mu}}_i, \widetilde{\boldsymbol{\sigma}}_i, \widetilde{\boldsymbol{\gamma}}_i, \widetilde{\boldsymbol{\kappa}}_i, \widetilde{\boldsymbol{RMS}}_i, \widetilde{\boldsymbol{max}}_i, \widetilde{\boldsymbol{CF}}_i, \widetilde{\boldsymbol{D}}_{mean,i}\right] \quad (3.36)$$

$$\boldsymbol{f}_i = [\boldsymbol{a}_i, \boldsymbol{b}_i] \quad (3.37)$$

$$\boldsymbol{M}_{TR} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_C \end{bmatrix} \quad (3.38)$$

where $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are the feature vectors from the raw pressure measurements and the residuals, respectively. $\boldsymbol{\mu}_i = \{\mu_{i,1}, \mu_{i,2}, \cdots, \mu_{i,T}\}$ is a row vector of dimension $(1 \times T)$ (i.e., in this work, $T = 15$). All types of features except for three MAEs have the same dimension. $\boldsymbol{MAE}_i$ is a row

vector of dimension (1×10), and $\boldsymbol{MAE}_{mean,i}$ and $\boldsymbol{MAE}_{RMS,i}$ have (1×5) vector dimension. $\boldsymbol{f}_i$ is the total feature vector. Let F to denote the total number of features included in each cycle so that $\boldsymbol{f}_i$ is a row vector containing F features. One of the advantages of the FSM framework is that it is flexible to have different features for different steps. In the next section, I will discuss how to select the features in each step. Another advantage of FSM framework is that it can handle unequal step duration and asynchronous cycle trajectories without any pre-processing methods such as DWT. Therefore, regardless of different cycle duration and step duration, the training feature matrix $\boldsymbol{M}_{TR}$ has a $(C \times F)$ dimension. The test feature matrix $\boldsymbol{M}_{TE}$ are calculated in the same way except that $\boldsymbol{MAE}_{mean}$ and $\boldsymbol{MAE}_{RMS}$ are computed with reference to the training data (*i.e.*, $\mu_{global,j}$ and $RMS_{global,j}$ in equations 3.31 and 3.32).

3.4.2.3 FSM-kNN algorithm and fault detection

The kNN method can be used for fault detection based on the idea that distance between a fault sample and its nearest neighboring training samples is much larger than that between a normal sample and its nearest neighboring training samples, as shown in Figure 3.8. A periodic cycle behavior can be better characterized by the features extracted from pressure measurements of each step than by the cycle pressure measurements themselves. Therefore, the proposed method FSM-kNN utilizes the features, instead of the raw process variable (*i.e.*, pressure measurements).

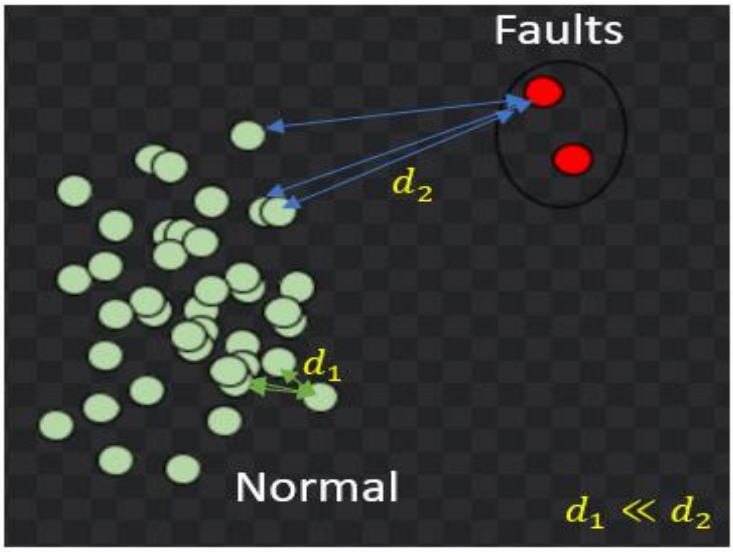


Figure 3.8 Scatter plot of normal and fault samples

Figure 3.9 shows the flow chart of the proposed FSM-kNN approach, which contains two steps –model building and fault detection.

Part I: model building

1. As discussed in the section 3.4.2.2, the features are extracted from the pressure measurements of each step and then the feature matrix is generated. Since the features have different scales, the autoscaling is performed to have zero mean and unit variance, which can reduce any bias resulting from the scale's difference of the features.
2. A training pool with a specific time constraint is constructed for ith training sample. For given α time constraint, the training pool consists of only the training samples of which cycle durations are within the range of $P_i - \alpha$ and $P_i + \alpha$, where P_i is the cycle duration of ith training sample. The time constraint helps avoid the occurrence that a fault sample in a particular cycle duration becomes normal in a very distinct cycle duration.
3. Find k-nearest neighbors for each training sample in corresponding training pool.

4. Compute the kNN squared distance for each sample in the training set by taking summation of k-nearest neighbors' distances. The kNN squared distance of ith sample (D_i^2) is calculated as follows:

$$D_i^2 = \sum_{j=1}^{k} d_{i,j}^2 \quad (3.39)$$

where $d_{i,j}^2$ is the squared Euclidean distance between ith sample and jth nearest neighbor.

5. Once the kNN squared distances for all the training samples are obtained, the threshold should be determined to monitor whether the process is under normal condition or not. The threshold is computed using a noncentral chi-square distribution based on assumption that kNN squared distances (D^2) follow a normal distribution [97]. However, if the metric does not satisfy the normal assumption, the threshold may be inaccurate, resulting in increase of false alarm rate and/or missing fault rate. In this work, kernel density estimation (KDE) is used to estimate the distribution of kNN squared distances (D^2). The univariate kernel estimator is defined as

$$\hat{f}_h(D^2) = \frac{1}{C}\sum_{i=1}^{C} K\left(\frac{D^2 - D_i^2}{h}\right) \quad (3.40)$$

where $\hat{f}_h(D^2)$ is the estimated probability density of D^2, K and h are a kernel function and the bandwidth of kernel function, respectively. In this work, the Gaussian kernel is used. C is the number of training cycles. More details about KDE can be found in [119]. The 99% confidence limit D_α^2 is determined as follows:

$$\int_{-\infty}^{D_\alpha^2} \hat{f}_h(D^2)\, dD^2 = 1 - \alpha, \text{ where the significance level } \alpha = 0.01 \quad (3.41)$$

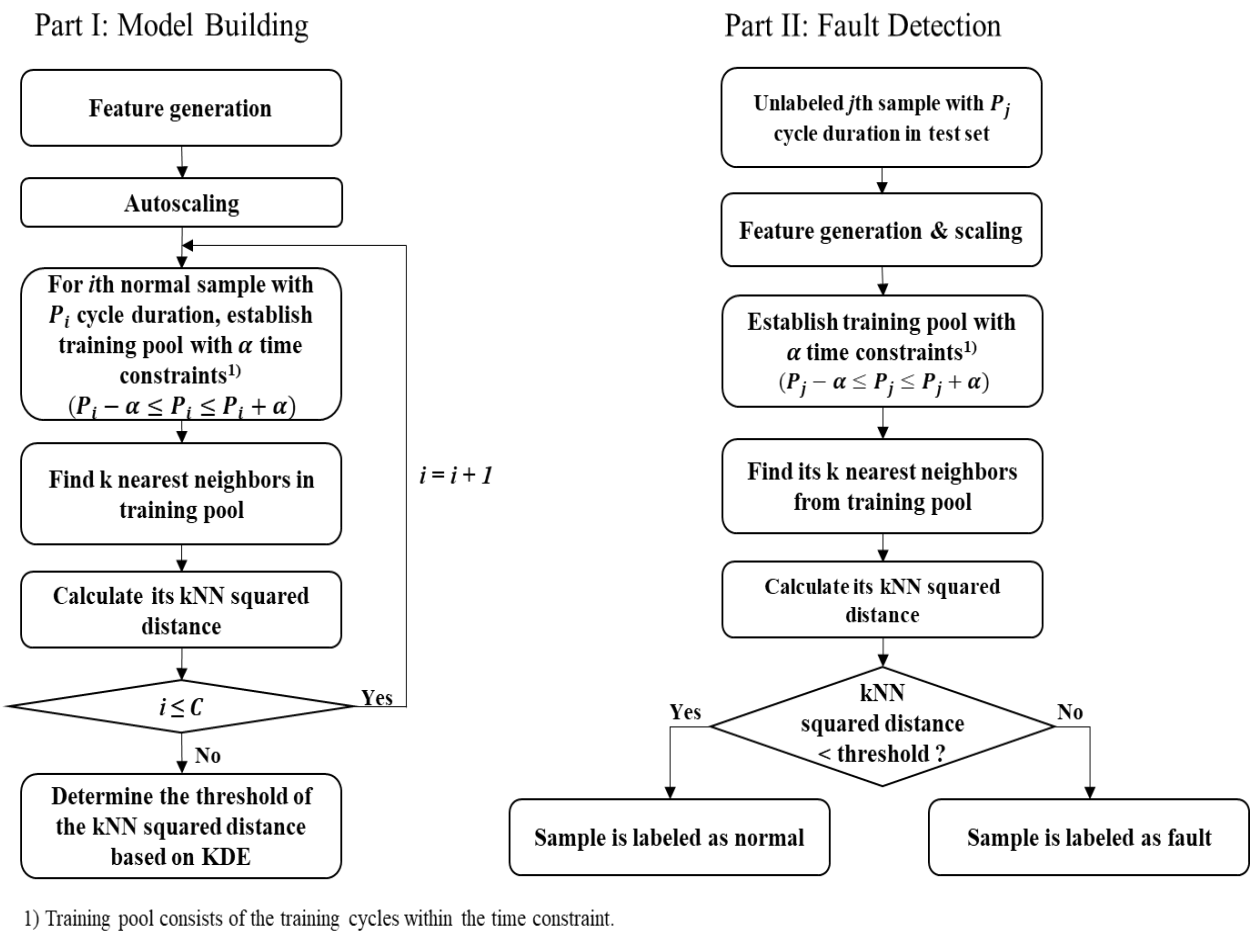


Figure 3.9 Flow chart of the proposed FSM-kNN approach. (a) model building; (b) fault detection.

Part II: fault detection

1. The features are extracted for *j*th test sample in the test set. Then, in order to eliminate the scale's difference of the features, the scaling is performed based on same mean and standard deviation as autoscaling used.
2. A training pool with α time constraint is constructed for *j*th test sample.
3. Find k-nearest neighbors for *j*th test sample in corresponding training pool.
4. Calculate the kNN squared distance D_j^2 of *j*th test sample using equation 3.39.

5. Compare the threshold D_α^2 with D_j^2 to determine if the process is in normal condition or not. If $D_j^2 \leq D_\alpha^2$, it is identified as a normal one. Otherwise, it is classified as an abnormal one.

3.4.2.4 Step-wise fault diagnosis

Once a fault is detected, fault diagnosis is performed to identify the root cause of fault condition. The contribution plot is well known diagnostic tools. In this work, I propose a step-wise kNN-based fault diagnosis method. The proposed fault diagnosis is the contribution-based approach, where a step with the most significant contribution is regarded as the root cause of the fault. In other words, the contribution plot on the steps indicates the effect of each step on the kNN squared distance. The kNN squared distance of ith sample (D_i^2) can be decomposed by step:

$$D_i^2 = \sum_{j=1}^{k} d_{i,j}^2 = \sum_{s=1}^{T} d_{i,s}^2 \quad (3.42)$$

where $d_{i,j}^2$ is the squared Euclidean distance between ith sample and jth nearest neighbor that is calculated based on the features from all the steps. $d_{i,s}^2$ is the summation of all k nearest neighbors' squared Euclidean distances at sth step (*i.e.*, $d_{i,s}^2 = \sum_{j=1}^{k} d_{i,j,s}^2$). Figure 3.10 show the decomposition of D_i^2 into the step-wise squared Euclidean distances ($d_{i,s}^2$). The contribution of $d_{i,s}^2$ on D_i^2 is defined as follows:

$$C_{i,s} = \frac{d_{i,s}^2}{D_i^2} \times 100\% \quad (3.43)$$

where $\sum_{s=1}^{T} C_{i,s} = 100$. The step(s) with the largest contribution(s) (*i.e.*, $C_{i,s}$) are considered as the root cause of the fault.

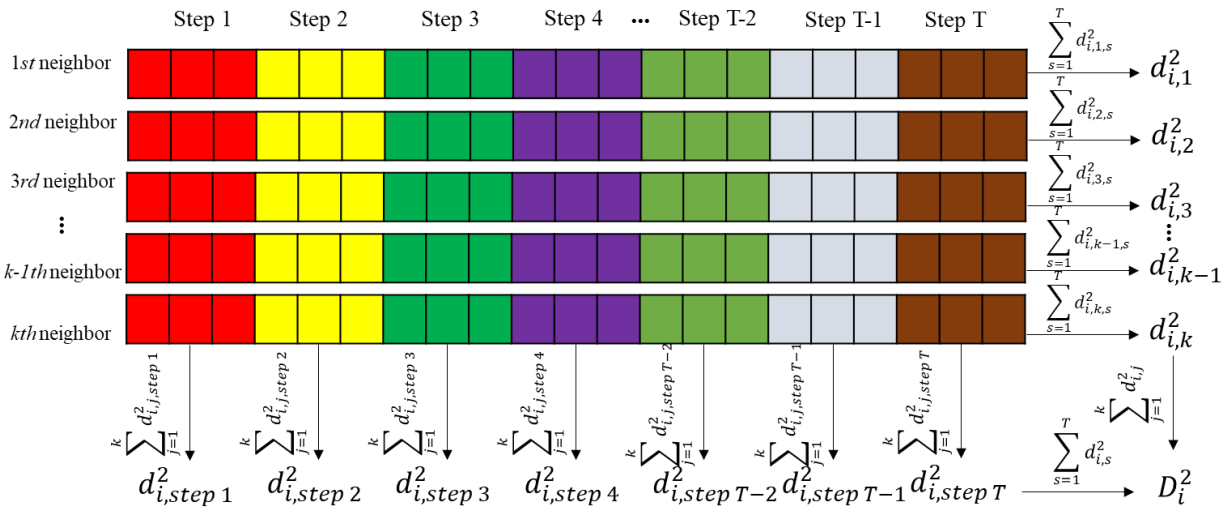


Figure 3.10 the decomposition of D_i^2 into the step-wise squared Euclidean distances ($d_{i,s}^2$). The different colors denote the different steps and the boxes in each step represent the features corresponding to the step.

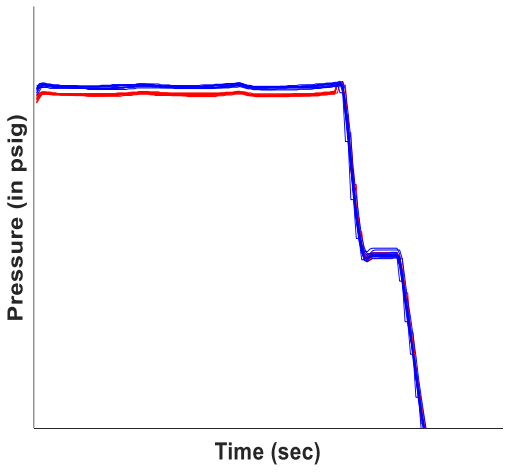
3.5 Industrial case studies

In this section, I demonstrate the effectiveness of fault detection and diagnosis of proposed FSM-kNN method compared to the conventional MPCA and two kNN based fault detection methods, FD-kNN and SkNN, using four simulated faults and one real fault from an industrial PSA process. Figure 3.11 and Table 3.1 describe the five fault scenarios. The first four scenarios are simulated faults while the last scenario is a real fault in actual PSA operation. For the simulated faults, similar behaviors have been observed in actual operations, but the historical data for those types of faults are no longer available. In these cases, the faults were introduced by manipulating the real industrial data under normal operations.

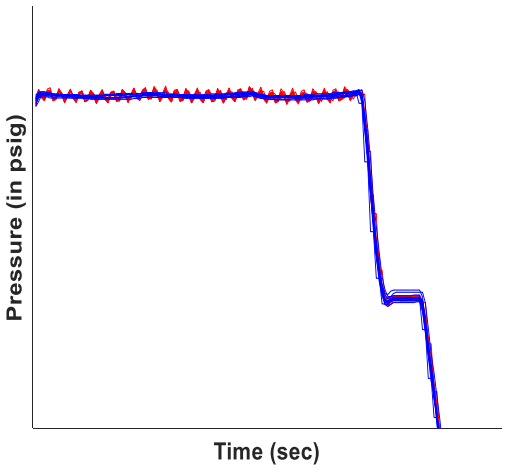
Table 3.1 Description of fault scenarios

Fault scenarios #	Description

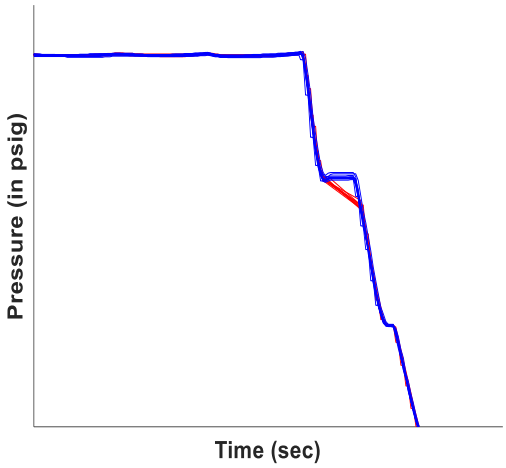
1	During adsorption step, the normal cycles have almost constant pressure without significant change in pressure. The faulty cycles have lower pressure than normal cycles.
2	During adsorption step, the faulty cycles have higher pressure variations than normal cycles.
3	During a hold step, the normal cycles keep constant pressure. The pressure of the faulty cycles decreases instead of being held steady.
4	During an equalization step, the pressure of the normal cycles linearly decreases. The pressure of the faulty cycles was held steady followed by a sudden drop instead of smooth decrease, which significantly deviate from the straight pressure profile.
5	During re-pressurization, the pressure of the normal cycles linearly increases. The pressure of the faulty cycles does not follow the normal cycle trajectory. The faulty cycles have the non-linear pressure profile.



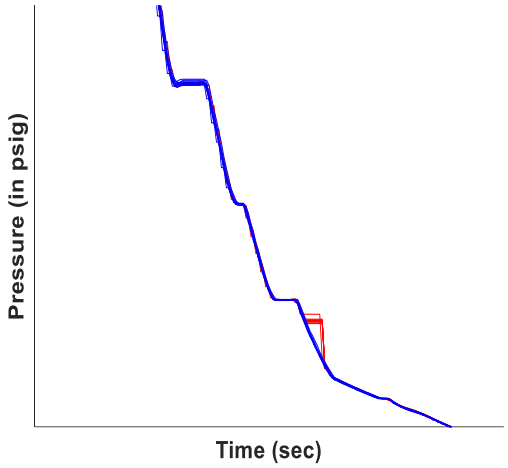
(a)



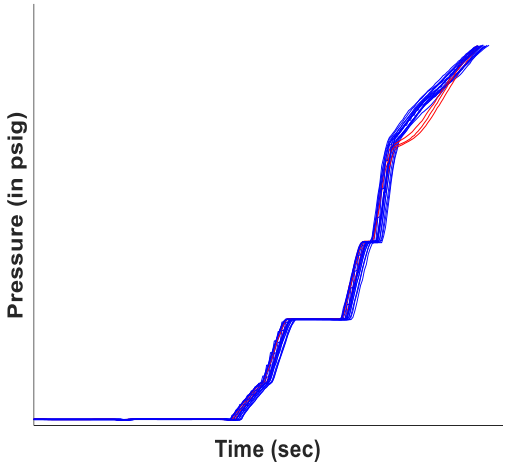
(b)



(c)



(d)



(e)

Figure 3.11 Plot of fault scenarios. (a) fault scenario 1, (b) fault scenario 2, (c) fault scenario 3, (d) fault scenario 4, (e) fault scenario 5. Blue and red lines denote normal and fault cycles, respectively.

For MPCA, FD-kNN and SkNN methods, DTW is employed to synchronize the pressure profile of each step, since they require that all cycles should have the same step duration, Therefor, for these methods, the number of variables after unfolding is 619. However, the proposed FSM-kNN uses only 69 features. In this work, I propose guidance on how to select the

features. Initially, the steps of the PSA process shown in Figure 3.6 can be grouped into four classes. There are three criteria to classify the steps:

1. Shape of the pressure profile of the steps
2. Sensitivity of pressure of each step to cycle duration
3. Length of the steps

Figure 3.12 shows the diagram of step classification. First, I classify the steps into steps with flat pressure profile and steps with sloped pressure profile based on the shape of the pressure profile of the steps. Second, the steps with flat pressure profile can be further divided into two groups, adsorption step, and hold 1/2/3 and purge steps based on the sensitivity of pressure of each step to cycle duration. Regardless of cycle duration, the adsorption step has consistent pressure measurements, compared to hold and purge steps. Lastly, for steps with sloped pressure profile, I can further classify the steps based on the length of the steps. Since the step lengths can give different impact on features, I divide them into short and long length steps. It is worth noting that each class can contain different features because the process characteristics of the class are different and can be better captured by the different subset of features.

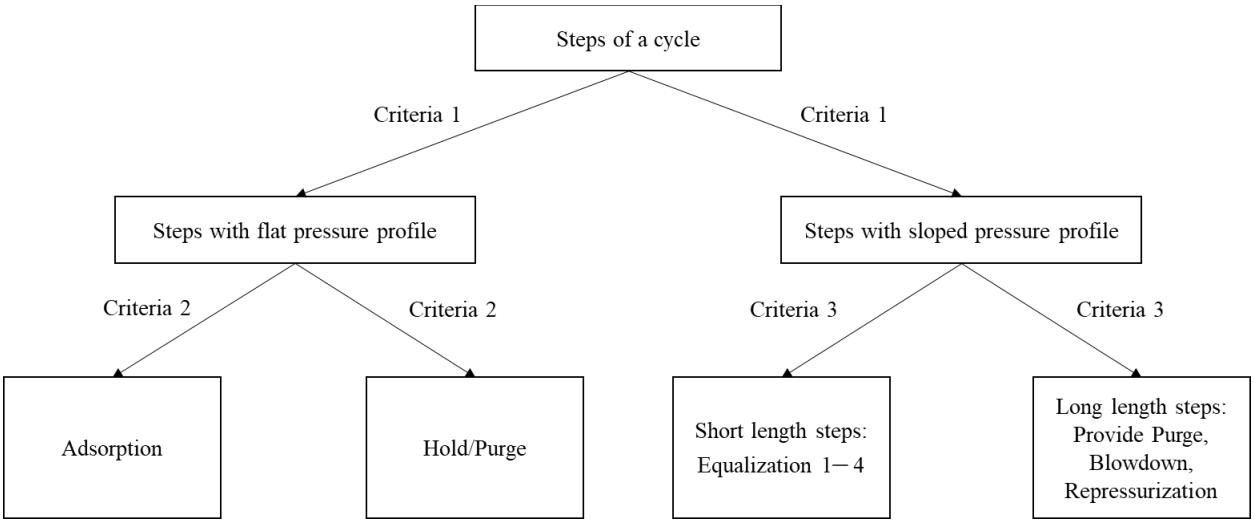


Figure 3.12 Diagram of step classification

As I discussed in section 3.4.2.2, each feature can explain the different aspect of process conditions. Therefore, I can select the appropriate features based on understanding on characteristics of both abnormal process disturbance and normal condition so that I can reduce redundant features as much as possible and build a parsimony model. For adsorption step, pressure measurements are consistent regardless of cycle duration. If there is any pressure shift in the step, the features related to central tendency can be significantly influenced by the disturbance. For abnormal pressure fluctuation, the features measuring the dispersion of pressure distribution are well suitable to detect the anomaly. In addition, the features calculated from the residual space help detect the abnormality because the residuals contain more unexplained variation. For pressure profile distortion, we need features measuring degree of shape/distribution distortion such as skewness, slope of linear regression line and so on. Like the pressure fluctuation, the features extracted from the residual space can capture the pressure distortion. Table 3.2 shows the selected features for each class.

Table 3.2 The selected features for each class.

	Steps with flat pressure profile		**Steps with sloped pressure profile**	
	Adsorption	Hold/Purge	Short length steps	Long length steps
Features	$\mu, \sigma, \widetilde{D}_{mean}$, MAE_{mean}, MAE_{RMS}	$\mu, \sigma, \widetilde{D}_{mean}$, MAE_{mean}, S_{LL}	μ, σ, γ	$\mu, \sigma, \gamma, D_{mean}$, $CF, MAE, \widetilde{D}_{mean}$, $\widetilde{RMS}, \widetilde{max}$

In this work, among total 6007 cycles under normal condition, the approximately 80% and 20% of the cycles are used as the training set and the validation set, respectively. For the fault scenarios 1 – 4, total 20 cycles are used as the test set and among which 10 are normal cycles and the others are faulty cycles. For the fault scenario 5 (*i.e.*, real fault), total 19 cycles are

utilized as the test set and among which 16 cycles are normal and 3 cycles are fault. For all methods, the tuning parameters are determined through the validation. In addition, the control limits on Hoteling's T^2 and SPE for MPCA and on kNN squared distance for kNN based methods are determined using KDE at confidence level 99%. All information about the datasets and the tuning parameters is summarized in Table 3.3.

Table 3.3 The description of the datasets and the methods

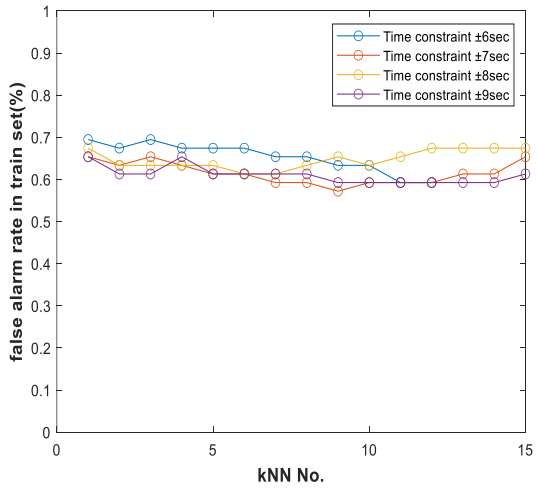
	MPCA	**FD-kNN**	**SkNN**	**FSM-kNN**
# of variables/features		619		69
# of PC's or kNN	14 PCs	3 NNs	11 NNs	9 NNs
Training / Validation		4895 / 1112 normal cycles		
Testing		20 cycles (10 normal, 10 fault)[1)] 19cycles (16 normal, 3 fault)[2)]		
Confidence level		99%		

1) Test set for fault scenarios 1 – 4, 2) Test set for fault scenario 5

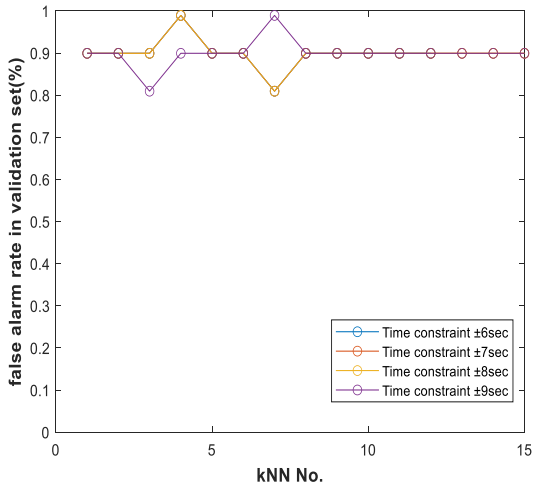
3.5 Results

The proposed FSM-kNN method has two tuning parameters – the number of nearest neighbors and time constraint for training pool. In this work, these tuning parameters are optimized based on two metrics: (1) false alarm rates in the training set and validation set; (2) Kullback-Leiber divergence (KLD). KLD is used to measure the similarity between the distributions of kNN squared distances of the training set and the validation set. If KLD is 0, two distributions of kNN squared distances are equal. The lower the KLD value, the more similar the distributions of kNN squared distances. Also, the lower false alarm rates in the training set and validation set, the better the fault detection method. Therefore, the optimal tuning parameters

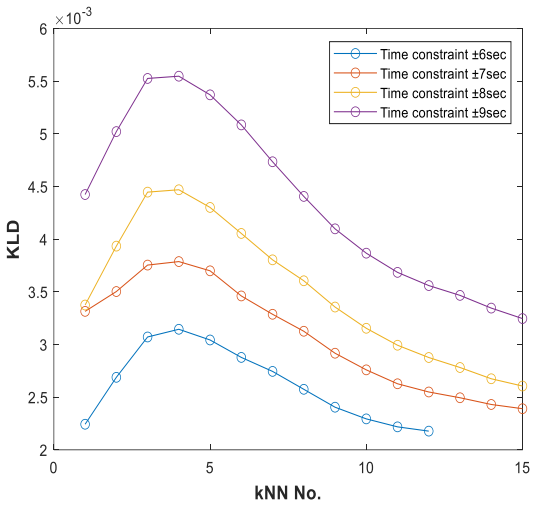
enable the model to have a small KLD value and low false alarm rates. The effect of tuning parameters on two metrics are shown in Figure 3.13 (a) – (c). Figure 3.13 (a) and (b) show the false alarm rates in the training set and validation set against the number of NNs and time constraints, respectively. The four lines denote the different time constraints for training pool. I can see only two lines in Figure 3.13 (b) because the false alarm rates for time constraint 6 – 8 seconds are exactly equivalent. This observation demonstrates the choice of time constraint is uncritical. However, in case that the time constraint is too large, a fault sample in a particular cycle duration can be normal in a very distinct cycle duration, which can deteriorate performance of the fault detection method. Therefore, in this work, I do not study the large time constraints.



(a)



(b)



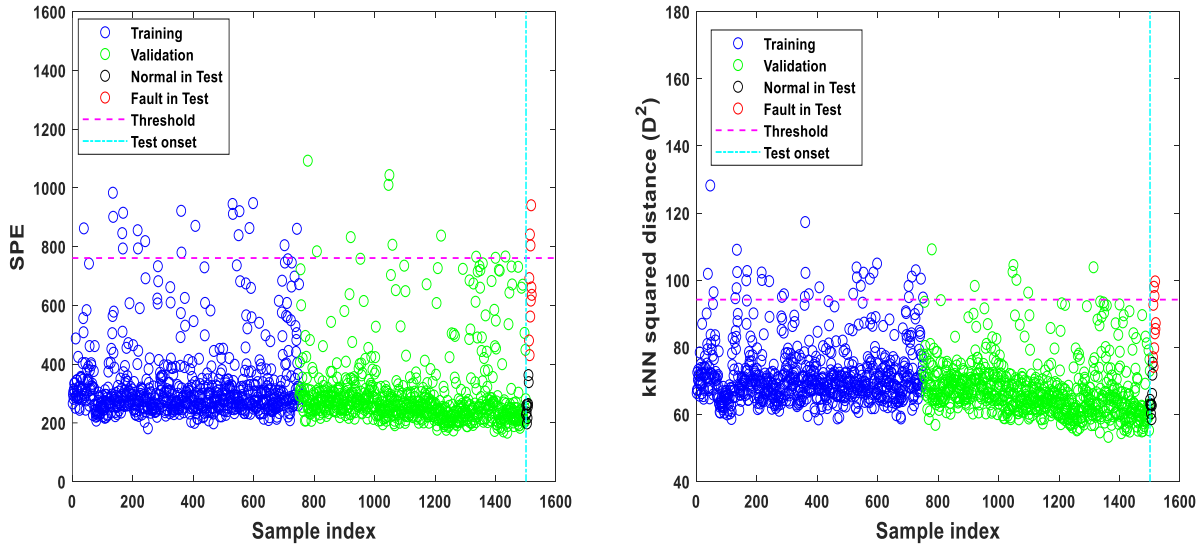
(c)

Figure 3.13 Plot of two metrics against tuning parameters. (a) false alarm rate in training set (%) against kNN and time constraints, (b) false alarm rate in validation set (%) against kNN and time constraint, (c) KLD against kNN and time constraint.

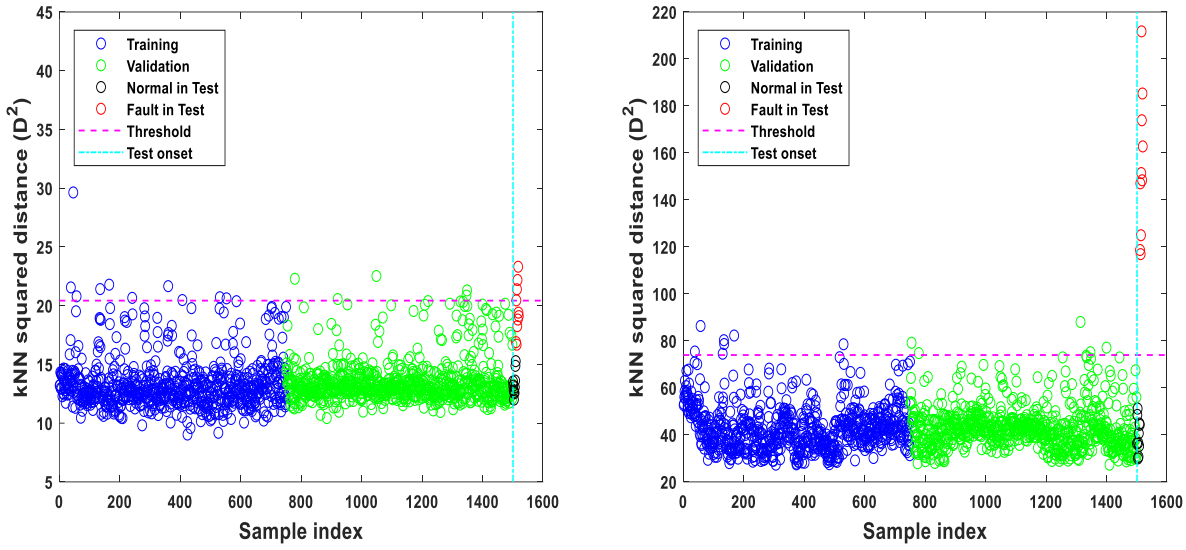
In addition, the number of NNs do not significantly influence the false alarm rates in training set and validation set. This fact is consistent with the result that the performance of the model is not sensitive to the selection of kNN [97]. Note that like the time constraint, if the number of NNs is too large, the fault detection method has difficulty in detecting faults because the boundaries between normal and fault cycles are less conspicuous. As shown in Figure 3.13 (c), with increase of time constraints, KLD increases and as the number of NNs exceeds a certain value, KLD decreases. Therefore, I determine 9-NNs and ±7 seconds time constraint as the optimal one due to low false alarm rates and low KLD. However, it is worth noting that FSM-kNN method provides a robust solution over a wide range of tuning parameters.

The fault detection and diagnosis results of fault scenarios 2, 3 and 5 are shown in Figures 3.14 – 3.19. In Figures 3.14 – 3.16, only the initial 750 sample points from the respective training set and validation set are shown for simplicity. Figure 3.14 illustrates all the fault

detection methods except for the proposed FSM-kNN have difficulty in detecting the faulty cycles. They can only identify three or four faults correctly. However, FSM-kNN detects all ten faulty cycles without any false alarms. For fault scenario 2, all the methods correctly identify the root cause of the faulty step as the adsorption step as shown in Figure 3.17. Figure 3.15 shows that all the faulty cycles are successfully classified as the faults by all the methods. However, only FSM-kNN can identify the faulty step as the equalization 4 step, but the other methods mis-identify the root cause of the faulty step as shown in Figure 3.18. The fault scenario 5 demonstrates the capability of the proposed FSM-kNN to capture the abnormal disturbance of PSA process condition. Figure 3.16 illustrates that the conventional fault detection methods fail to detect any faulty cycles, which lead to misleading identification of the root cause of the fault. Again, only FSM-kNN successfully detects all faulty cycles. Besides, FSM-kNN recognizes that the abnormal condition results from the disturbance in the repressurization step as shown in Figure 3.19.

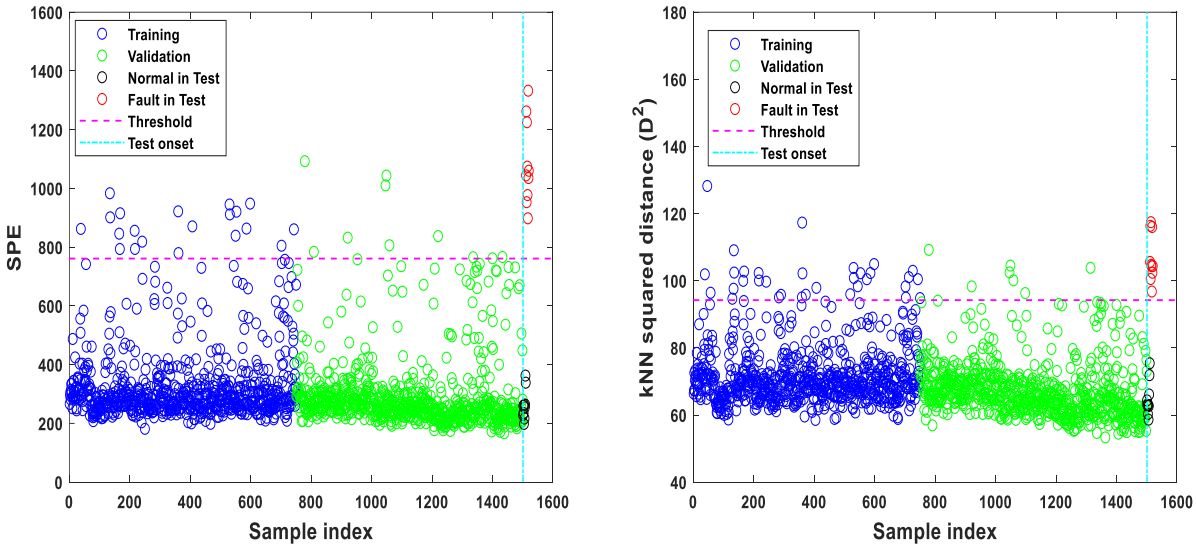


(a) (b)

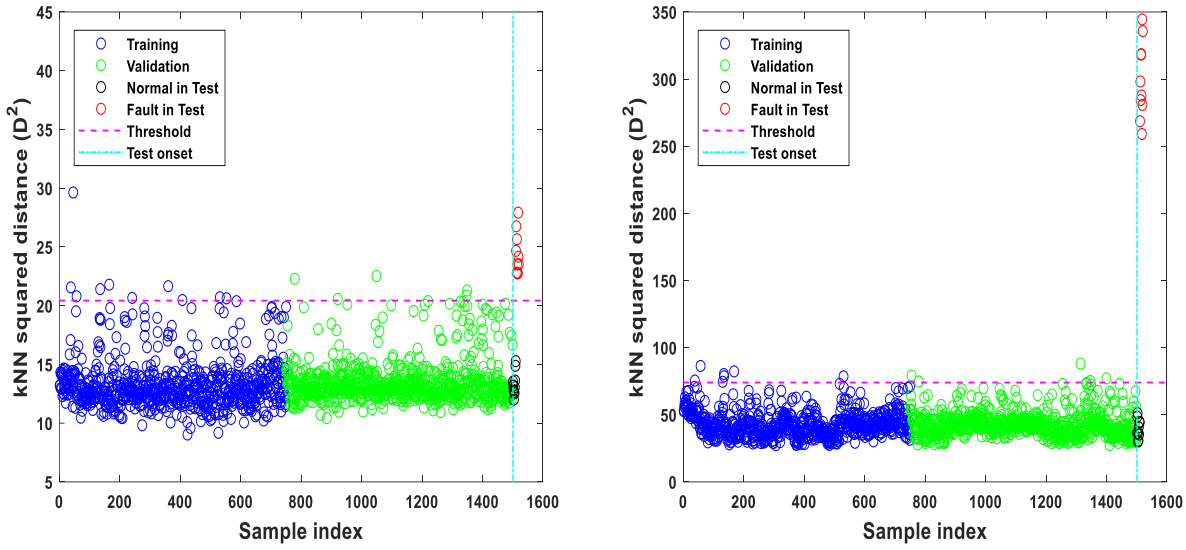


(c) (d)

Figure 3.14 Fault detection for fault scenario 2: (a) MPCA, (b) FD-kNN, (c) SkNN, (d) FSM-kNN.

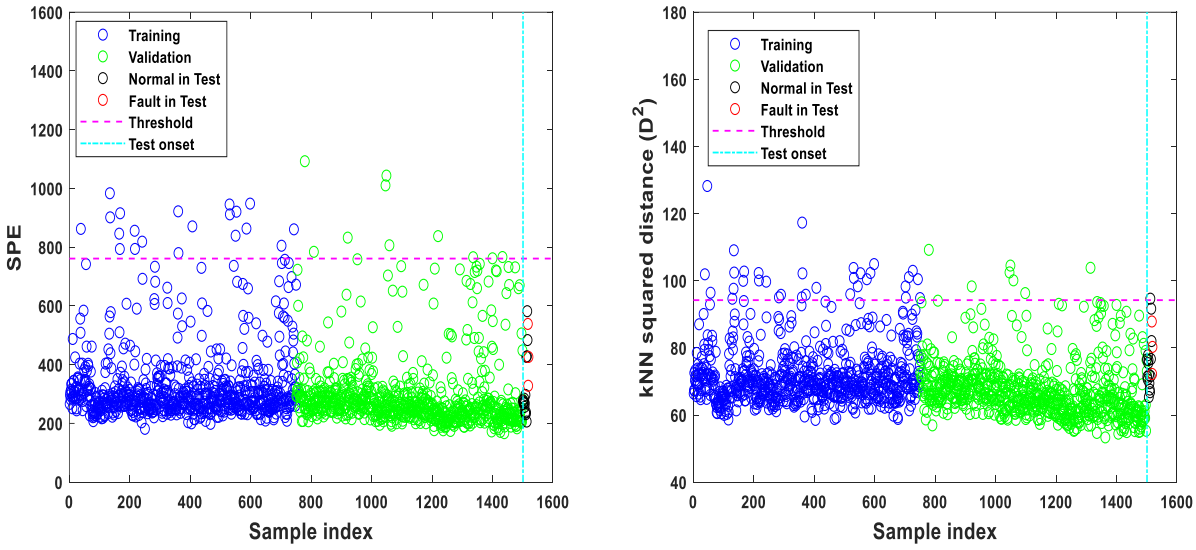


(a) (b)

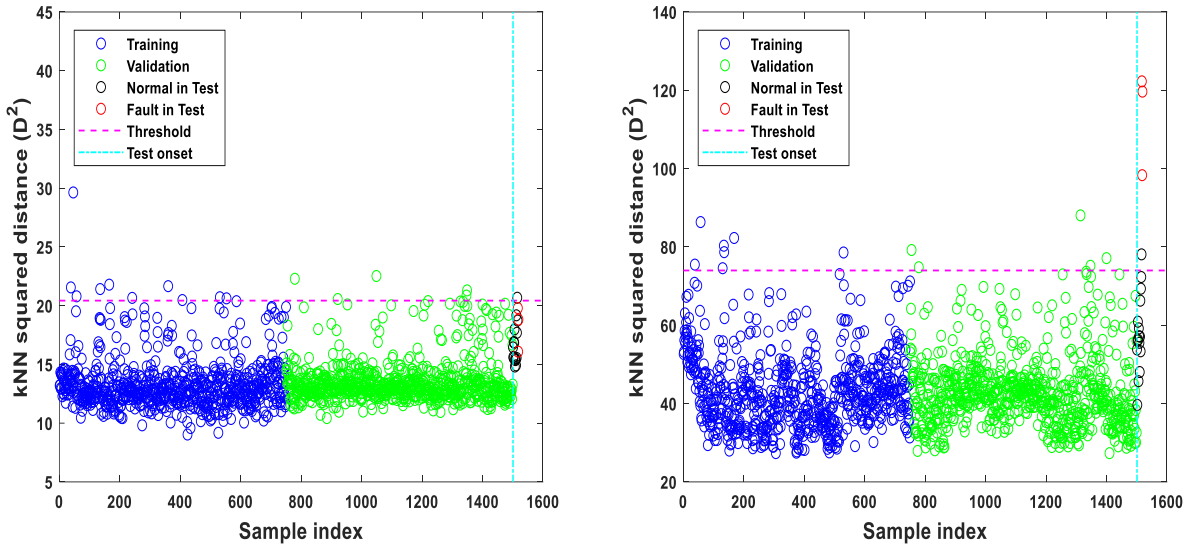


(c) (d)

Figure 3.15 Fault detection for fault scenario 4: (a) MPCA, (b) FD-kNN, (c) SkNN, (d) FSM-kNN.

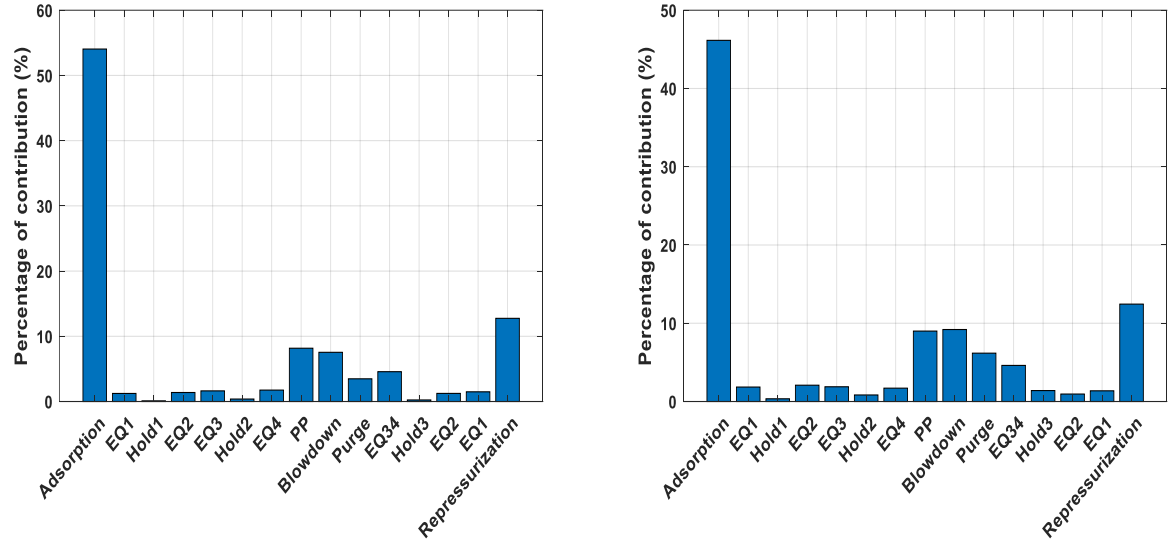


(a) (b)

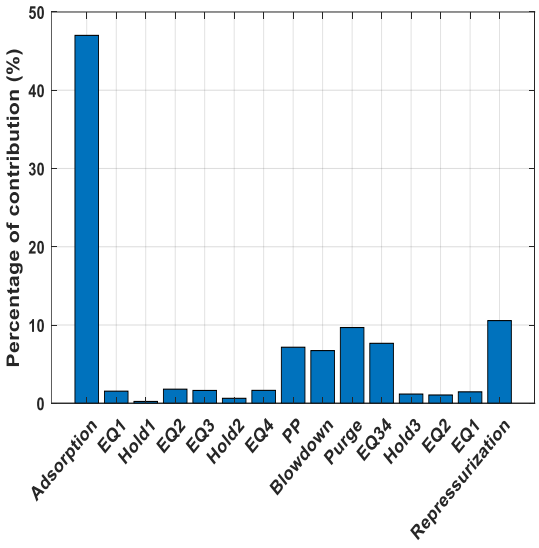


(c) (d)

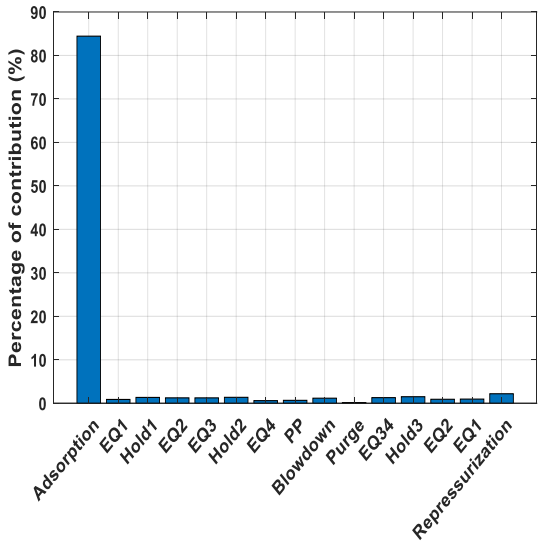
Figure 3.16 Fault detection for fault scenario 5: (a) MPCA, (b) FD-kNN, (c) SkNN, (d) FSM-kNN.



(a) (b)

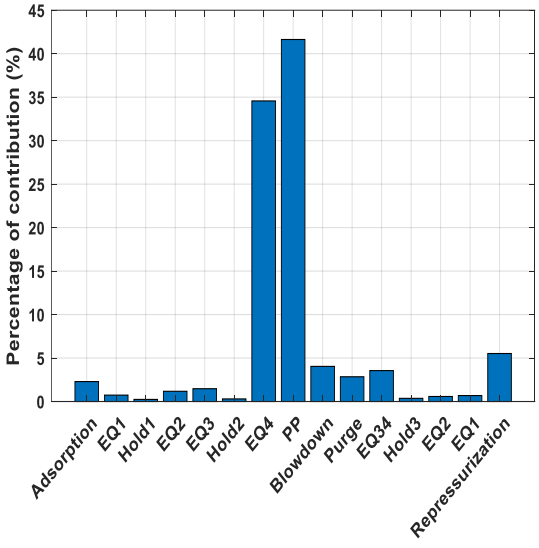


(c)

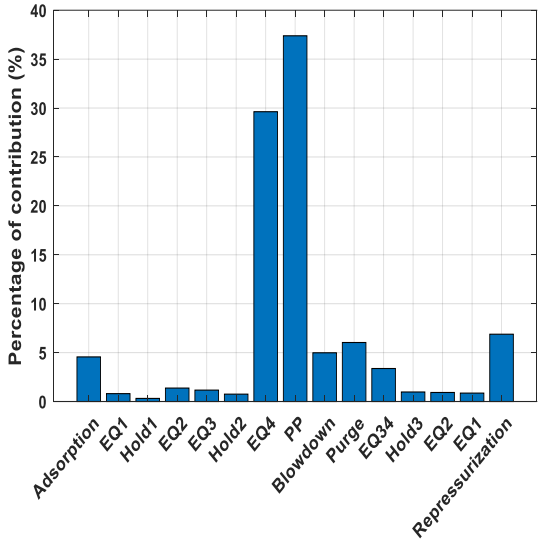


(d)

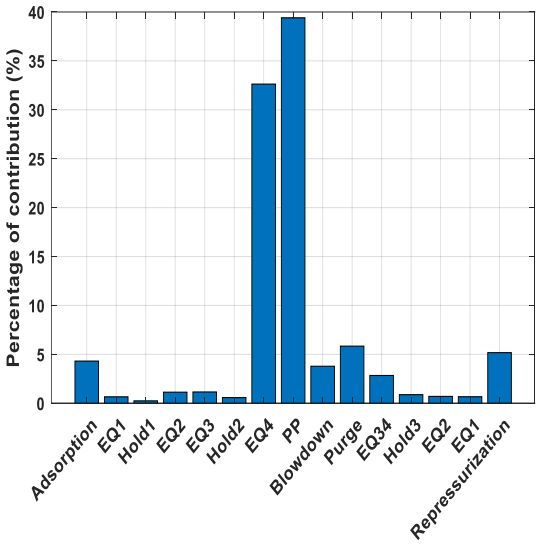
Figure 3.17 Fault diagnosis for fault scenario 2: (a) MPCA, (b) FD-kNN, (c) SkNN, (d) FSM-kNN.



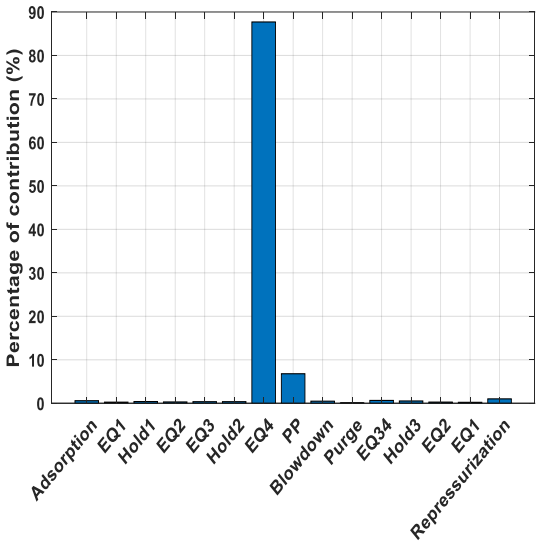
(a)



(b)

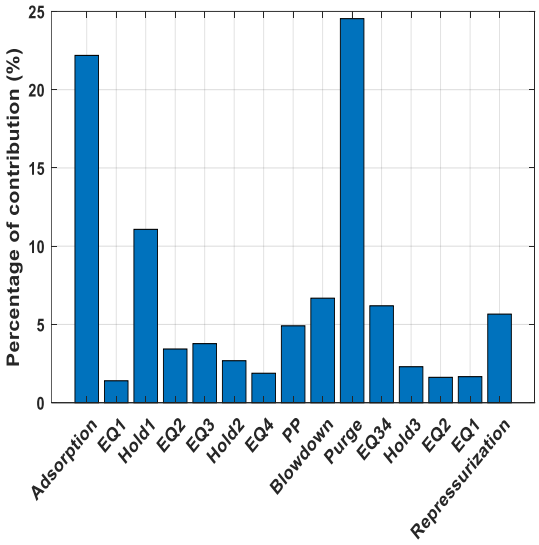


(c)

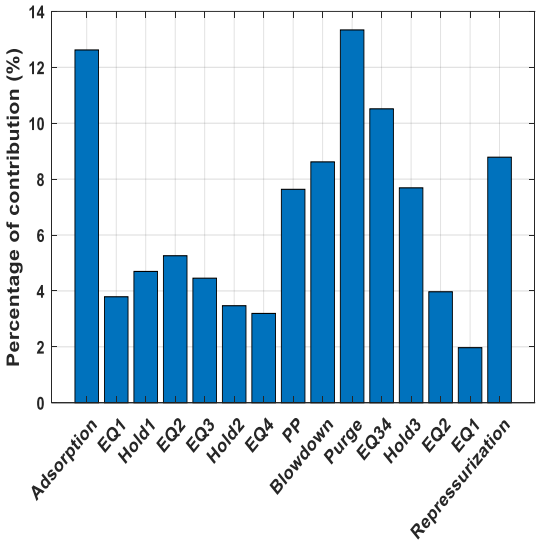


(d)

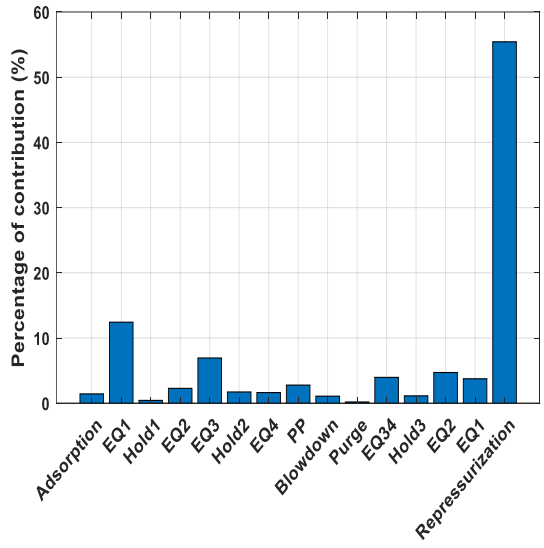
Figure 3.18 Fault diagnosis for fault scenario 4: (a) MPCA, (b) FD-kNN, (c) SkNN, (d) FSM-kNN.



(a)



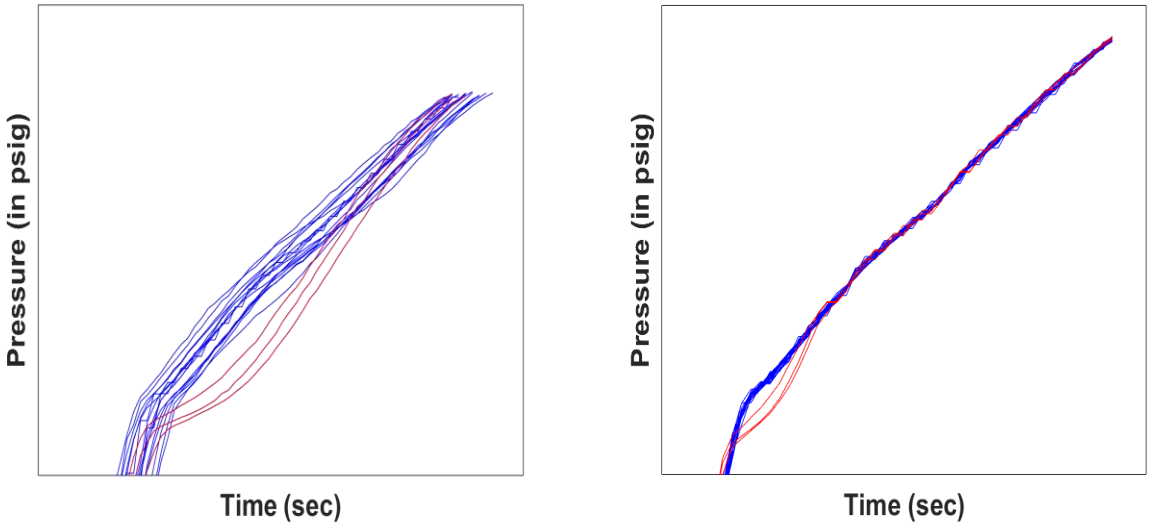
(b)



(c)

Figure 3.19 Fault diagnosis for fault scenario 5: (a) FD-kNN, (b) SkNN, (c) FSM-kNN.

Interestingly, all conventional fault detection methods provide the unsatisfactory fault detection results for fault scenario 5. Further investigation is conducted to understand the reason for the failure of the fault detection methods. Since the conventional methods employ DTW to synchronize the cycle trajectories, I suspect that the failure is related to data pre-processing by DTW. Figures 3.20 (a) and (b) show the pressure profiles in repressurization step before DTW and after DTW, respectively. These plots illustrate that the trajectories of faulty cycles in the original pressure profiles are distorted after DTW. This observation demonstrates that DTW can introduce bias, resulting in missed detection of faulty cycles.



(a) (b)

Figure 3.20 Plots of the pressure profiles in repressurization step: (a) before DTW, (b) after DTW. Blue and red lines denote the normal and faulty cycles, respectively.

To summarize overall results for fault scenarios 1 – 5, the fault detection rates and diagnosis are listed in Table 3.4, and false alarm rates in training set and test set are shown in Table 3.5. In Table 3.4, the parenthesis indicates the root cause of faults that respective methods identify. Only the proposed FSM-kNN can detect all faulty cycles for fault scenarios 1 – 5 and identify the root cause of faults correctly. In addition, FMS-kNN has lower false alarm rates in training set and test set, compared to other fault detection methods. This satisfactory results stem from the following advantages of FSM-kNN method. First, the proposed method utilizes the features, instead of the process variable. The features can better capture process characteristics such as nonlinearity and non-Gaussianity than the raw process variables. Besides, the feature space can significantly reduce the dimension of input space, which may reduce risk of overfitting. Second, FSM-kNN does not require any data-preprocessing to synchronize the cycle trajectories. It can handle unequal step duration and asynchronous cycle trajectories through feature generation. Therefore, it can avoid any information loss or distortion caused by data pre-

processing methods. Lastly, FSM-kNN is well suitable for multimode processes. This is because kNN models can be built locally. Therefore, it does not require complex algorithms for multimode processes.

Table 3.4 Fault detection rate and diagnosis

Fault #	MPCA (T^2)	MPCA (SPE)	FD-kNN	SkNN	FSM-kNN
1	100% (Adsorption)	90% (Adsorption)	100% (Adsorption)	100% (Adsorption)	100% (Adsorption)
2	0% (-)	30% (Adsorption)	30% (Adsorption)	40% (Adsorption)	100% (Adsorption)
3	0% (-)	0% (-)	0% (-)	0% (-)	100% (Hold 1)
4	0% (-)	100% (PP)	100% (PP)	100% (PP)	100% (EQ4)
5	0% (-)	0% (-)	0% (-)	0% (-)	100% (Repressurization)

Table 3.5 False alarm rates in training set and test set

Fault #	MPCA (T^2) (training/test)	MPCA (SPE) (training/test)	FD-kNN (training/test)	SkNN (training/test)	FSM-kNN (training/test)
1	1.65% / 0%	1.16% / 0%	1.33% / 0%	0.98% / 0%	0.57% / 0%
2	1.65% / 0%	1.16% / 0%	1.33% / 0%	0.98% / 0%	0.57% / 0%
3	1.65% / 0%	1.16% / 0%	1.33% / 0%	0.98% / 0%	0.57% / 0%
4	1.65% / 0%	1.16% / 0%	1.33% / 0%	0.98% / 0%	0.57% / 0%
5	1.65% / 0%	1.16% / 0%	1.33% / 6.25%	0.98% / 6.25%	0.57% / 6.25%

3.6 Summary and conclusions

Periodic operations, such as pressure swing adsorption (PSA) and simulated moving bed (SMB), have gained wider applications in industries. This is because they can produce high quality product with low energy and low capital cost. Despite extensive industrial applications of PSA processes, process monitoring of these operations has received limited attention compared to continuous or batch processes. One potential reason is that PSA processes have three distinct characteristics from continuous or batch processes, which pose the challenges of the process monitoring of these operations; (1) periodic processes are operated at unsteady-state, (2) processes have very complex nonlinear behaviors, and (3) processes have multimodal operations due to a change of cycle time. To address these challenges, I propose the k-nearest neighbor-based feature space monitoring (FSM-kNN) method to monitor the PSA processes. First, I employ FSM framework where statistics and morphological features are extracted from each step of a cycle since these statistics/features can better capture process characteristics than the original process variables. Therefore, these features are used for effective fault detection of PSA processes instead of the raw process variables. In addition, FSM can handle asynchronous cycle trajectories that stem from unequal step and/or cycle duration through the generation of the statistics/features. Second, I use the kNN method for fault detection. It utilizes the attribute that distance between a faulty cycle and its nearest neighboring training cycles is much larger than that between a normal cycle and its neighboring training cycles. I demonstrate the effectiveness of fault detection and diagnosis performance of FSM-kNN compared to three conventional fault detection methods using five PSA fault scenarios. It is worth noting that FSM-kNN is only method that can detect all the faulty cycles in each fault scenario. Besides, FSM-kNN correctly identifies the root cause of faults for all the fault scenarios. In comparison, the conventional fault

detection methods fail to detect all the faults and to identify the root cause of faults in some fault scenarios. The success of the proposed method results from the following factors: (1) FSM-kNN method is well suited for multimodal datasets as the models are built locally; (2) FSM-kNN method makes no assumption about the linearity of a dataset. When a dataset has strong nonlinearity, the method can perform efficient process monitoring while linear MSPM methods such as PCA could have high false alarm rates and/or miss faults; (3) FSM-kNN method is simple and practical. It does not require any complex preprocessing methods, which may distort process trajectories and introduce bias. In this work, I apply the proposed method to the PSA process. However, FSM-kNN can be utilized to monitor other periodic processes as well as batch and/or continuous processes.

Chapter 4. Knowledge-guided path analysis for understanding the effect of specialization on hospital performance

4.1 Background

The healthcare spending in the United States accounts for about 17% of US GDP, making the US the highest healthcare expenditure per capita in the world. However, the United States has not seen an increase in life expectancy to match its huge outlay on healthcare. Therefore, there are areas for improvement in the healthcare system and hospital care is one of them. This is because hospital care represents the single largest national health expenditure by the type of services, accounting for approximately 31% of total healthcare costs [121]. One of many potential routes is through healthcare specialization based on the so-called focused factory theory originated from manufacturing, which states that factories that concentrate on narrow range of services or operations produce better products at low costs. There have been many debates over the subject. Consequently, several studies have been conducted to investigate the effect of specialization on the hospital cost and patient outcome, as well as the effects of specialization on other measures of hospital performance such as efficiencies quantified by data envelopment analysis (DEA) [122]. In general, some researchers believe that hospital specialization not only leads to optimal allocation of resources to right places, but also reduces needless waste of materials and processing time [123], [124]. In addition, some argue that physicians in specialized hospitals are likely to have more opportunities to improve their expertise in treating specific diseases, resulting in more effective treatment – better outcome at lower cost [125]. However, most of these studies were based on few individual cases, or a limited number of hospitals [123],

---

This chapter was excerpted from " Understanding the effect of specialization on hospital performance through knowledge-guided machine learning " published in Computers & Chemical Engineering [120]. The author is the first author of these papers.

[125], [126]. There were studies that have investigated factors associated with high-quality/low-cost hospital performance using national databases such as the healthcare cost and utilization project (HCUP). However, specialization was not considered in those studies [127].

Focused factory theory or specialization has been extensively studied for process industry [128]. For example, it has been modeled using Bayesian framework [129], and it has been investigated for its role in decision making for planning, scheduling and dispatching tasks [130]. Hospital operations and services share many similarities with chemical processing or manufacturing processes in general. For example, they all consist of multiple units or unit operations. These units, which are not isolated but rather highly connected, interact with each other to form a dynamic system that determines the cost and patient outcome in the case of hospital operations, and cost/yield and quality of product in the case of manufacturing. Therefore, I believe that some process systems engineering (PSE) principles and techniques, such as machine learning regression and discriminant analysis that are developed to model manufacturing processes, can be adapted to model hospital operations. In this work, I examine whether the focused factory theory is applicable to hospital operations. Specifically, using a large national healthcare cost and utilization project (HCUP) dataset, I examine whether the hospitals that are specialized in certain diseases achieve better performance in terms of cost and patient outcome, measured by total charge (TOTCHG) and death of patient (DIED) during hospitalization, respectively. Pure data-driven machine learning (ML) approaches, and knowledge-guided ML approach (*i.e.*, path analysis) are used to investigate the effect of hospital specialization on hospital performance (*i.e.,* TOTCHG and DIED during hospitalization).

4.2 Introduction to Health Cost and Utilization Project (HCUP) dataset

The Healthcare Cost and Utilization Project (HCUP) is a family of healthcare databases and related software tools and products developed through a Federal-State-Industry partnership and sponsored by the Agency for Healthcare Research and Quality (AHRQ). HCUP databases bring together the data collection efforts of state data organizations, hospital associations, private data organizations, and the federal government to create a national information resource of encounter-level healthcare data (HCUP Partners). HCUP includes the largest collection of longitudinal hospital care data in the United States, with all-payer, encounter-level information beginning in 1988. These databases enable research on a broad range of health policy issues, including cost and quality of health services, medical practice patterns, access to healthcare programs, and outcomes of treatments at the national, state, and local market levels [131]. The National Inpatient Sample (NIS) used in this study is part of HCUP, which covers all patients, including individuals covered by Medicare, Medicaid, or private insurance, as well as those who are uninsured. Overall, NIS covers more than 95 percent of the U.S. population and includes more than 94 percent of discharges from U.S. community hospitals. The 2102 NIS dataset used in this study includes 7,296,968 cases (*i.e.*, inpatient stays). Each case has 481 variables, which include de-identified (*i.e.*, all personally identifiable information has been removed to protect individual identities and privacy) patient data such as age, gender, ethnicity, etc.; disease, diagnosis and procedure data such as disease severity, length of stay (LOS), diagnosis related group (DRG), cost, total charge (TOTCHG), whether the patient died during hospitalization (DIED), etc. In addition, each case is also linked to hospital related data/information such as bed-size, location, ownership, and teaching status of the hospital, which enable us to evaluate the performance of different hospitals. However, data with such diverse sources and formats presents

challenges for machine learning based quantitative analysis. For example, the data contains various data format including both number and strings. Even for numbers, some of them are categorical or ordinal. In addition, there are missing values, invalid values and potentially outliers due to reasons such as human error. Therefore, the traditional machine learning methods cannot be readily applied. After examining all 481 variables, I identified 20 variables that could potentially affect hospital performance and listed them in Table 4.1.

Table 4.1 Key variables used in regression model

Variable	Type	Classification
Specialization index (I_S)	Continuous	Independent
Total charge (TOTCHG)	Continuous	Dependent[1]
Died during hospitalization (DIED)	Categorical	Dependent[2]
Length of stay (LOS)	Continuous	Control[3]
No. of diagnosis (NDX)	Continuous	Control[3]
No. of procedure (NPR)	Continuous	Control[3]
Age	Continuous	Control
No. of chronical conditions	Continuous	Control
Gender	Categorical	Control
Race	Categorical	Control
Severity of illness	Categorical	Control
Risk of mortality	Categorical	Control
Hospital location	Categorical	Control
Hospital region	Categorical	Control
Wage index	Categorical	Control
Hospital bed size	Categorical	Control
Diagnosis Related Group (DRG)	Categorical	Control
Hospital ownership	Categorical	Control
Hospital teaching status	Categorical	Control
Payment type	Categorical	Control

[1]TOTCHG is a control variable in analyzing the effect of I_S on DIED.
[2]DIED is a control variable in analyzing the effect of I_S on TOTCHG.
[3] LOS, NDX and NPR are mediator variables in path analysis of the indirect effect of I_S on TOTCHG.

To reduce the effect of confounding variables/factors, in observational (vs. experimental) study like this work, I can either add filters to make the other variables taking fixed values, or include

the confounding variables in the regression. Because the former dramatically reduces the number of observations/cases to be included in a model which reduces the statistical power of the analysis, I choose the latter approach as indicated in Table 4.1. It is worth noting that it would be ideal if I could analyze some DRG's individually as well as collectively to see if the effects of specialization on TOTCHG are consistent across different DRG's. Also, when DRG's are analyzed collectively, it would be ideal if they are associated with fundamentally different procedures or physiological systems. Unfortunately, the 2012 HCUP data I have do not have sufficient samples for such analyses. Instead, I select five most expensive DRG's based on the following three criteria: (a) Each DRG makes a different contribution to the inpatient quality indicator (IQI) defined by AHRQ; (b) The selected DRG's TOTCHG must be high compared to that of other DRG's because the DRG's with low TOTCHG may not reflect the characteristics of specialization; and (c) The selected DRG's must have enough observations. In addition, I focus on non-maternal adult patients only. Based on the above criteria, I select the five most expensive DRG's based on the national median TOTCHG with at least 4,500 cases. The descriptive statistics of the five selected DRG's are listed in Table 4.2. Any case with one or multiple missing, invalid or outlier values was excluded from this study. After these preprocessing steps, totally 86,999 cases were included in this study. It is worth noting that although three selected DRG's are associated with cardiovascular health, each DRG contributes to a different IQI: DRG 233 is associated with the coronary artery bypass graft (CABG) mortality rate; DRG 238 is associated with the abdominal aortic aneurysm (AAA) repair mortality rate; and DRG 246 is associated with the acute myocardial infarction (AMI) mortality rate. Nevertheless, I acknowledge the above-mentioned limitations of this study, which are undoubtably worth further investigation should I obtain additional data.

Table 4.2 The five most expensive DRG's with at least 4,500 cases in the 2012 HCUP-NIS dataset

DRG	**No. of cases**	**Mean TOTCHG**	**Median TOTCHG**	**Std. Deviation**	**DRG Description**
233	4,634	$206,471	$170,646	$133,408	CORONARY BYPASS W CARDIAC CATH W MCC[1]
25	5,420	$135,469	$102,905	$116, 583	CRANIOTOMY & ENDOVASCULAR INTRACRANIAL PROCEDURES W MCC
238	5,325	$99,803	$87,900	$53,735	MAJOR CARDIOVASC PROCEDURES W/O MCC
246	10,755	$99,010	$82,551	$67,254	PERC CARDIOVASC PROC MCC
470	60,865	$55,190	$48,929	$27,826	MAJOR JOINT REPLACEMENT W/O MCC

[1] MCC: multiple chronic conditions

4.3 Methods

In this section, I introduce the specialization index used to quantify hospital specialization. I also briefly review pure data-driven ML approaches used in this work. To address the limitations of pure data-driven ML approaches, I propose a knowledge-guided ML approach (*i.e.*, path analysis) to investigate the complete effect of hospital specialization on hospital performance in terms of cost and patient outcome. Finally, I introduce measures used to quantify the statistical significance of regression coefficients and model goodness-of-fit.

4.3.1 Specialization quantification

Farley and Hogan (1990) have proposed an index of specialization based on information theory. It has been shown that the index provided intuitively reasonable results in characterizing patterns of specialization across hospitals. In Farley and Hogan (1990), let Φ_i represent the baseline proportion of cases in DRG category *i*, and let p_{ih} denote the proportion of cases in the h^{th} hospital observed in DRG category *i*. The information theory index of specialization (ITI) for hospital *h* collapses information about differences between the Φ_i's and p_{ih}'s as follows [132]:

$$ITI_h = \sum_{i=1}^{I}\{p_{ih} \cdot ln(p_{ih}/\Phi_i)\} \quad (4.1)$$

In this work, because I only focus on the five DRG's listed in Table 4.2, I define a specialization index (I_S) of h^{th} hospital by modifying equation 4.1 as follows:

$$I_{sh} = \sum_{i=1}^{5}\{w_{ih} \cdot ln(p_{ih}/\Phi_i)\} \quad (4.2)$$

where $w_{ih} = p_{ih}/\sum_{i=1}^{5} p_{ih}$ such that $\sum_{i=1}^{5} w_{ih} = 1$.

4.3.2 Pure data-driven machine learning approaches

In this work, the following pure data-driven machine learning approaches are used to investigate the effect of specialization on hospital performance [133].

- **Multiple linear regression (MLR)**: MLR is performed to investigate the effect of each independent variable on total charge (TOTCHG). The independent and control variables include the following continuous variables: specialization index (I_S), wage index, length of stay (LOS), age, number of chronical conditions, number of diagnosis (NDX), number of procedures (NPR); as well as the following categorical variables: bed-size (small/medium/large), location (rural/urban), teaching status (non-teaching/teaching), ownership (public/private not-for-profit/private for profit), region (northeast/Midwest/south/west), payment (private/Medicare/Medicaid/others), DRG type (470/233/25/238/ 246), race (white/black/Hispanic/others), sex (male/female), severity of illness (minor/moderate/major/extreme) and risk of mortality (minor/moderate/major/extreme). Dummy variables are introduced for categorical variables and the first variable in each category was used as the reference. In this work, it is made sure that the following assumptions are satisfied for MLR: (a) linearity, i.e., the relationships between independent and dependent variables are linear; (b) constant variance of residuals, i.e., random error terms are independent and normally distributed with zero mean and constant variance. In order to satisfy the above two assumptions, I

consider transformation of variables. For LOS, which is right skewed, the square root helps to meet the constant variance of residuals assumption. For TOTCHG, which is more strongly right skewed than LOS. It is found that the 4^{th} root of TOTCHG helps satisfy the above two assumptions. After variable transformations, all variables are scaled or standardized to zero mean and unit variance. The same variable transformations (for LOS and TOTCHG) and standardization (for all variables) are used for all analyses throughout this work. The MLR analysis is performed in R.

- **Principal Component Regression (PCR) with mixed variables**: The same transformed and standardized dependent and independent variables used in MLR are used in principal component regression (PCR). The standard PCR cannot be directly applied for this case because of the categorical variables. Therefore, the R package of PCAmixdata is used, which handles a mixture of qualitative and quantitative variables. Because of the large number of independent variables and potential multicollinearity among them, I expect that PCR could address the potential multicollinearity while reducing the impact of noise through dimensionality reduction.
- **Principal Component Regression (PCR) and partial least squares (PLS) with continuous variables only**: I also perform standard PCR and PLS using the following continuous variables only: specialization index (I_S), wage index, length of stay (LOS), age, number of chronical conditions, number of diagnosis, number of procedures, and total discharge (to represent hospital bed size). All variables are transformed (for LOS and TOTCHG) and standardized.
- **Fisher discriminant analysis (FDA)**: In this analysis, all cases are divided into quartiles based on TOTCHG. Only the lowest TOTCHG group (*i.e.*, the bottom 25%) and the

highest TOTCHG group (*i.e.*, the top 25%) are used in Fisher discriminant analysis (FDA) to find which variables contribute highly to the separation of these two groups, and whether specialization (I_S) is one of them. The same transformed and standardized variables used in MLR are used in FDA, which is performed in SPSS.

- **Ordinal regression (OR)**: To reduce the noise in TOTCHG, I categorize TOTCHG for all cases into four groups and use ordinal regression (OR) to investigate the effect of specialization. In this study, TOTCHG is divided into four groups based on the quartiles of cases (*i.e.*, dividing all cases into four TOTCHG groups containing an approximately equal number of cases in each group). The same transformed and standardized variables used in MLR are used in the ordinal regression, which is performed in SPSS.
- **Logistic regression (LR)**: LR is performed to investigate the effect of each independent variable on hospital performance in terms of patient outcome, which is measured by the death of patient during hospitalization (coded DIED in HCUP). The dependent variable is DIED (alive/died). The independent/control variables are the same ones used in MLR with the exclusion of DIED and addition of TOTCHG. LR is performed in SPSS.

It is worth noting that there are many statistical and machine learning methods, linear or nonlinear, have been developed and applied to analyze healthcare data. For example, generalized additive models (GAMs) have been extensively used for nonlinear regressions [134]. Interested readers are referred to some recent review articles on the subject [124], [135]–[138].

4.4 Proposed knowledge-guided path analysis

For pure data-driven machine learning approaches discussed in Section 4.3.2, variables are divided into dependent or response variable and independent or explanatory variables, and relationship or linkage between the independent and dependent variables are estimated by

minimizing certain objective function such as sum of squared residuals (SSR) or maximizing discriminant ratio (DR). This type of analyses do not take any relationship among independent variables into account other than confounding effects. In other words, they can only reveal the direct effects. However, by discussing with experts in health services administration, it is recognized that direct effect alone does not reveal the full picture of the relationship between specialization and total charge. Based on domain knowledge, I hypothesize that specialization could also influence total charge through indirect effects. Specifically, if specialization improves hospital performance, it can achieve so through reducing length of stay (LOS) at the hospital, number of diagnosis (NDX) conducted, and number of procedures (NPR) performed. This indirect effect is also known as mediation or mediator effect, which occurs when a change in a dependent variable is driven by the successive change of an independent variable and a third variable, termed as a mediator [139]. A change in an independent variable causes a change in a mediator and this change leads to a change in a dependent variable. Therefore, I examine these potential indirect effects through path analysis (PA) [140], [141], which can capture an indirect path from an independent variable to a dependent variable through a mediator [140], [142]. By applying PA, I examine not only the direct effect, but also the indirect effect of the specialization (I_s) on total charge (TOTCHG) via three potential mediator variables – length of stay (LOS), number of procedures (NPR) and number of diagnoses (NDX). PA is established by the following equations in this work.

$$M_1 = a_1 + a_{X1}X + \sum_{i=1}^{29} a_{1i} C_i + e_{M1} \quad (4.3)$$

$$M_2 = a_2 + a_{X2}X + \sum_{i=1}^{29} a_{2i} C_i + e_{M2} \quad (4.4)$$

$$M_3 = a_3 + a_{X3}X + \sum_{i=1}^{29} a_{3i} C_i + e_{M3} \quad (4.5)$$

$$Y = b_0 + b_{X1}X + b_{M1}M_1 + b_{M2}M_2 + b_{M3}M_3 + \sum_{i=1}^{29} b_i C_i + e_Y \quad (4.6)$$

where all the variables are listed in Table 4.3 and all the e terms are model errors or residuals.

Equations 4.3 – 4.5 are used for examining the relationship between a mediator (M) and an independent variable (X), which represents the first stage of an indirect effect of X on dependent variable (Y). In other words, these models would determine if I_s influences mediators - LOS, NPR and NDX. After substituting equations 4.3 – 4.5 into equation 4.6 and rearranging the terms, I yield equation 4.7.

$$Y = X[b_{X1} + a_{X1}b_{M1} + a_{X2}b_{M2} + a_{X3}b_{M3}] + [b_0 + a_1 b_{M1} + a_2 b_{M2} + a_3 b_{M3} + b_{M1}\sum_{i=1}^{29} a_{1i}C_i + b_{M2}\sum_{i=1}^{29} a_{2i}C_i + b_{M3}\sum_{i=1}^{29} a_{3i}C_i + \sum_{i=1}^{29} b_i C_i] + b_{M1}e_{M1} + b_{M2}e_{M2} + b_{M3}e_{M3} + e_Y \quad (4.7)$$

Equation 4.7 describes the total effect of X on Y in the coefficient term $[b_{X1} + a_{X1}b_{M1} + a_{X2}b_{M2} + a_{X3}b_{M3}]$, which combines the direct and indirect effects where each indirect effect is explained by the product of the first stage effect and the second stage effect. All the effects are depicted in Figure 4.1.

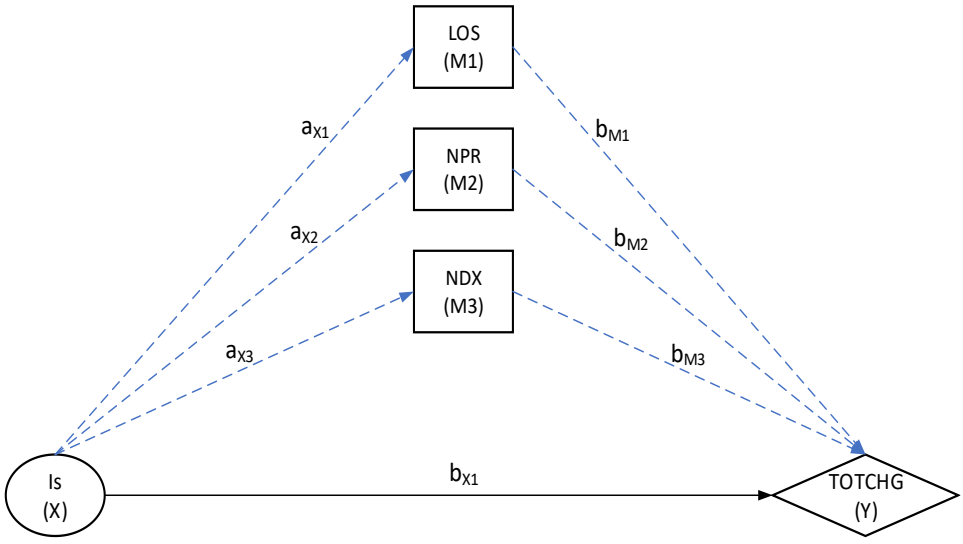


Figure 4.1 Path analysis for direct effect (denoted as black solid line) and indirect effects (denoted as blue dashed line)

Table 4.3 Full list of variables for path analysis

Variable	Description	Function	Type
$\boldsymbol{X}$	Specialization Index (I_s)	Independent variable	Continuous
$\boldsymbol{Y}$	Total charge (TOTCHG$^{1/4}$)	Dependent variable	Continuous
$\boldsymbol{M_1}$	Length of stay (LOS$^{1/2}$)	Mediator 1	Continuous
$\boldsymbol{M_2}$	Number of procedures (NPR)	Mediator 2	Continuous
$\boldsymbol{M_3}$	Number of diagnosis (NDX)	Mediator 3	Continuous
$\boldsymbol{C_1}$	WAGE INDEX	Control variable	Continuous
$\boldsymbol{C_2}$	AGE	Control variable	Continuous
$\boldsymbol{C_3}$	Number of chronic condition (NCHRONIC)	Control variable	Continuous
	Bedsize reference_Small		Dummy
$\boldsymbol{C_4}$	Bedsize_Medium	Control variable	Dummy
$\boldsymbol{C_5}$	Bedsize_Large	Control variable	Dummy
$\boldsymbol{C_6}$	Location_Urban	Control variable	Binary
$\boldsymbol{C_7}$	Teaching status	Control variable	Binary
	Ownership reference_Public		Dummy
$\boldsymbol{C_8}$	Owenership_Private not-for-profit	Control variable	Dummy
$\boldsymbol{C_9}$	Owener_Private for profit	Control variable	Dummy
	Region reference_Northeast		Dummy
$\boldsymbol{C_{10}}$	Region_Midwest	Control variable	Dummy
$\boldsymbol{C_{11}}$	Region_South	Control variable	Dummy
$\boldsymbol{C_{12}}$	Region_West	Control variable	Dummy
	Pay source reference_Private		Dummy
$\boldsymbol{C_{13}}$	Pay source_Medicare	Control variable	Dummy
$\boldsymbol{C_{14}}$	Pay source_Medicaid	Control variable	Dummy
$\boldsymbol{C_{15}}$	Pay source_Others	Control variable	Dummy
	DRG reference_DRG470		Dummy
$\boldsymbol{C_{16}}$	DRG_DRG233	Control variable	Dummy
$\boldsymbol{C_{17}}$	DRG_DRG25	Control variable	Dummy
$\boldsymbol{C_{18}}$	DRG_DRG238	Control variable	Dummy
$\boldsymbol{C_{19}}$	DRG_DRG246	Control variable	Dummy
	Race reference_White		Dummy
$\boldsymbol{C_{20}}$	Race_Black	Control variable	Dummy
$\boldsymbol{C_{21}}$	Race_Hispanic	Control variable	Dummy
$\boldsymbol{C_{22}}$	Race_Others	Control variable	Dummy
$\boldsymbol{C_{23}}$	Sex	Control variable	Binary
	Severity reference_Minor		
$\boldsymbol{C_{24}}$	Severity_Moderate	Control variable	Dummy
$\boldsymbol{C_{25}}$	Severity_Major	Control variable	Dummy
$\boldsymbol{C_{26}}$	Severity_Extreme	Control variable	Dummy
	Mortality reference_Minor		
$\boldsymbol{C_{27}}$	Mortality_Moderate	Control variable	Dummy
$\boldsymbol{C_{28}}$	Mortality_Major	Control variable	Dummy
$\boldsymbol{C_{29}}$	Mortality_Extreme	Control variable	Dummy

The models (*i.e.,* equations 4.3 – 4.6) are estimated by ordinary least squares (OLS) regressions to estimate the parameters a's and b's as depicted in Figure 4.1, and significant coefficients are identified by p-values. Once I estimate the parameters a's and b's, the overall relationship described by equation 4.7 is followed to calculate the direct, indirect, and total effects of specialization (I_s) on total charge (TOTCHG). PA was performed in SPSS.

Coefficient statistical significance and model goodness-of-fit

The statistical significance of a coefficient is measured by its p-value and model goodness-of-fit is measured by adjusted R^2.

- **P-value**: The statistical significance of a coefficient is measured by its p-value through bootstrapping and t-test. I performed random sampling 1,000 times as it is typical in bootstrapping for estimating a distribution [143]. For the test of statistical significance of coefficients generated by the product of coefficients from these equations such as indirect and total effect, bootstrapping was employed to obtain the sampling distribution of the coefficients and bias-corrected confidence intervals [140], [143].
- **Adjusted R^2**: In this study, the adjusted R^2 [144] is used to compare the explanatory power of regression models. The adjusted R^2 is a modified version of R^2 that has been adjusted for the number of predictors in the model. The adjusted R^2, which is always lower than R^2, increases only if the new independent variable improves the model more than by chance. Similarly, the adjusted McFadden's R^2 [145] was used for logistic regression.

4.5 Results

Preliminary analysis and visualization

For the five most expensive DRG's listed in Table 4.2, there are totally 8,699 cases with the histogram of specialization index (I_S) shown in Figure 4.2, which is approximately normally distributed after standardization.

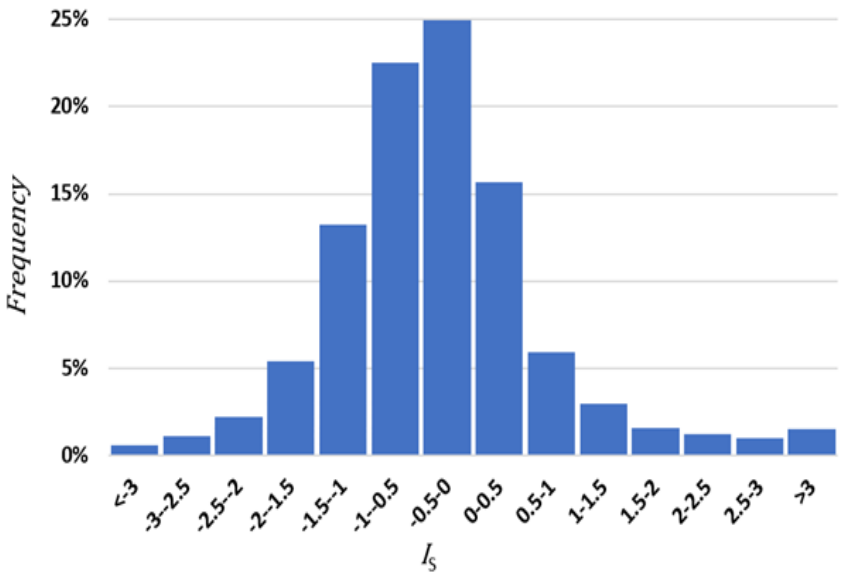
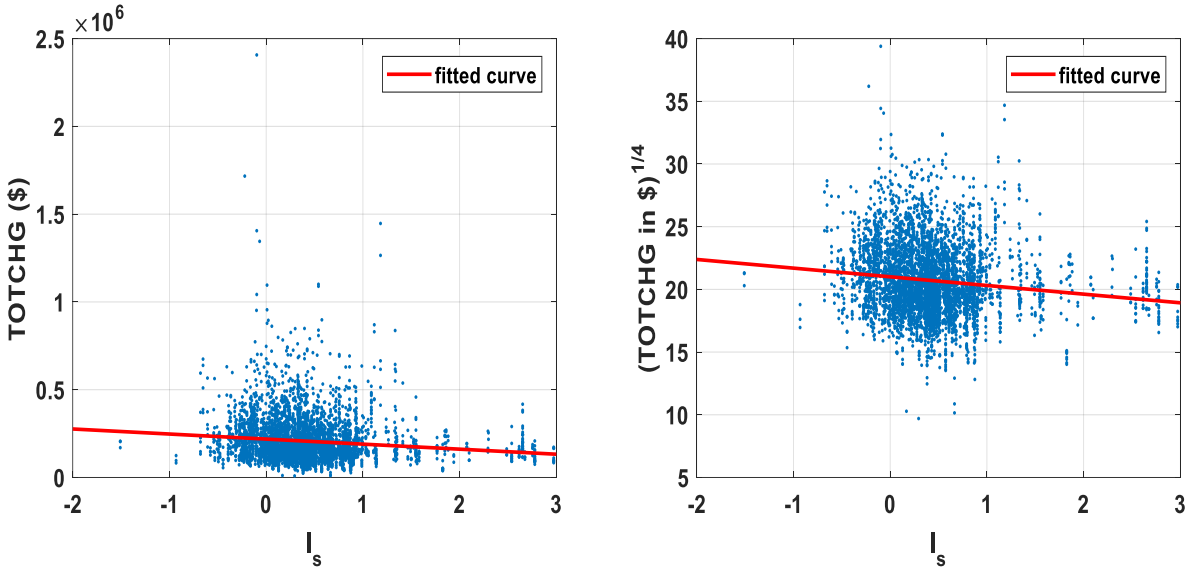


Figure 4.2 Histogram shows approximately normal distribution of I_S after standardization

To visually inspect the effect of specialization on the total charge, in Figure 4.3 (a), I plot TOTCHG vs. I_s for all cases of the five expensive DRG's listed in Table 4.2. Because TOTCHG is strongly right skewed, it can be seen from Figure 4.3 (a) that some of the positive residuals of a linear fit line are rather large. In contrast, I also plot TOTCHG$^{1/4}$ vs. I_s as shown in Figure 4.3 (b), which shows more balanced residuals of a linear fitting between TOTCHG$^{1/4}$ and I_S, supporting the choice of TOTCHG$^{1/4}$ as the dependent variable in all regression analyses rather than the original TOTCHG. It is worth noting that I do not control the confounding variables in generating Figure 4.3. Therefore, Figure 4.3 cannot provide the true quantitative relationship between I_S and TOTCHG$^{1/4}$.

I have also performed PCA on the combination of dependent variable TOTCHG$^{1/4}$ and all independent variables excluding I_s. In the score plot of the first and the third principal

components, I use different colors for different ranges of I_s as shown in Figure 4.4. Although the scores were generated without the inclusion of I_s, they are clustered for the same range of I_s, indicating the underlying correlation between I_s and other variables.



(a) (b)

Figure 4.3 (a) Scatter plot of TOTCHG vs. I_s (dots) with linear fitting (solid line); (b) Scatter plot of TOTCHG$^{1/4}$ vs. I_s (dots) with linear fitting (solid line), which shows better and more balanced residuals compared to (a).

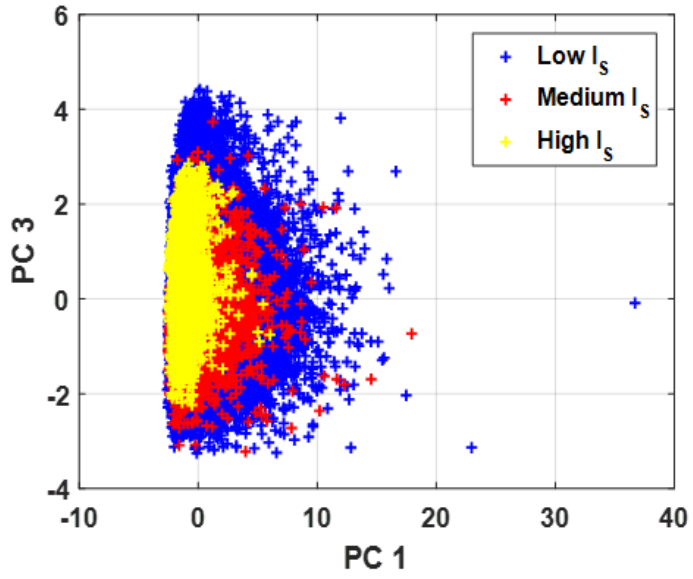


Figure 4.4 Score plot of 1st and 3rd principal components of PCA on all variables other than I_s

Effect of I_s on TOTCHG from data-driven machine learning approaches

Different data-driven machine learning approaches discussed in Section 4.3.2, including regression and discriminant analysis methods, are used to model the relationship between $TOTCHG^{1/4}$ and various independent variables and the results are listed in Table 4.4. For PCR and PLS, the Kaiser criterion is used to determine the number of principal components (PC's), where components with eigenvalues less than 1 were dropped. In other words, I only retained PCs that contain more information than the average information contained by each variable. The third column of Table 4.4 shows that I_S has a negative effect on $TOTCHG^{1/4}$ in all analyses, indicating that higher hospital specialization leads to lower TOTCHG and the p-values of I_S coefficients listed in the last column indicate that the effects are statistically significant in all regression models. The reasonably high adjusted R^2 values confirm the high quality of the regression models.

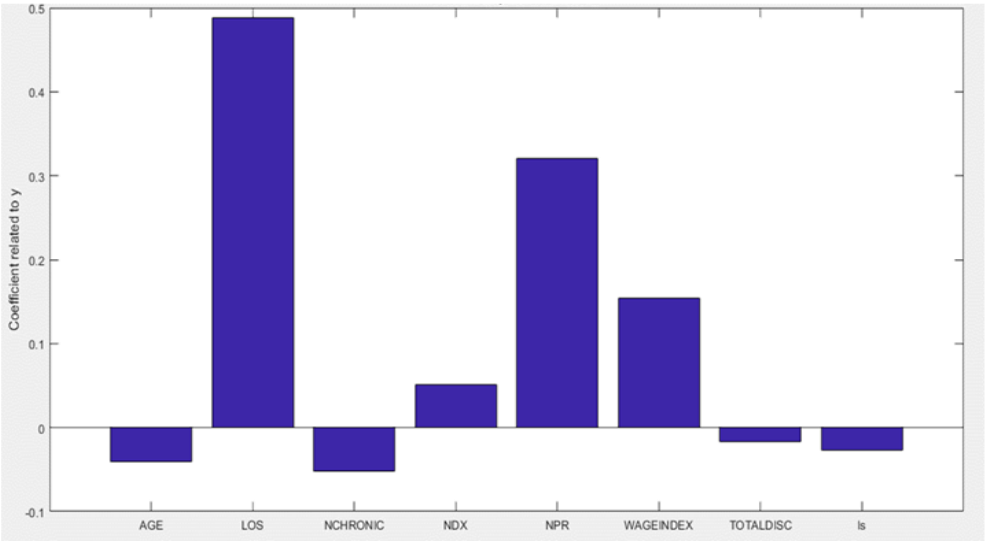
Table 4.4 The results of regression and discriminant analyses on the effect of I_S on $TOTCHG^{1/4}$

Method	# of PC's	Effect of I_S on $TOTCHG^{1/4}$	Adjusted R^2	p-value of I_s coefficient
MLR	-	(-)	0.5871	<0.0001
PCR (mixed variables)	11	(-)	0.5151	<0.0001
PCR (continuous variables only)	3	(-)	0.4009	<0.0001
PLS (continuous variables only)	2	(-)	0.5255	<0.0001
FDA	-	(-)	-	-
Ordinal regression	-	(-)	-	<0.0001

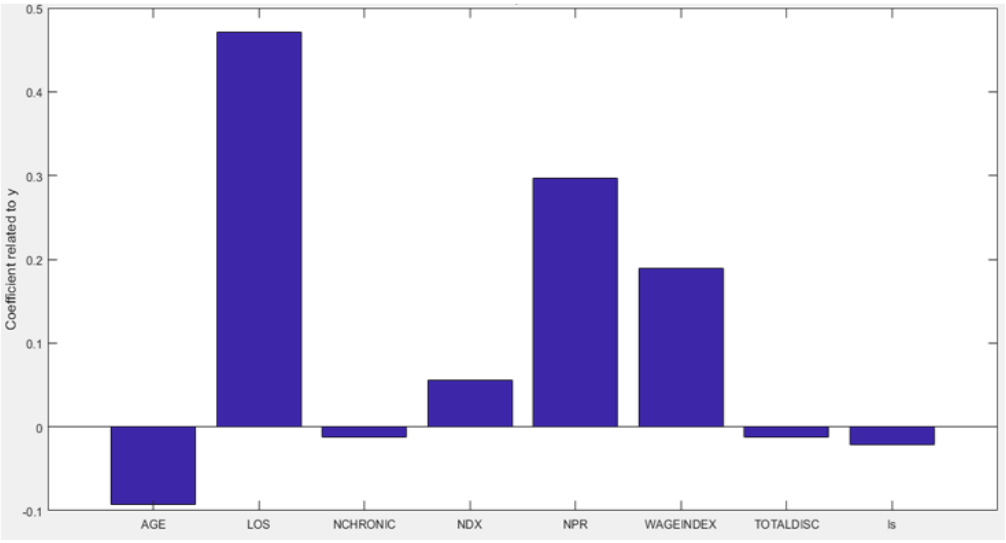
To quantitatively examine the effect of I_S on $TOTCHG^{1/4}$, I compare the contribution of I_S to $TOTCHG^{1/4}$ to those of other factors based on the regression models of PCR and PLS using the following eight continuous variables only: I_s, age, LOS, NCHRONIC, NDX, NPR, wage index (WAGEINDEX), and total discharge (TOTALDISC) to replace and represent hospital bed size, which is a categorical variable. The contribution plots based on PCR and PLS are shown in

Figure 4.5. As expected, LOS, NPR and WAGEINDEX are the top three positive contributors to TOTCHG$^{1/4}$. In other words, higher LOS/NPR/WAGEINDEX leads to higher TOTCHG. On the other hand, age, I_s, and TOTALDISC are the top three negative contributors to TOTCHG$^{1/4}$, although NCHRONIC is also ranked high in PCR. The negative effects of age and NCHRONIC are somewhat surprising. However, age, and hence NCHRONIC, are closely related to insurance or Pay Source such as private, Medicare and Medicaid, which is not modeled. Therefore, it makes sense in this case because elderly patients are usually having higher number of chronical conditions and often paid by Medicare/Medicaid, which is usually charged less. For the main target I_s that I am focusing on, although it has higher influence on reducing total charge compared to hospital size, its influence dwarfs those of LOS, NPR, WAGEINDEX and NDX.

However, data-driven machine learning approaches can only reveal the direct effect of I_s. Based on domain knowledge, I hypothesize that I_s could influence total charge through reducing LOS, NDX and NPR, three of the four most influential factors revealed in Figure 4.5. Therefore, I examine these potential indirect or mediator effects through path analysis (PA) in the following part.



(a)



(b)

Figure 4.5 Contribution of various factors to TOTCHG by (a) PCR and (b) PLS

Effect of I_s on TOTCHG from knowledge-guided path analysis

As discussed in Section 4.4, based on the knowledge-guided hypotheses, equations 4.3 – 4.6 are used to describe the mediator and direction effects. The parameters a's and b's in equations 4.3 – 4.6 are estimated using OLS. Bootstrapping (*i.e.*, random sampling 1,000 times) is employed to obtain the sampling distribution of the coefficients and bias-corrected confidence intervals [140], [143]. Significant coefficients are identified by p-values of t-test. All modeling and calculation are carried out using SPSS. Table 4.5 lists a_{X1}, a_{X2}, and a_{X3}, which are standardized coefficients estimated from equations 4.3 – 4.5, where standardized LOS$^{1/2}$, NPR, and NDX are dependent variables, respectively. b_{X1}, b_{M1}, b_{M2}, and b_{M3} are standardized coefficients estimated from equation 4.6 where standardized TOTCHG$^{1/4}$ is the dependent variable. The high values of adjusted R^2 in Table 4.5 for all four models (*i.e.*, equations 4.3 – 4.6) indicate good fitting of the models to the observations/cases. Table 4.5 indicates that specialization leads to decrease in LOS$^{1/2}$ at the confidence level of 0.01, decrease in NDX at the confidence level of 0.05, while having no effect on NPR. On the other hand, specialization, leads

to decrease in TOTCHG$^{1/4}$ directly at the confidence level of 0.01. LOS and NPR have positive effects on TOTCHG$^{1/4}$ at the confidence level of 0.01 but NDX has no effect on TOTCHG$^{1/4}$.

Table 4.5 Estimated coefficients and their p-values for I_S and mediators (M1 – M3)

Parameter	a_{X1}	a_{X2}	a_{X3}	b_{X1}	b_{M1}	b_{M2}	b_{M3}
Value	-0.09**	0.003	-0.004*	-0.041**	0.363**	0.183**	-0.003
Adjusted R^2	0.513	0.717	0.791	0.590			

*p < 0.05 ** p < 0.01

By considering both stages, I obtain the indirect effects of I_S on TOTCHG$^{1/4}$ as listed in Table 4.6, which are calculated with coefficients in Table 4.5 based on equation 4.7. Table 4.6 indicates that only the indirect effect of I_S on TOTCHG$^{1/4}$ through LOS$^{1/2}$ is statistically significant, which leads to decreased TOTCHG, while the other two indirect effects are statistically insignificant. Table 4.7 summarizes the direct, indirect and total effects of I_S on TOTCHG$^{1/4}$, where the total effect is the summation of direct and indirect effects. Table 4.7 shows that the negative effect of the I_s on TOTCHG is strengthened by the mediator effect, resulting in greater total effect than direct effect alone. This full picture is revealed only when ML is guided by domain knowledge, i.e., by defining the model structure based on knowledge-guided hypotheses. From managerial point of view, operational efficiency and physician effectiveness are two important factors determining TOTCHG [123], [124]. I argue that the specialization is associated with both operational efficiency and physician effectiveness. Specifically, the first factor, operational efficiency, is likely connected to the direct effect. This is because hospital specialization not only leads to optimal allocation of resources to right places, but also reduces needless waste of materials and processing time [123]. The other factor, physician effectiveness, is represented by the indirect effects. A physician's decisions influence how many and what procedures a patient receives, and together with the physician's experiences

and effectiveness, affect how long the patient needs to stay in the hospital, which in turn impacts TOTCHG.

Table 4.6 Indirect effects of I_S on TOTCHG$^{1/4}$ through mediators

Mediator	**LOS$^{1/2}$**	**NPR**	**NDX**
Parameter	$a_{X1}b_{M1}$	$a_{X2}b_{M2}$	$a_{X3}b_{M3}$
Value	-0.033**	0.001	-0.00001

*p < 0.05 ** p < 0.01

Table 4.7 Direct, indirect, and total effect of I_S on TOTCHG$^{1/4}$ through mediators

Effect	**Direct**	**Indirect**	**Total**
Value	-0.041**	-0.033**	-0.074**

*p < 0.05 ** p < 0.01

Effect of I_S on patient outcome

For hospital performance in terms of patient outcome, which is measured by patient death during hospitalization (coded DIED in HCUP) in this work, I believe the potential mediators are doctors or surgeons' experiences, hospital equipment, etc. However, because I do not have these information from the HCUP data, I have no means to analyze or control these potential mediators. Therefore, I could not perform separate analyses to quantify the direct and indirect effects of specialization on hospital performance in terms of patient outcome. Instead, I performed logistic regression (LR) to estimate the total effect of specialization on patient outcome. The dependent variable is DIED (alive/died). The independent variables are the same ones used in MLR. Since the original data is highly imbalanced (i.e., ~98% alive vs. ~2% died during hospitalization), I also performed a more balanced LR using synthetic minority over-sampling technique (SMOTE) [146] to obtain an improved ratio. The adjusted McFadden's R^2 is used to compare the explanatory power of regression models. The LR results based on the original data are listed in Table 4.8, which indicates that I_S has a negative effect on DIED. In

other words, specialization reduces patient death during hospitalization. The adjusted McFadden's R^2 is 0.403, which is greater than the recommended range (>0.2~0.4), indicating excellent modeling of the data with LR. By deploying SMOTE, the alive-to-died ratio is improved to 82.3% alive vs. 17.7% died. Table 4.8 shows that the model quality is improved by SMOTE as indicated by the higher adjusted McFadden's R^2. The balanced LR analysis also indicates that I_S leads to better patient outcome (i.e., lower DIED).

Table 4.8 Effect of I_S on DIED based on logistic regression

Data	Effect of I_S on DIED	Adjusted McFadden's R^2
Original imbalanced data	(-)	0.403
SMOTE balanced data	(-)	0.528

4.6 Summary and conclusions

In this work, using a national healthcare cost and utilization project (HCUP) dataset, I apply pure data-driven and knowledge-guided ML approaches to investigate direct, indirect, and total effects of hospital specialization on hospital performance in terms of cost and patient outcome.

The results show that pure data-driven ML approaches only reveal direct effect of specialization (I_S) on total charge (TOTCHG), which does not explain the complete relationship between I_S and TOTCHG. To address this limitation, I propose a knowledge-guided ML approach (*i.e.*, path analysis) by defining model structures based on domain knowledge and hypothesis. The results show that specialization reduces TOTCHG both directly and indirectly (through reducing LOS). In addition, the results demonstrate that specialization positively influences patient outcome (*i.e.*, reducing patient death during hospitalization). One of the reasons specialization improves patient outcome is that it has a positive impact on physician effectiveness. This is consistent with other studies indicating physicians are likely to have more

opportunities to improve their expertise in treating specific diseases, resulting in more effective treatment – a better outcome at low cost. The results of this study indicate that the focused factory theory is applicable to hospital operation as well as manufacturing operation and specialization is a vital factor in improving hospital performance.

Chapter 5. Contributions and Proposed Future work

5.1 Summary of contributions

In this work, I aim to advance industrial process monitoring and soft sensor techniques by developing hybrid machine learning techniques through feature engineering, feature selection, and domain knowledge integration. The proposed knowledge-guided feature engineering and feature selection help to extract physically and/or statistically meaningful and relevant features from process variables and to select informative and predictive features that should be included in the model, respectively. This study also shows that models integrated with domain knowledge improve their performance and usefulness in terms of predictive power and interpretability compared to models without domain knowledge. These contributions are demonstrated in three applications detailed in the following subsections.

5.1.1 Spectroscopy-based soft sensor

Spectroscopic techniques such as near-infrared (NIR) spectroscopy have gained wide applications in various industries. As a result, various soft sensors have been developed to predict sample properties from spectroscopic readings because accurate prediction of sample properties aids in monitoring the quality of products and process conditions. In this work, a novel spectroscopy-based soft sensor is proposed by integrating a *feature engineering* approach – Statistics Pattern Analysis (SPA) – with a new *feature selection* approach – Consistency Enhanced Evolution for Variable Selection (CEEVS) – to improve the predictive accuracy and the consistency of feature selection.

One contribution of this work is a new variable evaluation metric, which is proposed to measure the importance of variables more reliably regardless of the choice of training samples. Both regression coefficients and VIP scores are used to calculate the importance of variables

(*i.e.*, stability and probability), which plays an important role in improving the consistency of variable selection. In addition, a novel evolution process is proposed for efficient variable selection. Unlike GA, where the initial population is generated randomly, the initial population of CEEVS is produced based on the variable stability and probability, which enables it to include more important variables. This helps CEEVS to find truly relevant variables accurately and consistently. The evolution process has a mechanism in which less important variables can be re-evaluated during the variable selection process. This step prevents a variable of lower stability by itself yet still informative when combined with other variables from being eliminated. Another contribution of this work is that many tuning parameters are eliminated in CEEVS compared to the GA method, making CEEVS easy to implement and reducing the risk of being stuck in a local minimum. Finally, it is proven that the performance of CEEVS is not sensitive to the selection of tuning parameters if respective tuning parameters are determined at recommended range.

Since CEEVS employs the linear PLS-based framework, there are areas for further improvement of predictive power with the consideration of nonlinearity. Therefore, I propose a new method that integrates the SPA framework (a feature engineering approach) with CEEVS. Despite the successful application of the SPA framework to process monitoring and soft sensors in the past, there was a lack of explanation of how the SPA framework can handle nonlinearity. In this work, I demonstrate that features can capture the nonlinear relationship between spectral readings and sample properties. Through SPA framework, the linear PLS-based SPA-CEEVS method is able to outperform complex nonlinear methods such as SVR and GPR as well as the linear model-based variable selection methods such as CARS, SVP, GA and Elastic Net.

5.1.2 Process monitoring for PSA processes

In recent years, with increasing demand for tighter product quality monitoring and reliable process operation, process monitoring has been widely applied in a variety of fields. In this work, a new process monitoring method, FSM-kNN, is developed for fault detection and diagnosis for PSA processes. In FSM framework, statistics and morphological features are extracted from each step of a cycle, which can handle asynchronous pressure profiles due to unequal step and/or cycle duration. To the best of my knowledge, the utilization of step-wise features for monitoring periodic processes has not been studied. In addition, I propose criteria to classify fifteen steps of the PSA process into four groups based on the process characteristics of each step. This classification provides rough guidance on how to select the features of each group. By utilizing the statistics/features of pressure profiles of each step instead of pressure profiles themselves, the proposed FSM-kNN improves the detection performance compared to both PCA-based and kNN rule-based fault detection methods.

The FSM-kNN employs kNN rule for fault detection. The basic idea of the FSM-kNN method is that a distance between a faulty cycle and its nearest neighboring training cycles is greater than that between a normal cycle and its neighboring training cycles. Since the kNN method is a nonparametric ML approach, the proposed method makes no assumption about the linearity of the data set. Therefore, the FSM-kNN is well suited for monitoring PSA process condition, where nonlinear behaviors are dominant. In the proposed method, a time constraint is considered to build a training pool. This constraint helps avoid the occurrence false negative, reducing missed fault detection rates.

In this work, I propose a step-wise fault diagnosis method. The proposed fault diagnosis is the contribution-based approach, where a step with the most significant contribution is

regarded as the root cause of the fault. Through the proposed fault diagnosis, FSM-kNN outperforms all existing representative fault detection and diagnosis methods and is the only method that can correctly identify the root cause of faults for all fault scenarios.

5.1.3 Knowledge-guided path analysis

The United States has greater healthcare spending than any other developed country, and healthcare spending has increased more rapidly than GDP and wages. However, considering the huge outlay on healthcare, clinical outcomes are not improving proportionally. As hospital care represents the single largest national health expenditure by the type of services, it is expected that the better functioning of hospitals can significantly improve the efficiency of the whole healthcare system. In this work, I hypothesize that hospital specialization could be one of the potential solutions to improving the efficiency of hospital system based on focused factory theory in business field. Specifically, I examine whether the hospitals that are specialized in certain diseases achieve better hospital performance in terms of costs and patient outcomes. Because hospital operations and manufacturing processes share some similarities at systems level, in this work I employ some of the ML approaches frequently used as process systems engineering tools for process and control performance monitoring to quantify the level of specialization of a hospital for a certain disease and reveal the relationship between the specialization and hospital performance. These pure data-driven ML approaches deliver results that the specialization has a negative effect on total charge. However, the contribution of specialization to the total charge is small compared to other factors such as length of stay (LOS), number of procedures (NPR), and number of diagnosis (NDX). It has been recognized that the pure data-driven approaches to complex systems may lead to incomplete conclusions as they do not incorporate any domain knowledge on the system. To understand complete effect of

specialization on hospital performance, I propose a knowledge-guided path analysis. To the best of my knowledge, it is the first study to take possible paths that specialization can influence hospital performance into account to investigate the effect of specialization on hospital performance. I demonstrate that specialization not only has a direct effect on total charge through improved administrative efficiency and stronger negotiation power, but also has indirect effects on total charge through effective treatment (*i.e.*, reducing LOS). The proposed method delivers a comprehensive result which indicates domain knowledge plays a crucial role in machine learning applications.

5.2 Potential directions for future work

In this section, I suggest possible future research directions that can enable a better understanding of the methods studied in this dissertation and make further enhancements to the techniques.

5.2.1 Spectroscopy-based soft sensor

As discussed in this work, the SPA-CEEVS is a promising spectroscopy-based soft sensor in terms of the predictive power and understanding/interpretation of results. However, the SPA-CEEVS has a high computational cost since the method evaluates fitness for each round of evolution. This expensive computation is common for other variable selection methods based on Darwin's evolution theory of "survival of the fittest". One way to reduce computation burden is to set the threshold of stopping the offspring generation. Currently, the evolution process continues until the fitness of an offspring chromosome is better than that of the parent chromosome. Therefore, the evolution process is inevitably repeated many times until the method finds an optimal offspring chromosome. If an offspring chromosome that satisfies a specific threshold (*e.g.*, a minimum improvement in the fitness of the parent chromosome by α

percent) is regarded as the optimal one, the effort to search for the best offspring chromosome can be significantly reduced. Even though the optimal offspring chromosome determined with the threshold has slightly worse fitness value than that selected with current criteria, it will not considerably hurt the final performance. Since the optimal offspring chromosome is not directly used as the final variable subset, the offspring chromosome does not need to have the best fitness. Therefore, I believe that the adoption of threshold in the evolution process can improve computational effectiveness.

For many NIR datasets, spectroscopic readings are nonlinearly correlated with sample properties. Many researchers have studied nonlinear methods such as SVR, GPR, and ANN to improve predictive accuracy instead of the linear PLS model. Despite many applications of nonlinear methods, the study of nonlinear model-based variable selection has not been actively conducted. This work shows that CEEVS outperforms the other variable selection methods in terms of the predictive accuracy and consistency of variable selection. It is worth studying the integration of the CEEVS algorithm with nonlinear models such as SVR, XGBoost, and GPR for further improvement of predictive power.

As demonstrated in this study, statistical features extracted in each wavelength segment can describe the nonlinearity between input and output variables. Therefore, it is desirable to study more features based on domain knowledge to further improve the predictive power and better understand physical/chemical relationship between chemical functional groups and properties of interest. Especially, I suggest considering the robust features (*e.g.*, median, interquartile range, etc.) because some features I utilized in the study (*e.g.*, mean, standard deviation) are sensitive to outlier and/or spectral noises. Therefore, I expect that these features help make more robust model.

In this study, I proved the proposed method could significantly improve the feature selection consistency through special evolution process using real NIR datasets. However, it is sometimes difficult to fully understand the nature of the proposed method through real NIR datasets. Therefore, it is worth studying the simulation experiments which control specific factors – the proportion of the number of relevant input variables, the magnitude of correlations between input variables, and the magnitude of signal to noise – for better exploration of nature of the proposed method.

5.2.2 Process monitoring for PSA processes

The proposed FSM-kNN method is well suited to monitor periodic processes and to identify the root cause of faults. The framework first generates the statistics/features from the pressure profile of each step in a cycle. Then, similarity metrics between normal training cycles are measured using those features to detect faulty cycles, which have different similarity metrics from normal cycles. In an online detection, the framework requires that a cycle should be completed to monitor the abnormality of the cycle. To reduce the detection delay, a step-wise fault detection approach can be implemented. Currently, the FSM-kNN uses all the information from all steps to monitor the process condition. As a future work, individual FSM-kNN models can be built for monitoring the operation of each step. In other words, one FSM-kNN model is built for each step. In this way, fault detection can be immediately conducted right after each step is completed, instead of waiting for the entire cycle (*i.e.*, all steps) to complete, which can reduce the detection delay.

In addition, in this work I provide a rough way to select the statistics/features based on the known characteristics of the process. However, this approach may be incomplete when the process has a large number of variables. Therefore, an automatic feature selection algorithm is

ultimately required to eliminate redundant features that do not help monitor the process condition. The feature selection can be performed using either grid search or conventional feature selection methods such as GA, particle swarm optimization (PSO), and CEEVS. In this case, it can be a good practice to make a larger statistics/features pool by incorporating more statistics/features that capture the characteristics of periodic processes so that the feature selection methods can find a better feature subset.

It is also worth studying the weighted FSM-kNN method. The contribution of statistics/features to specific fault scenarios is different because they can capture other aspects of the process characteristics. Therefore, instead of assigning constant weight to all the features, the different weights of features can further improve the detection and diagnosis performance.

5.2.3 Knowledge-guided path analysis

In this work, I investigate if hospital specialization can affect the hospital performance in terms of cost and patient outcome. Since the 2012 HCUP data do not have sufficient samples for individual DRG's, I select five most expensive DRG's to conduct the hypothesis test. One future direction I suggest is to analyze some DRG's individually to see if the effects of specialization on hospital performance are consistent across different DRG's using several years' HCUP data that provide enough samples for each DRG. In addition, it is worth studying the effect of hospital specialization on hospital performance for five inexpensive DRG's to investigate if the effect of specialization depends on DRG's. Since this DRG group can be used as a moderator, the path analysis includes more paths where the specialization influences hospital performance. It enables the path analysis to reveal the full effect of specialization.

References

[1] S. Alonso, A. Morán, M. Á. Prada, P. Reguera, J. J. Fuertes, and M. Domínguez, "A data-driven approach for enhancing the efficiency in chiller plants: A hospital case study," *Energies*, vol. 12, no. 5, 2019.

[2] J. Wang, Q. Chang, G. Xiao, N. Wang, and S. Li, "Data driven production modeling and simulation of complex automobile general assembly plant," *Comput. Ind.*, vol. 62, no. 7, pp. 765–775, 2011.

[3] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemom. Intell. Lab. Syst.*, vol. 171, no. September, pp. 16–25, 2017, doi: 10.1016/j.chemolab.2017.09.021.

[4] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, 2012.

[5] T. Ngo and K. W. H. Tsui, "A data-driven approach for estimating airport efficiency under endogeneity: An application to New Zealand airports," *Res. Transp. Bus. Manag.*, vol. 34, no. January, p. 100412, 2020.

[6] J. MacGregor and A. Cinar, "Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods," *Comput. Chem. Eng.*, vol. 47, pp. 111–120, 2012.

[7] S. X. Ding, "Data-driven design of monitoring and diagnosis systems for dynamic processes: A review of subspace technique based schemes and some recent results," *J. Process Control*, vol. 24, no. 2, pp. 431–449, 2014.

[8] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, 2014.

[9] Z. Ge, Z. Song, and F. Gao, “Review of recent research on data-based process monitoring,” *Ind. Eng. Chem. Res.*, vol. 52, no. 10, pp. 3543–3562, 2013.

[10] P. Kadlec, B. Gabrys, and S. Strandt, “Data-driven Soft Sensors in the process industry,” *Comput. Chem. Eng.*, vol. 33, no. 4, pp. 795–814, 2009.

[11] A. Heng, S. Zhang, A. C. C. Tan, and J. Mathew, “Rotating machinery prognostics: State of the art, challenges and opportunities,” *Mech. Syst. Signal Process.*, vol. 23, no. 3, pp. 724–739, 2009, doi: 10.1016/j.ymssp.2008.06.009.

[12] C. C. Pantelides and J. G. Renfro, “The online use of first-principles models in process operations: Review, current status and future needs,” *Comput. Chem. Eng.*, vol. 51, pp. 136–148, 2013, doi: 10.1016/j.compchemeng.2012.07.008.

[13] N. Bhutani, G. P. Rangaiah, and A. K. Ray, “First-principles, data-based, and hybrid modeling and optimization of an industrial hydrocracking unit,” *Ind. Eng. Chem. Res.*, vol. 45, no. 23, pp. 7807–7816, 2006, doi: 10.1021/ie060247q.

[14] Y. Li, S. Billington, C. Zhang, T. Kurfess, S. Danyluk, and S. Liang, “Adaptive prognostics for rolling element bearing condition,” *Mech. Syst. Signal Process.*, vol. 13, no. 1, pp. 103–113, 1999, doi: 10.1006/mssp.1998.0183.

[15] Y. Li, T. R. Kurfess, and S. Y. Liang, “Stochastic prognostics for rolling element bearings,” *Mech. Syst. Signal Process.*, vol. 14, no. 5, pp. 747–762, 2000, doi: 10.1006/mssp.2000.1301.

[16] C. J. Li and H. Lee, “Gear fatigue crack prognosis using embedded model, gear dynamic model and fracture mechanics,” *Mech. Syst. Signal Process.*, vol. 19, no. 4, pp. 836–846, 2005, doi: 10.1016/j.ymssp.2004.06.007.

[17] Q. He and J. Wang, “Valve stiction modeling: First-principles vs data-drive approaches,”

Proc. 2010 Am. Control Conf. ACC 2010, pp. 3777–3782, 2010, doi: 10.1109/acc.2010.5531561.

[18] W. A. Shewhart, *Economic control of quality of manufactured product*. Van Nostrand, Princeton, 1931.

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

[20] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, 2012, doi: 10.1016/j.arcontrol.2012.09.004.

[21] J. Lee, J. Flores-Cerrillo, J. Wang, and Q. P. He, "Consistency-Enhanced Evolution for Variable Selection Can Identify Key Chemical Information from Spectroscopic Data," *Ind. Eng. Chem. Res.*, vol. 59, no. 8, pp. 3446–3457, 2020.

[22] J. Lee, J. Wang, J. Flores-Cerrillo, and Q. P. He, "Improving featured-based soft sensing through feature selection," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11338–11343, 2020, [Online]. Available: https://doi.org/10.1016/j.ifacol.2020.12.542.

[23] E. Tamburini, G. Vaccari, S. Tosi, and A. Trilli, "Near-infrared spectroscopy: A tool for monitoring submerged fermentation processes using an immersion optical-fiber probe," *Appl. Spectrosc.*, vol. 57, no. 2, pp. 132–138, 2003.

[24] 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.

[25] H. Li, Y. Liang, Q. Xu, and D. Cao, "Key wavelengths screening using competitive adaptive reweighted sampling method for multivariate calibration," *Anal. Chim. Acta*, 2009.

[26] D. W. Hopkins, "Shoot-out 2002: Transfer of Calibration for Content of Active in a

Pharmaceutical Tablet," *NIR news*, vol. 14, no. 5, pp. 10–13, 2003.

[27] R. M. Balabin, R. Z. Safieva, and E. I. Lomakina, "Gasoline classification using near infrared (NIR) spectroscopy data: Comparison of multivariate techniques," *Anal. Chim. Acta*, vol. 671, no. 1–2, pp. 27–35, 2010, [Online]. Available: http://dx.doi.org/10.1016/j.aca.2010.05.013.

[28] R. M. Balabin and E. I. Lomakina, "Support vector machine regression (SVR/LS-SVM) - An alternative to neural networks (ANN) for analytical chemistry? Comparison of nonlinear methods on near infrared (NIR) spectroscopy data," *Analyst*, vol. 136, no. 8, pp. 1703–1712, 2011.

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

[30] T. Chen, J. Morris, and E. Martin, "Gaussian process regression for multivariate spectroscopic calibration," *Chemom. Intell. Lab. Syst.*, vol. 87, no. 1, pp. 59–71, 2007.

[31] Y. Chen, S. S. Thosar, R. A. Forbess, M. S. Kemper, R. L. Rubinovitz, and A. J. Shukla, "Prediction of drug content and hardness of intact tablets using artificial neural network and near-infrared spectroscopy," *Drug Dev. Ind. Pharm.*, vol. 27, no. 7, pp. 623–631, 2001.

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

[33] C. M. Andersen and R. Bro, "Variable selection in regression-a tutorial," *J. Chemom.*, vol. 24, no. 11–12, pp. 728–737, 2010.

[34] I. G. Chong and C. H. Jun, "Performance of some variable selection methods when

multicollinearity is present," *Chemom. Intell. Lab. Syst.*, vol. 78, no. 1, pp. 103–112, 2005.

[35] R. Gosselin, D. Rodrigue, and C. Duchesne, "A Bootstrap-VIP approach for selecting wavelength intervals in spectral imaging applications," *Chemom. Intell. Lab. Syst.*, vol. 100, no. 1, pp. 12–21, 2010.

[36] P. Kump, E. W. Bai, K. S. Chan, B. Eichinger, and K. Li, "Variable selection via RIVAL (removing irrelevant variables amidst Lasso iterations) and its application to nuclear material detection," *Automatica*, vol. 48, no. 9, pp. 2107–2115, 2012.

[37] S. Wold, E. Johansson, and M. Cocchi, "3D QSAR in drug design: theory, methods and applications," *ESCOM*, pp. 523–550, 1993.

[38] V. Centner, D. L. Massart, O. E. De Noord, S. De Jong, B. M. Vandeginste, and C. Sterna, "Elimination of Uninformative Variables for Multivariate Calibration," *Anal. Chem.*, vol. 68, no. 21, pp. 3851–3858, 1996.

[39] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *J. R. Stat. Soc. Ser. B*, 1996, doi: 10.1111/j.2517-6161.1996.tb02080.x.

[40] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, 2005.

[41] R. Leardi, "Genetic algorithms in chemistry," *Journal of Chromatography A*, vol. 1158, no. 1–2. pp. 226–233, 2007.

[42] R. Leardi and A. Lupiáñez González, "Genetic algorithms applied to feature selection in PLS regression: How and when to use them," *Chemom. Intell. Lab. Syst.*, vol. 41, no. 2, pp. 195–207, 1998.

[43] R. Leardi, "Application of genetic algorithm-PLS for feature selection in spectral data sets," in *Journal of Chemometrics*, 2000, vol. 14, no. 5–6, pp. 643–655.

[44] J. Chen, C. Yang, H. Zhu, Y. Li, and W. Gui, “A novel variable selection method based on stability and variable permutation for multivariate calibration,” *Chemom. Intell. Lab. Syst.*, vol. 182, no. August, pp. 188–201, 2018, doi: 10.1016/j.chemolab.2018.09.009.

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

[46] J. Lee, J. Flores-Cerrillo, J. Wang, and Q. P. He, “A Variable Selection Method for Improving Variable Selection Consistency and Soft Sensor Performance,” *Proc. Am. Control Conf.*, vol. 2020-July, pp. 725–730, 2020.

[47] 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.

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

[49] 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.

[50] S. Wold, M. Sjostrom, and L. Eriksson, “PLS-regression: a basic tool of chemometrics,” *Chemom. Intell. Lab. Syst.*, vol. 58, pp. 109–130, 2001, [Online]. Available: http://dx.doi.org/10.1016/S0169-7439(01)00155-1.

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

[52] Z. Xiaobo, Z. Jiewen, M. J. W. Povey, M. Holmes, and M. Hanpin, “Variables selection methods in near-infrared spectroscopy,” *Anal. Chim. Acta*, vol. 667, no. 1–2, pp. 14–32, 2010, doi: 10.1016/j.aca.2010.03.048.

[53] B. Tan *et al.*, “Identification of free fatty acids profiling of type 2 diabetes mellitus and

exploring possible biomarkers by GC-MS coupled with chemometrics," *Metabolomics*, vol. 6, no. 2, pp. 219–228, 2010.

[54] D. Wu and D. W. Sun, "Application of visible and near infrared hyperspectral imaging for non-invasively measuring distribution of water-holding capacity in salmon flesh," *Talanta*, vol. 116, pp. 266–276, 2013.

[55] W. Fan, Y. Shan, G. Li, H. Lv, H. Li, and Y. Liang, "Application of Competitive Adaptive Reweighted Sampling Method to Determine Effective Wavelengths for Prediction of Total Acid of Vinegar," *Food Anal. Methods*, vol. 5, no. 3, pp. 585–590, 2012.

[56] D. Xu *et al.*, "Simultaneous determination of traces amounts of cadmium, zinc, and cobalt based on UV-Vis spectrometry combined with wavelength selection and partial least squares regression," *Spectrochim. Acta - Part A Mol. Biomol. Spectrosc.*, vol. 123, pp. 430–435, 2014.

[57] A. J. SMOLA and B. SCHOLKOPF, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, pp. 199–222, 2004, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=1CAD92EF8CCE726A305D8A41F873EEFC?doi=10.1.1.114.4288&rep=rep1&type=pdf%0Ahttp://download.springer.com/static/pdf/493/art%3A10.1023%2FB%3ASTCO.0000035301.49549.88.pdf?auth66=1408162706_8a28764ed0fae9.

[58] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," 1996.

[59] C. E. Rasmussen, "Evaluation of Gaussian Processes and other Methods for Non-Linear Regression," University of Toronto, 1997.

[60] K. Zheng *et al.*, "Stability competitive adaptive reweighted sampling (SCARS) and its applications to multivariate calibration of NIR spectra," *Chemom. Intell. Lab. Syst.*, vol.

112, pp. 48–54, 2012.

[61] “Corn dataset.” http://www.eigenvector.com/data/Corn/index.html.

[62] “Diesel Fuels dataset,” 2005. http://eigenvector.com/data/SWRI/index.html.

[63] “Pharmaceutical dataset.” .

[64] “Wheat dataset.” https://www.wiley.com/legacy/wileychi/chemometrics/datasets.html.

[65] L. Nørgaard *et al.*, “Interval partial least-squares regression (iPLS): a comparative chemometric study with an example from near-infrared spectroscopy,” *Appl. Spectrosc.*, vol. 54, no. 3, pp. 413–419, 2000, doi: 10.1366/0003702001949500.

[66] J. Bin *et al.*, “An efficient variable selection method based on variable permutation and model population analysis for multivariate calibration of NIR spectra,” *Chemom. Intell. Lab. Syst.*, vol. 158, pp. 1–13, 2016.

[67] E. Schulz, M. Speekenbrink, and A. Krause, “A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions,” *J. Math. Psychol.*, vol. 85, pp. 1–16, 2018, [Online]. Available: https://doi.org/10.1016/j.jmp.2018.03.001.

[68] Q. 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, Jul. 2018.

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

[70] H. Bin Qu, D. L. Ou, and Y. Y. Cheng, “Background correction in near-infrared spectra of plant extracts by orthogonal signal correction,” *J. Zhejiang Univ. Sci.*, vol. 6 B, no. 8, pp. 838–843, 2005.

[71] S. Wold, H. Antti, F. Lindgren, and J. Öhman, “Orthogonal signal correction of near-infrared spectra,” *Chemom. Intell. Lab. Syst.*, vol. 44, no. 1–2, pp. 175–185, 1998.

[72] A. Candolfi, R. De Maesschalck, D. Jouan-Rimbaud, P. A. Hailey, and D. L. Massart, "The influence of data pre-processing in the pattern recognition of excipients near-infrared spectra," *J. Pharm. Biomed. Anal.*, vol. 21, no. 1, pp. 115–132, 1999.

[73] L. Jiang, V. G. Fox, and L. T. Biegler, "Simulation and optimal design of multiple-bed pressure swing adsorption systems," *AIChE J.*, vol. 50, no. 11, pp. 2904–2917, 2004.

[74] C. T. Choi and H. Wen-Chung, "Incorporation of a valve equation into the simulation of a pressure swing adsorption process," *Chem. Eng. Sci.*, vol. 49, no. 1, pp. 75–84, 1994.

[75] C. tung Chou and W. C. Huang, "Simulation of a Four-Bed Pressure Swing Adsorption Process for Oxygen Enrichment," *Ind. Eng. Chem. Res.*, vol. 33, no. 5, pp. 1250–1258, 1994.

[76] D. Nikolic, A. Giovanoglou, M. C. Georgiadis, and E. S. Kikkinides, "Generic modeling framework for gas separations using multibed pressure swing adsorption processes," *Ind. Eng. Chem. Res.*, vol. 47, no. 9, pp. 3156–3169, 2008.

[77] F. Boukouvala, M. M. F. Hasan, and C. A. Floudas, "Global optimization of general constrained grey-box models: new method and its application to constrained PDEs for pressure swing adsorption," *J. Glob. Optim.*, vol. 67, no. 1–2, pp. 3–42, 2017.

[78] L. Jiang, L. T. Biegler, and V. G. Fox, "Simulation and optimization of pressure-swing adsorption systems for air separation," *AIChE J.*, vol. 49, no. 5, pp. 1140–1157, 2003.

[79] D. M. Hawkins and D. H. Olwell, *Cumulative sum charts and charting for quality improvement*. Springer, 1998.

[80] J. S. Hunter, "Exponentially Weigthed Moving Average," *J. Qual. Technol.*, vol. 18, pp. 203–210, 1986.

[81] R. Dunia and S. J. Qin, "Subspace approach to multidimensional fault identification and

reconstruction," *AIChE J.*, vol. 44, no. 8, pp. 1813–1831, 1998.

[82] Q. P. He, S. J. Qin, and J. Wang, "A new fault diagnosis method using fault directions in Fisher discriminant analysis," *AIChE J.*, vol. 51, no. 2, pp. 555–571, 2005.

[83] T. Kourti, P. Nomikos, and J. F. MacGregor, "Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS," *J. Process Control*, vol. 5, no. 4, pp. 277–284, 1995.

[84] P. Nomikos and J. F. MacGregor, "Multi-way partial least squares in monitoring batch processes," *Chemom. Intell. Lab. Syst.*, vol. 30, no. 1, pp. 97–108, 1995.

[85] S. J. Qin, S. Valle, and M. J. Piovoso, "On unifying multiblock analysis with application to decentralized process monitoring," *J. Chemom.*, vol. 15, no. 9, pp. 715–742, 2001.

[86] J. A. Westerhuis, T. Kourti, and J. F. Macgregor, "Comparing alternative approaches for multivariate statistical analysis of batch process data," *J. Chemom.*, vol. 13, no. 3–4, pp. 397–413, 1999.

[87] Y. Pan, C. K. Yoo, J. H. Lee, and I. B. Lee, "Process monitoring for continuous process with periodic characteristics," *J. Chemom.*, vol. 18, no. 2, pp. 69–75, 2004.

[88] R. Wang, T. F. Edgar, and M. Baldea, "A geometric framework for monitoring and fault detection for periodic processes," *AIChE J.*, vol. 63, no. 7, pp. 2719–2730, 2017.

[89] E. Arslan, D. Neogi, X. J. Li, and P. Misra, "Apparatus and methods to monitor and control cyclic process units in a steady plant environment," 2014.

[90] F. Wang, S. Tan, J. Peng, and Y. Chang, "Process monitoring based on mode identification for multi-mode process with transitions," *Chemom. Intell. Lab. Syst.*, vol. 110, no. 1, pp. 144–155, 2012, [Online]. Available: http://dx.doi.org/10.1016/j.chemolab.2011.10.013.

[91] S. Zhang, C. Zhao, and F. Gao, “Two-directional concurrent strategy of mode identification and sequential phase division for multimode and multiphase batch process monitoring with uneven lengths,” *Chem. Eng. Sci.*, vol. 178, pp. 104–117, 2018, [Online]. Available: https://doi.org/10.1016/j.ces.2017.12.025.

[92] B. Song, Y. Ma, and H. Shi, “Multimode process monitoring using improved dynamic neighborhood preserving embedding,” *Chemom. Intell. Lab. Syst.*, vol. 135, pp. 17–30, 2014, [Online]. Available: http://dx.doi.org/10.1016/j.chemolab.2014.03.013.

[93] W. C. Sang, E. B. Martin, A. J. Morris, and I. B. Lee, “Fault detection based on a maximum-likelihood principal component analysis (PCA) mixture,” *Ind. Eng. Chem. Res.*, vol. 44, no. 7, pp. 2316–2327, 2005.

[94] J. Lee, A. Kumar, J. Flores-Cerrillo, J. Wang, and Q. P. He, “Feature based fault detection for pressure swing adsorption processes,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 11301–11306, 2020, [Online]. Available: https://doi.org/10.1016/j.ifacol.2020.12.529.

[95] X. Xie and H. Shi, “Dynamic multimode process modeling and monitoring using adaptive gaussian mixture models,” *Ind. Eng. Chem. Res.*, vol. 51, no. 15, pp. 5497–5505, 2012.

[96] S. Zhang, F. Wang, S. Tan, S. Wang, and Y. Chang, “Novel Monitoring Strategy Combining the Advantages of the Multiple Modeling Strategy and Gaussian Mixture Model for Multimode Processes,” *Ind. Eng. Chem. Res.*, vol. 54, no. 47, pp. 11866–11880, 2015.

[97] Q. P. He and J. Wang, “Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes,” *IEEE Trans. Semicond. Manuf.*, vol. 20, no. 4, pp. 345–354, 2007.

[98] B. Song, S. Tan, H. Shi, and B. Zhao, “Fault detection and diagnosis via standardized k

nearest neighbor for multimode process," *J. Taiwan Inst. Chem. Eng.*, vol. 106, pp. 1–8, 2020, [Online]. Available: https://doi.org/10.1016/j.jtice.2019.09.017.

[99] S. Sircar, "Pressure swing adsorption," *Ind. Eng. Chem. Res.*, vol. 41, no. 6, pp. 1389–1392, 2002.

[100] F. Amiri, "Fault detection in pressure swing adsorption systems," 2019.

[101] B. M. Wise, N. B. Gallagher, S. W. Butler, D. D. White, and G. G. Barna, "A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process," *J. Chemom.*, vol. 13, no. 3–4, pp. 379–396, 1999.

[102] C. K. Yoo, J. M. Lee, P. A. Vanrolleghem, and I. B. Lee, "On-line monitoring of batch processes using multiway independent component analysis," *Chemom. Intell. Lab. Syst.*, vol. 71, no. 2, pp. 151–163, 2004.

[103] S. W. Choi, C. Lee, J. M. Lee, J. H. Park, and I. B. Lee, "Fault detection and identification of nonlinear processes based on kernel PCA," *Chemom. Intell. Lab. Syst.*, vol. 75, no. 1, pp. 55–67, 2005, doi: 10.1016/j.chemolab.2004.05.001.

[104] S. Kim, N. H. Kim, and J. H. Choi, "Prediction of remaining useful life by data augmentation technique based on dynamic time warping," *Mech. Syst. Signal Process.*, vol. 136, p. 106486, 2020, [Online]. Available: https://doi.org/10.1016/j.ymssp.2019.106486.

[105] A. Barré, F. Suard, M. Gérard, and D. Riu, "A Real-time Data-driven Method for Battery Health Prognostics in Electric Vehicle Use," *Phmsociety.Org*, pp. 1–8, 2014, [Online]. Available: http://www.phmsociety.org/sites/phmsociety.org/files/phm_submission/2014/phmce_14_0

42.pdf.

[106] L. Tao, C. Lu, and A. Noktehdan, “Similarity recognition of online data curves based on dynamic spatial time warping for the estimation of lithium-ion battery capacity,” *J. Power Sources*, vol. 293, pp. 751–759, 2015, [Online]. Available: http://dx.doi.org/10.1016/j.jpowsour.2015.05.120.

[107] E. Keogh and C. A. Ratanamahatana, “Exact indexing of dynamic time warping,” *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, 2005.

[108] A. Kassidas, J. F. MacGregor, and P. A. Taylor, “Synchronization of Batch Trajectories Using Dynamic Time Warping,” *AIChE J.*, vol. 44, no. 4, pp. 864–875, 1998.

[109] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemom. Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, Aug. 1987, Accessed: Oct. 24, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0169743987800849.

[110] S. J. Qin, “Statistical process monitoring: basics and beyond,” *J. Chemom.*, vol. 17, no. 8–9, pp. 480–502, 2003.

[111] J. E. Jackson and G. S. Mudholkar, “Control procedures for residuals associated with principal component analysis,” *Technometrics*, vol. 21, no. 3, pp. 341–349, 1979.

[112] K. A. Kosanovich, M. J. Piovoso, K. S. Dahl, J. F. MacGregor, and P. Nomikos, “Multi-way PCA applied to an industrial batch process,” *Proc. Am. Control Conf.*, vol. 2, pp. 1294–1298, 1994.

[113] P. Nomikos and J. F. MacGregor, “Monitoring batch processes using multiway principal component analysis,” *AIChE J.*, vol. 40, no. 8, pp. 1361–1375, 1994.

[114] J. M. Lee, C. K. Yoo, and I. B. Lee, “Enhanced process monitoring of fed-batch penicillin cultivation using time-varying and multivariate statistical analysis,” *J. Biotechnol.*, vol.

110, no. 2, pp. 119–136, 2004.

[115] Y. Zhang, T. Mao, Z. Huang, H. Gao, and D. Li, “A statistical quality monitoring method for plastic injection molding using machine built-in sensors,” *Int. J. Adv. Manuf. Technol.*, vol. 85, no. 9–12, pp. 2483–2494, 2016, doi: 10.1007/s00170-015-8013-2.

[116] J. Yang, Z. Lv, H. Shi, and S. Tan, “Performance monitoring method based on balanced partial least square and Statistics Pattern Analysis,” *ISA Trans.*, vol. 81, no. December 2017, pp. 121–131, 2018, [Online]. Available: https://doi.org/10.1016/j.isatra.2018.07.038.

[117] H. Zhang, X. Tian, X. Deng, and Y. Cao, “Multiphase batch process with transitions monitoring based on global preserving statistics slow feature analysis,” *Neurocomputing*, vol. 293, pp. 64–86, 2018.

[118] B. Zhou and X. Gu, “Multi-block statistics local kernel principal component analysis algorithm and its application in nonlinear process fault detection,” *Neurocomputing*, vol. 376, pp. 222–231, 2020.

[119] F. He and J. Xu, “A novel process monitoring and fault detection approach based on statistics locality preserving projections,” *J. Process Control*, vol. 37, pp. 46–57, 2016, [Online]. Available: http://dx.doi.org/10.1016/j.jprocont.2015.11.004.

[120] J. Lee and Q. P. He, “Understanding the effect of specialization on hospital performance through knowledge-guided machine learning,” *Comput. Chem. Eng.*, vol. 125, 2019.

[121] M. Hartman, A. Martin, P. McDonnell, and A. Catlin, “National health spending in 2007: Slower drug spending contributes to lowest rate of overall growth since 1998,” *Health Aff.*, vol. 28, no. 1, pp. 246–261, 2009, doi: 10.1377/hlthaff.28.1.246.

[122] Y. A. Ozcan, *Performance measurement using data envelopment analysis (DEA), in*

Health care benchmarking and performance evaluation. Springer, 2014.

[123] D. Delen, A. Oztekin, and L. Tomak, "An analytic approach to better understanding and management of coronary surgeries," *Decis. Support Syst.*, 2011, doi: 10.1016/j.dss.2011.11.004.

[124] G. R. Hobbs, "Data mining and healthcare informatics," *Am. J. Health Behav.*, vol. 25, no. 3, pp. 285–289, 2001.

[125] S. R. Eastaugh, "Hospital specialization and cost efficiency: benefits of trimming product lines," *J. Healthc. Manag.*, vol. 37, no. 2, p. 223, 1992.

[126] R. E. Herzlinger, "Specialization and its discontents: The pernicious impact of regulations against specialization and physician ownership on the US healthcare system," *Circulation*, vol. 109, no. 20, pp. 2376–2378, 2004, doi: 10.1161/01.CIR.0000130782.33860.E0.

[127] H. J. Jiang, B. Friedman, and J. W. Begun, "Factors associated with high-quality/low-cost hospital performance," *J. Health Care Finance*, vol. 32, no. 3, pp. 39–52, 2006.

[128] M. Ketokivi and M. Jokinen, "Strategy, uncertainty and the focused factory in international process manufacturing," *J. Oper. Manag.*, vol. 24, no. 3, pp. 250–270, 2006, doi: 10.1016/j.jom.2004.07.011.

[129] G. Wang, R. B. Chinnam, I. Dogan, Y. Jia, M. Houston, and J. Ockers, "Focused factories: A Bayesian framework for estimating non-product related investment," *Int. J. Prod. Res.*, vol. 53, no. 13, pp. 3917–3933, 2015, doi: 10.1080/00207543.2014.975373.

[130] K. N. McKay and V. C. S. Wiers, "Integrated decision support for planning, scheduling, and dispatching tasks in a focused factory," *Comput. Ind.*, vol. 50, no. 1, pp. 5–14, 2003, doi: 10.1016/S0166-3615(02)00146-X.

[131] C. Steiner, A. Elixhauser, and J. Schnaier, "The healthcare cost and utilization project: an

overview.," *Eff. Clin. Pract.*, vol. 5, no. 3, pp. 143–151, 2002.

[132] D. E. Farley and C. Hogan, "Case-mix specialization in the market for hospital services.," *Health Serv. Res.*, vol. 25, no. 5, p. 757, 1990.

[133] J. Lee and Q. Peter He, *Evaluating Hospital Performance Using Process Systems Engineering Tools*, vol. 44. 2018.

[134] S. N. Wood, *Generalized additive models: an introduction with R*. Chapman and Hall/CRC, 2017.

[135] P. Diehr, D. Yanez, A. Ash, M. Hornbrook, and D. Y. Lin, "Methods for analyzing health care utilization and costs," *Annu. Rev. Public Health*, vol. 20, no. 1, pp. 125–144, 1999.

[136] W. I. L. L. I. A. M. R. Hersh, "Healthcare data analytics," *Heal. informatics Pract. Guid. Healthc.*, 2014.

[137] B. Mihaylova, A. Briggs, A. O'hagan, and S. G. Thompson, "Review of statistical methods for analysing healthcare resources and costs," *Health Econ.*, vol. 20, no. 8, pp. 897–916, 2011.

[138] C. K. Reddy and C. C. Aggarwal, *Healthcare data analytics*. CRC Press, 2015.

[139] D. F. Alwin and R. M. Hauser, "The Decomposition of Effects in Path Analysis," *Am. Sociol. Rev.*, vol. 40, no. 1, pp. 37–47, 1975.

[140] J. R. Edwards and L. S. Lambert, "Methods for integrating moderation and mediation: A general analytical framework using moderated path analysis," *Psychol. Methods*, vol. 12, no. 1, pp. 1–22, 2007, doi: 10.1037/1082-989X.12.1.1.

[141] D. P. MacKinnon and A. J. Fairchild, "Current directions in mediation analysis," *Curr. Dir. Psychol. Sci.*, vol. 18, no. 1, pp. 16–20, 2009, doi: 10.1111/j.1467-8721.2009.01598.x.

[142] P. E. Shrout and N. Bolger, “Mediation in experimental and nonexperimental studies: New procedures and recommendations,” *Psychol. Methods*, vol. 7, no. 4, pp. 422–445, 2002, doi: 10.1037/1082-989X.7.4.422.

[143] R. Stine, “An Introduction to Bootstrap Methods: Examples and Ideas,” *Sociol. Methods Res.*, 1989, doi: 10.1177/0049124189018002003.

[144] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Wiley Series in Probability and Statistics : Introduction to Linear Regression Analysis (5)*. 2013.

[145] P. D. Allison, “Measures of Fit for Logistic Regression,” *SAS Glob. Forum 2014*, vol. 2, no. 1970, pp. 1–12, 2014.

[146] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.