Fault Diagnosis and Fault Severity Prediction Based on Computational Intelligence Techniques for Industrial Processes

Ruosen Qi

A Thesis Submitted for the degree of Doctor of Philosophy

School of Engineering

Newcastle University

United Kingdom

Feb 2024

Abstract

As industrial processes become more integrated and complex, the impact of faults rises, and process monitoring, and maintenance becomes more challenging. Though fault detection is crucial for mitigating risk, a comprehensive understanding of fault diagnosis and prognosis is essential for identifying the root causes and ensuring systemic safety. Previously, fault diagnosis based on reconstruction methods using principal component analysis (PCA) has been oriented towards sensor faults outside of the control loop, neglecting the intricacies of process fault that impact multiple variables due to intricate correlations between these variables. This thesis bridges this gap by offering enhanced statistical and machine learning methodologies for diagnosing and prognosing process faults.

Typically, a process fault will cause process measurements to move in a specific direction within the measurement space. The first principal component (loading vector) of a PCA model, when applied to known faulty data, can effectively capture the primary characteristics of the fault, thereby delineating the direction of deviation. This study enhances fault reconstruction methodologies by applying PCA to historical fault data, thereby establishing a fault direction matrix that encapsulates the multidimensional nature of process faults. It is imperative to note that this approach presupposes the availability of historical fault data for model training and is primarily effective for previously encountered faults. For emergent faults, without historical data, alternative methods or the incorporation of new data types would be necessary for effective diagnosis.

In this study, fault prognosis focuses on the long-range prediction of the reconstructed fault magnitudes. In this thesis, time series prediction models based on machine learning are developed, utilizing reconstructed fault magnitude data from historical process data and are used to predict future fault magnitudes. The thesis presents using extreme learning machine (ELM) models and long short-term memory (LSTM) networks to build fault magnitude prediction models, as well as autoregressive (AR) models as a baseline for comparison. ELM models show improved prediction results compared to AR models. Nevertheless, during the training phase, the adjustment of the ELM network's parameters is confined to the output weights. The hidden layer weights and biases, initially set at random, remain unchanged. This design choice, made by the developer, aims to streamline the

training process, but it may lead to less consistent predictive performance across different datasets. In contrast, LSTM networks, with their memory units, are better suited for modelling dynamic relationships and handling long-term dependencies. To enhance the reliability and accuracy of long-range prediction, this study employs an LSTM network model for multi-step prediction of fault magnitude. The applicability of this method is illustrated through a simulated continuous stirred tank reactor (CSTR) process, suggesting that such model could be beneficial in industries such as chemical production, pharmaceutical manufacturing, and energy generation. This model could be integrated into monitoring systems to optimize predictive maintenance, enabling early fault detection, and preserving the integrity of production processes.

While this thesis primarily focuses on the individual contributions and applications of PCA-based methods, ELM, and LSTM models in process fault prediction, it acknowledges the potential of their integrated use. The combination of these methods, although not explored in depth within this work, poses a promising avenue for future research. Such a hybrid approach could synergistically utilize the early detection strengths of PCA-based methods, the rapid prediction capabilities of ELM, and the long-term dependency learning of LSTM models. This integrated model could potentially lead to more advanced, robust, and comprehensive fault prediction systems, forming an exciting and valuable direction for future research endeavours.

Acknowledgements

As I conclude this monumental journey of my PhD, my gratitude extends to my supervisor, Dr. Jie Zhang, whose steadfast guidance and wisdom have been instrumental over the past five years. His invaluable counsel permeated not just my academic endeavours but enriched the ambiance of our research environment as well. With each stride in my research, whether it culminated in a published paper or contributed to this final thesis, Dr. Zhang's meticulous feedback and insightful suggestions have been a driving force. His expansive knowledge, unwavering commitment, meticulous academic approach, and his awe-inspiring passion for science have deeply inspired me. These qualities have etched an indelible mark on me and will continue to motivate my journey in the years to come.

I extend a warm thank you to my friends, both in the UK and China, for standing by me during challenging times and for providing comfort during moments of self-doubt. Your continued support helped restore my confidence, encouraging me to navigate through obstacles and persist with my endeavours.

A special note of gratitude to my girlfriend, Genevieve Moat - our paths crossed at the perfect time, offering a companionship that has been nothing short of a blessing. Your patience, understanding, and shared burden of my worries have granted me the serenity to focus on my studies and nurture my personal growth alongside you.

Lastly, my deepest appreciation is for my parents, whose unwavering support, love, and consistent encouragement have been my pillar of strength throughout my studies. Their faith in me has been the bedrock of my journey, and for that, I am forever grateful.

Table of Contents

List of Figures

List of Tables

Nomenclature

C_P	heat capacity of input fluid ($cal/(gK)$)
C_{AA}	concentration of reactant ($kmol/m^3$)
C_{AS}	concentration of solvent ($kmol/m^3$)
F_α	the function of the F distribution
F_C	cooling water input flow (m^3/min)
F_A	reactant A input flow (m^3/min)
F_S	solvent input flow (m^3/min)
k	the number of principal components
K_C	proportional gain
m	the number of observed variables
n	the number of samples
N	the length of the time series
Ns	the number of input samples of CNN
I	the number of input layers of ELM
l	the number of hidden layers of ELM
O	the number of output layers of ELM
p_i	loading vector
P	loading matrix
$\tilde{P}$	projection of P onto the residual subspace
q	the order of autoregressive model
S	covariance matrix
t_i	score vector
T	score matrix
$\tilde{T}$	projection of T onto the residual subspace

T_i	integral gain
T_0	input temperature (K)
T_{cin}	cooling water input temperature (K)
X^o	original normal operation process data matrix
X	Standardised original normal operation process data matrix
$\hat{X}$	projection of X onto the principal component subspace
$\tilde{X}$	projection of X onto the residual subspace
z_α	standard normal distribution
w_t	white noise
w	weights between the input layer to the hidden layer
v	weights between the hidden layer to the output layer
b	hidden layer bias
f	the fault magnitude
F	CSTR input flow
P_k	loading matrix extracted for the corresponding fault
V	reactor volume (m^3)
$-\Delta H_{rxn}$	heat of reaction ($cal/(kmol)$)

Greek letters

λ_i	eigenvalue of matrix X
σ_i	singular value of matrix X
Λ	eigenvalue matrix
δ_T^2	control limits of T^2 statistic
δ_{SPE}^2	control limits of SPE statistic
α	the significance levels
φ_i	the regression coefficient

θ_j	the moving average coefficient
Ξ_i	fault direction matrix
$\widetilde{\Xi}_i$	projection of Ξ_i onto the residual subspace
ρ_k	autocorrelation coefficient between element of the time series
$\hat{\varphi}$	autoregressive model coefficient matrix
ρ	density of input fluid (g/m^3)
ρ_c	density of cooling water (g/m^3)

Mathematical operators

Σ	summation
ln	natural logarithm function

Abbreviations

AIC	Akaike Information Criterion
AR	autoregression
ARMA	autoregressive moving average
ARIMA	autoregressive integrated moving average
BIC	Bayesian Information Criterion
BP	back propagation
BPTT	back propagation through time
CPV	cumulative percent variance
CNN	convolutional neural network
CSTR	continuous stirred tank reactor
DBN	deep belief network
DC	direct current
EKF	extended Kalman filter

ELM	extreme learning machine
FNN	feedforward neural network
GRU	gated recurrent unit
HMM	hidden Markov model
ICA	independent component analysis
IUPF	improved unscented particle filter
KICA	kernel independent component analysis
KPCA	kernel principal component analysis
KPLS	kernel partial least squares
LSTM	long short-term memory
MA	moving average
MAE	mean absolute error
MAPE	mean absolute percentage error
MSPM	multivariate statistical process monitoring
PCA	principal component analysis
PCs	principal components
PCS	principal component subspace
PE	Paris-Erdogan
PLS	partial least squares
RBC	reconstruction-based contribution
RBF	radial basis function
RMSE	root mean square error
RNN	recurrent neural network
RS	residual subspace
RUL	remaining useful life

SGD	stochastic gradient descent
SLFN	single hidden layer forward neural
SPE	squared prediction error
SVM	support vector machine
SVD	singular value decomposition
TE	Tennessee Eastman

Chapter 1 Introduction

1.1 Background

With the integration of advanced computer technology in modern industrial processes, the complexity of production systems in industrial plants has increased significantly. These systems have evolved from simpler single loop control systems to more sophisticated multivariable control systems. Many industrial processes now feature continuous, large-scale production with complex reaction mechanisms. These processes are characterized by nonlinear behaviour and time-varying dynamics, often exhibiting strong correlations between numerous inputs and outputs. Additionally, uncertainties in feedstock and operational parameters add further complexity to monitoring these industrial production processes.

Modern large-scale production system typically composed numerous interconnected and interdependent process units. This integration enhances production and energy efficiency, yielding considerable economic benefits. However, it also means that a fault in one component can have a cascading effect, potentially impacting the entire production process. Minor faults, if not addressed promptly, can disrupt normal operations, or even necessitate a complete shutdown. While less severe faults might merely degrade performance, affecting the quality and quantity of output, serious faults can lead to complete system shutdowns, equipment damage, and in extreme cases, casualties. For examples, on April 22, 1992, a gas explosion took place in Guadalajara Mexico, due to the pipeline corrosion leading to fuel leaks to the sewer and a short circuit of the wire detonated the fuel (Bilbao and Alcerreca, 1994). More than 200 people were killed and 1470 injured. In 2004, an explosion occurred at an LNG plant in Skikda, 500 kilometres east of the Algerian capital, because of the pressure in the tank reached the limit resulting in a larger amount of steam leakage and leading a secondary explosion, which caused death of 27 people and 74 injured (Reddy and Yarrakula, 2016). On November 22, 2013, the rupture of the oil pipeline of Sinopec Pipeline Company in Qingdao, Shandong Province, China, caused the leakage of crude oil. During the repairs, a fire broke out and the accident caused 62 deaths, 136 injuries and direct economic losses of 750 million yuan ('TCE: Asia: Sinopec publicly reprimanded over pipeline blast,' 2014). On November 30, 2017, an accident occurred in the heat exchanger repair operation of PetroChina Urumqi Petrochemical Company's oil refinery, resulting in five deaths and 16 injuries (Pelayo *et al.*, 2017). Therefore, in

order to avoid the occurrence of serious accidents, reduce general accidents, prevent or eliminate the occurrence of faults, and enable modern industrial processes to achieve a higher level of safety, reliability and efficiency, industrial process monitoring has become increasingly important (Abbasi *et al.*, 2019). With the widespread use of distributed control systems, various intelligent instruments and control devices have been deployed in modern industrial processes and large amounts of process data are collected and stored. This data, encapsulating vital information about the process's operating status, is frequently underutilized. Therefore, a key area of interest is exploring how this data can be effectively leveraged for process monitoring. This topic has garnered significant attention in recent years, becoming a prominent research focus in the fields of automation and control within the process industry.

1.2 Process Monitoring Research

Generally, any unacceptable abnormal behaviour that causes a system or process to deviate from its normal operating conditions or state is considered a process fault (Isermann, 1997).

In the process industry, faults can be classified according to their source, such as sensor faults that affect process measurements, actuator faults that affect the manipulated variables, faults caused by malfunctions, and faults in the system components of process equipment (Chiang *et al.*, 2001). These faults can occur suddenly, such as the abrupt fault of process equipment, or they can occur slowly over time, such as faults caused by gradual wear and tear of equipment or scaling of pipework.

Process monitoring is defined as the real-time and effective monitoring of possible faults or abnormal events in the process to ensure that the system meets the specified performance requirements during operation and that the correct diagnosis is made in a timely manner. Information derived from process monitoring allows process maintenance personnel to accurately and promptly determine if an abnormal condition is occurring in the process and to quickly take appropriate action to eliminate the cause of the abnormal condition. Process monitoring reduces the time that faults are present throughout the process, thereby increasing the reliability and safety of the equipment and minimising the damage caused by the faults. As the industrial age continues to evolve, process monitoring can be divided into three areas: fault detection, fault diagnosis and fault prognosis (Reis and Gins, 2017a).

Fault detection is the process of identifying and detecting abnormal behaviour or faults in a system or process. It is typically the first step in a fault diagnosis and prognosis system, as it alerts operators or maintenance personnel that something is wrong and requires further investigation. The process of identifying the specific cause of a detected failure or abnormal behaviour is known as fault diagnosis. Typically, this involves analysing sensor data and other system information to find the main cause of the problem and to isolate the impacted component or system. The objective of fault diagnosis is to promptly and precisely identify the problem so that the necessary maintenance or repairs can be performed to minimise system downtime and maintain system performance. Fault prognosis is a proactive method for predicting system faults or failures before they occur or before they become significant. The aim of this approach is to improve system reliability and reduce downtime through preventative maintenance and repair. To achieve this goal, sensors, data analysis techniques and machine learning algorithms are used to monitor the performance of the system and identify any trend or anomalies that may indicate the possibility of a fault.

Process monitoring can be traced back to the 1970s when the American academic Beard laid the foundations of process monitoring theory by proposing the idea of using analytical redundancy instead of physical redundancy for fault monitoring and diagnosis (Das *et al.,* 2012). With the development of computer technology, artificial intelligence and expert systems, methods and techniques designed for process monitoring have spanned several areas such as mathematical analysis, signal processing, statistics, and artificial intelligence. Process fault monitoring methods are often classified as model-based, knowledge-based and data-based methods (Venkatasubramanian *et al.,* 2003a; Venkatasubramanian *et al.*, 2003b; Venkatasubramanian *et al.*, 2003c). The model-based approach relies on a model of the monitored process usually developed from the internal working of the production process or the transfer mechanism of the material flow. Residuals from the model are calculated by comparing the measured information of the monitored process with the estimated system information expressed by the model, which then serves to evaluated and diagnosis of faults. With a precise mathematical model, the model-based method for process monitoring is highly accurate, but such a model is usually difficult to obtain. Typical model-based methods include the parameter estimation method, the state estimation method, and the parity relations method.

Knowledge-based methods usually rely on process operation experience and process knowledge and tend to provide more intuitive monitoring results, but building a process knowledge base is always a time-consuming and difficult operation requiring long-term expertise and experience, represented by methods such as fault trees, expert systems, fuzzy reasoning, etc.

Modern industrial production processes are complex, and it is difficult to obtain accurate mechanistic models. It is also difficult to organise and establish a systematic expert knowledge base. Therefore model-based and knowledge-based methods are only applicable to certain processes and mechanistically explicit unit devices. Modern industry has been equipped with a large number of sensors and other instrumentation equipment, resulting in a huge amount of process history data. Along with the significant development of multivariate statistical analysis, machine learning, deep learning and other process data analysis techniques, data-based methods have become the mainstream method and research hotspot for process monitoring in the process industry in the past 20 years, which do not require the establishment of accurate mechanistic models, nor do they rely heavily on expert knowledge. Thus data-driven monitoring approaches have the advantages of easy implementation, simplicity, and good generality.

1.3 Motivation

From the above discussion, the data-based approaches are more suitable for the monitoring of modern industrial processes.

In the field of process monitoring, earlier data-based approaches have used multivariate statistics, signal processing and pattern recognition techniques to extract and model data derived from actual process operations. This information was then used to evaluate the current state of the process and implement process monitoring. In recent years, new research results have emerged from machine learning technology, represented by deep learning, which has provided strong technical support for data-based methods, making data-based process monitoring increasingly accurate. Based on the models used, the various types of data-driven process monitoring methods can be classified as multivariate statistical analysis-based, signal processing-based and deep learning-based methods. All three types of methods are extensively used in real-world industrial process monitoring, fault diagnosis and prognosis.

Despite the widespread use of data-based process monitoring techniques, there are several issues that need to be addressed in practice. This research has highlighted the following important issues:

a) Fault complexity. Most of the research on fault diagnosis based on fault reconstruction approach in chemical processes has concentrated on sensor faults on non-control loops, which do not propagate to other variables. This type of faults is typically caused by the failure of the sensors in the measured variables and can be characterised by the contribution of the single variable affected, as it is the only variable that breaks its correlation pattern with all other variables in the model. However, relatively less research has been conducted on the reconstruction of process faults, most of which arise from fundamental changes in the process and imply deviations from operating conditions, such as contamination by impurities in the concentration of reactants, and accumulation of debris and blockages in hydraulic cylinders, etc. Others arise in control loops, where sensor reading are used for feedback control, and the sensor faults can lead to inefficient or inaccurate control. These faults can have minor to extremely serious consequences and have an impact on multiple process variables, so the study of process faults is essential.
b) Incipient faults. Three types of faults can be distinguished according to their time dependence: abrupt faults, incipient faults, and intermittent faults. Incipient faults usually develop very slowly and are difficult to detect and diagnose. Early detection and prognosis of incipient faults are important and need to be further investigated. Therefore, incipient faults are a greater concern and research challenge in this thesis, and more accurate and sensitive approaches will be developed.
c) Fault prognosis and trend analysis. From the existing research results, the research work in the field of data-based process monitoring is mainly focused on the evaluation of the system condition and fault diagnosis, the concern is the "current" operating state of the system, i.e., whether a fault has occurred, the location of the fault, etc. However, in real industrial processes, many production processes are subject to demanding operating conditions. It is no longer sufficient to provide fault detection and diagnosis in the event of a fault, as the fault may be found in a dangerous operating

area, where it is difficult for the operator to restore the system to normal conditions, or where the fault has a serious impact on the economic efficiency of production. The prediction of fault propagation and trends is therefore of great benefit to the optimisation of industrial maintenance strategies.

1.4 Aims and Objectives

The main aim of this research is to enhance methodologies for fault diagnosis and fault severity prediction in industrial processes, with a particular focus on data-based methods. By integrating advanced statistical and machine learning techniques within a data-driven framework, this research seeks to offer more precise and reliable fault detection and prediction solutions. Emphasizing a data-driven approach, the objective is to contribute to the enhancement of safety and efficiency in industrial operations through improved fault analysis techniques. In order to achieve this aim, the following objectives are set:

a) Review existing techniques for fault diagnosis and prognosis in industrial processes.
b) Improve and optimise process fault detection and fault reconstruction-based fault diagnosis methods. As most of the reported works on fault reconstruction focus on sensor faults on non-control loops, a method for formulating the fault direction matrix for process faults is investigated.
c) Develop a new method for predicting process fault magnitudes by using ELM and LSTM network to provide long range fault magnitude prediction, based on reconstructed fault magnitudes.
d) Validate of the proposed method through simulated Continuous Stirred Tank Reactor.

1.5 Contributions

This thesis contributes to the field of process monitoring and fault diagnosis in several significant ways, particularly focusing on the complexities of process faults in industrial settings. The specific technical contributions reported in this thesis are as follows:

a) Enhancement of Fault Reconstruction Methods for Process Faults: This research contributes to the field by enhancing existing fault reconstruction methods to better accommodate the complexity of process faults in

industrial systems. By adapting the fault direction matrix to reflect the multi-dimensional nature of process disturbances more accurately, the study provides a refined diagnostic method that improves upon the precision of fault characterization. This enhancement allows for a more nuanced understanding of process faults, which is critical for the effective diagnosis of such faults in industrial applications.

b) Advancement in Machine Learning Methods for Fault Prognosis: In this thesis, distinct machine learning models, namely ELM and LSTM networks, are developed to predict the magnitude of process faults in industrial processes. These models are individually tailored to capitalize on historical fault data, with ELM models offering quick prediction capabilities and LSTM networks providing a deep analysis of time-series data to capture long-term dependencies. The study presents a critical evaluation of each model's predictive performance, offering insights into their applicability in the chemical process when compared to traditional AR models and highlighting the superiority of neural network approach in handling complex data structures and time dependencies.

c) Validation of Methods through Simulation of an Industrial Process: The thesis validates the developed fault diagnosis and prognosis methods using a simulated CSTR process. This simulation not only serves as a proof of concept but also illustrates the potential for these methods to be applied in actual industrial settings. The findings demonstrate how the models might be integrated within existing process monitoring frameworks, suggesting improvements for process fault reconstruction and fault magnitude prediction. This contribution underscores the practical relevance of the research and its applicability to enhancing operational safety and efficiency in industrial processes.

In conclusion, this thesis presents several methodological advancements in the domain of process monitoring and fault diagnosis. By developing new approaches for the diagnosis and long-range prognosis of process faults, this work seeks to supplement existing methodologies, addressing specific limitations they possess. While the full extent of the impact on industrial safety, efficiency, and economic outcomes will be contingent on practical application and further empirical validation, the contributions of this thesis provide a foundation for such

improvements. The expectation is that, with further development and integration into industrial practices, these contributions could offer valuable tools to enhance the management of industrial processes.

1.6 Structure of the Thesis

Chapter 1 provides an overview of the PhD topic, underscoring the purpose and significance of the research. It offers an introductory discourse on fault diagnosis and prognosis techniques, laying out some of the prominent issues within the current research landscape, leading to the focus and main content of this thesis.

Chapter 2 in-depth review of various methods used in fault detection and diagnosis in industrial processes. It covers a range of techniques, including model-based, knowledge-based, and data-based methods, and explores their applications and principles. The chapter also delves into fault prognosis methods, discussing model-based and data-driven approaches. This comprehensive review sets the stage for the thesis by highlighting the evolution of these methods.

Chapter 3 focuses on technical preliminaries and detailed descriptions of the algorithms used in the thesis. It begins with an overview of multivariate statistical process monitoring, including principal component analysis and its application in fault detection. The chapter then explores various data-driven modelling techniques for fault prediction, such as AR models, ELM, and LSTM neural networks. This chapter lays the foundational knowledge and methodological framework essential for the research conducted in subsequent chapters.

Chapter 4 elaborates on the enhancement of the reconstruction-based process fault diagnosis method. It discusses the development and application of a fault direction matrix for process faults, enhancing the precision and effectiveness of fault detection and diagnosis. The chapter validates these methods through simulation, demonstrating their practicality in industrial scenarios.

Chapter 5 focuses on predicting fault magnitudes using AR models and ELM. The chapter details the development of these models, discusses their application in fault magnitude prediction based on reconstructed fault data, and compares their performance, showcasing their effectiveness in fault prognosis.

Chapter 6 presents the long-range fault prognosis models using LSTM and Gated Recurrent Unit (GRU) networks. The chapter explores the optimization of these

models through various experiments and compares their predictive performances, emphasizing their potential in industrial process monitoring.

Finally, in Chapter 7, the main conclusions are summarised, innovative points are extracted. It also outlines potential directions for future research, emphasizing the practical implications and the potential integration of these methodologies in industrial process fault diagnosis and prognosis.

1.7 Publications

Journal paper:

Ruosen Qi, Jie Zhang and Katy Spencer (2023) 'A Review on Data-Driven Condition Monitoring of Industrial Equipment'. Algorithms 16(1), 9.

Conference papers:

Ruosen Qi and Jie Zhang (2019) 'Process Fault Detection and Reconstruction by Principal Component Analysis'. 24th International Conference on Methods and Models in Automation and Robotics (MMAR 2019), Międzyzdroje, Poland, August 26-29, 2019, 594-599.

Ruosen Qi and Jie Zhang (2020) 'Data-driven fault diagnosis and prognosis for process faults using principal component analysis and extreme learning machine'. 18th IEEE International Conference on Industrial Informatics (INDIN 2020), Warwick, United Kingdom, July 20-23, 2020, 755-780.

Ruosen Qi and Jie Zhang (2020) 'Fault Magnitude Prognosis in Chemical Process Based on Long Short-Term Memory Network'. Proceedings of the 2020 4th International Symposium on Computer Science and Intelligent Control (ISCSIC 2020), November 17-19, 2020, Newcastle upon Tyne, UK, 2020, 1-6, DOI: 10.1145/3440084.3441212.

Chapter 2 Literature Review

Before automation technology was widely used in industrial processes, human operators were primarily responsible for regulating process plants. as the scale and complexity of modern industry has expanded, it has become increasingly challenging to rely solely on human supervision to deal with abnormal events and emergencies. Whereas the gradual spread of computer control technology in the process industries has allowed many production processes to be automated, marking a significant milestone in industrial development.

Process monitoring is the monitoring or supervision of an industrial process to ascertain that the operational state of the system aligns the specified performance requirements. It also enables effective detection and diagnosis of potential faults or abnormal events in real-time. The process monitoring encompasses the following tasks:

1) Gaining insight into the current process operating status.
2) Performing Real-time analysis of the operating status and interaction with process maintenance personnel.
3) Detecting abnormal process operating conditions promptly and providing timely information regarding the cause of the condition.
4) Swift and effective elimination of abnormal process conditions and maintenance of process stability.
5) Evaluating the impacts of faults on product quality and minimising product quality degradation through adjustments to process operating conditions.

The key components of process monitoring include process modelling, fault detection, fault diagnosis, fault identification, fault isolation and process recovery, as shown in Figure 2.1. The process begins with determining whether a fault has occurred, followed by diagnosing the location and root cause of the fault, and finally minimising the effects of the fault based on the diagnostic information.

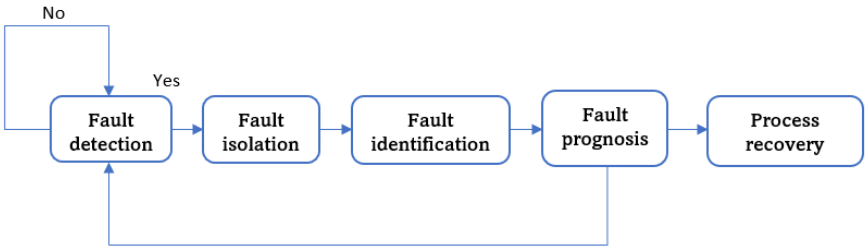

Figure 2.1 A diagram of process monitoring

An industrial plant primarily comprises sensors, actuators, various process units and their connections through pipes and valves. Given the prolonged exposure of most of these components to external conditions, a variety of factors can cause faults, such as component deterioration, defects of process equipment such as pumps and valves, non-compliant operating conditions, etc. Faults are different from deviations of parameters triggered by external environmental disturbances, as the latter can often be corrected promptly by an automatic control system, while the former typically requires manual intervention to resolve. In an industrial process, undetected faults may reduce the system's performance or cause certain functions to malfunction, or even fail. In this context, the definitions of failure and malfunction are not to be confused - a malfunction is defined as an intermittent irregularity in the operation of a system, whereas a failure is a permanent interruption in the system's ability to perform its tasks under defined operating conditions (Hamadache *et al.*, 2019). A fault in an industrial process is defined as an unacceptable deviation of at least one of the process characteristics (or variables) from acceptable behaviour, potentially leading to system performance deterioration (Severson *et al.*, 2016). The underlying cause of a fault can be several different reasons with the common ones including sensor faults, actuator faults and system faults.

In the operation of industrial processes, sensors and actuators play a pivotal role. Sensors detect, measure and transmit signals of any changes in the process and feedback to the control system, while actuator are used to receive control signals and execute control actions. A sensor fault refers to a significant deviation between the measured and actual values, resulting in inaccurate measurements of process variables, such as bias error, drift error, gain error, etc. An actuator fault refers to a significant discrepancy between the input command of the actuator and its actual output, such as a stuck valve.

System faults typically occur within the process itself, usually triggered by external changes or abrupt operating condition changes, resulting in sudden parameter changes within the system. As a result of these changes, the interrelationships and information interactions between different variables are affected. Such faults can be caused by conditions such as fouling inside a distillation column or heat exchanger, changes in feed concentration, pipe ruptures or leaks. Generally, these faults manifest as gradual changes in the process variables.

Depending on the time variation, faults are often categorized as abrupt fault, incipient fault, and intermittent fault (Zhang *et al.*, 2002). Incipient faults are slow changes in process characteristics and parameter drift. These faults generally do not significantly damage the process in the short term. However, their cumulative effect can eventually have a major impact on the process. In large plants, such as power generation and chemical plants, incipient faults may be precursors to catastrophic accidents. An abrupt fault is a sudden change in the characteristics of a component, or a step change in certain parameters, which can significantly impact system performance. An abrupt fault typically takes the form of a jump fault, e.g., a sensor output of a constant value independent of the measured parameter or an actuator jamming, etc. Such faults can have a catastrophic effect on the control system. Intermittent faults represent deviations or variations that appear and disappear repeatedly over time.

The procedure of process monitoring can be summarised as follows:

1) Process modelling: Developing a mathematical model of the process being monitored based on the a priori information, process operation data, and input-output relationships of a process.
2) Fault detection: Using models to assess the operational state of the process and determine whether the process is in a normal or abnormal operating condition. Fault detection forms the foundation of process monitoring technology.
3) Fault isolation: After a fault has been detected, this step isolates the component or subsystem where the root variable of the fault is located.
4) Fault diagnosis and identification: Determining the cause of the detected fault in the process, determine when the fault occurred, estimate its size, and identify its type.
5) Process recovery: Remedying the fault through various measures to restore the process to normal operation and devising countermeasures to manage potential adverse effects of the fault.
6) Fault prognosis: After fault detection and diagnosis, fault prognosis is the next task of process monitoring. It consists of two main levels: health state prediction and remaining useful life (RUL) estimation. The health state forecast determines the current state of the component or system during its health degradation process, such as normal state, degraded performance

state or fault state, etc. RUL estimation determines the RUL of a component or system based on its current state, historical state, and other information, using a suitable prognosis model.

2.1 Overview of Various Methods Used in Fault Detection and Diagnosis

Traditional classification concepts categorise fault detection and diagnostic techniques into three primary categories: model-based methods, knowledge-based methods, and signal-processing-based methods (Frank, 1990). However, with the rapid advancements in technology and the rise of big data, these classifications have been compelled to evolve. Specifically, within the context of process monitoring, the introduction of data-driven methods, which utilise substantial quantities of process data for decision-making, has prompted significant changes in the classification of fault detection and diagnosis techniques.

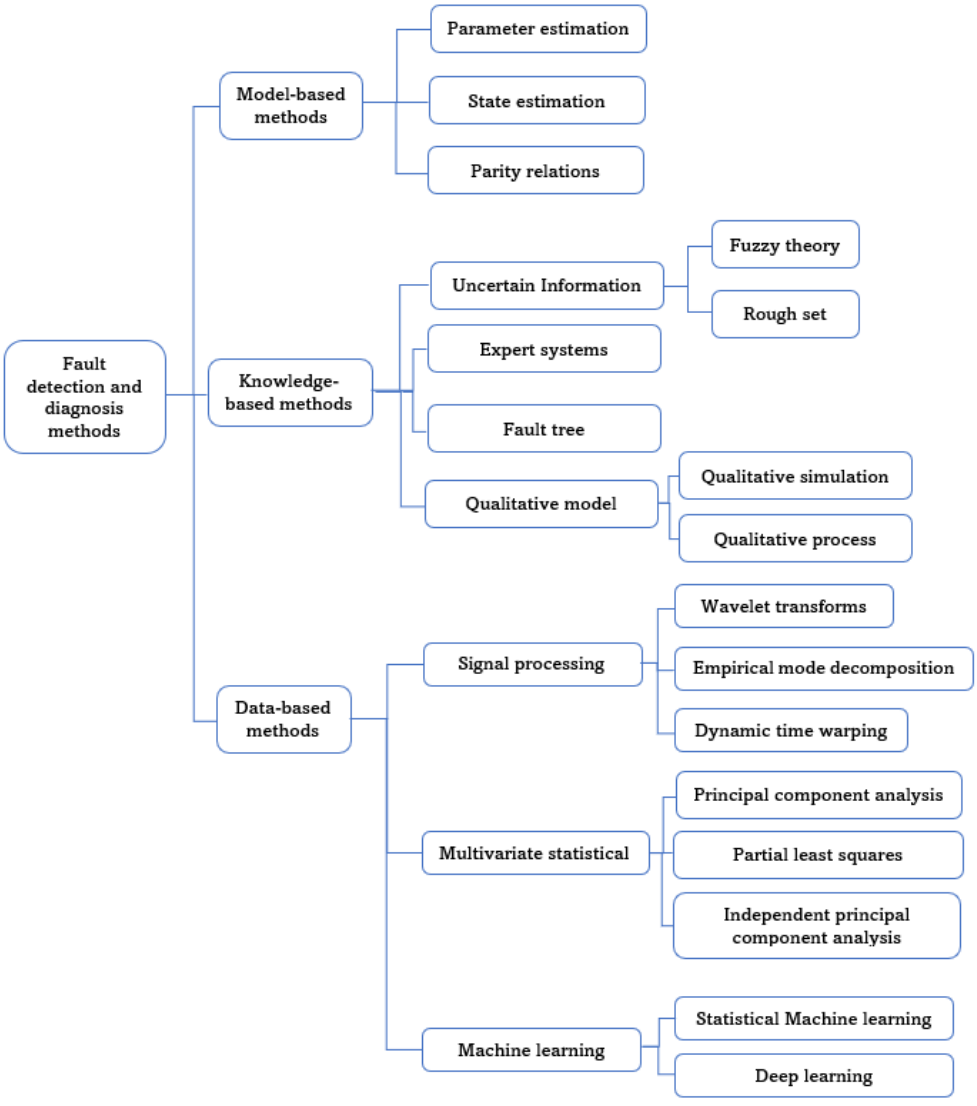

Figure 2.2 Classification of fault detection and diagnosis methods

Consequently, researchers and industry professionals now typically categorise fault diagnosis and prognosis techniques into three distinct groups: model-based methods, knowledge-based methods, and data-driven methods, as illustrated in Figure 2.2. This categorization encapsulates the progression within the field, where traditional model-based techniques, dependent on robust mathematical formulations of systems, and knowledge-based techniques, rooted in expert understanding and rule-based logic, are being increasingly complemented by data-driven methods. These contemporary methods utilize the wealth of data available from industrial processes, applying advanced algorithms to extract insights. The inclusion of data-driven methods within this classification signifies the field's shift towards leveraging empirical data, thereby integrating analytical and experiential knowledge bases with powerful computational analysis to push the boundaries of fault detection and diagnosis.

2.1.1 Model-based Methods

Developing earlier than other methods, the model-based methods primarily involve analysing the internal mechanisms and external influencing factors of the process objects. The aim is to establish accurate input-output models, followed by the construction of a residual signal from the actual process data and the predicted output of the mathematical model, which is then used for fault detection and diagnosis. Methods for fault detection and diagnosis mainly include parameter estimation methods, state estimation methods and parity relations methods (Zhang *et al.*, 2000). The parameter estimation method integrates modelling theory with parameter identification to describe faults by comparing significant jumps in the identified parameters with the fault threshold (Isermann, 1991). The state estimation method uses the state space model of the system and the observer to reconstruct the state of a controlled process. By comparing these with actual measurements, this method diagnoses process faults by analysing the sequence of residuals (Ge and Fang, 1988; Zarei and Shokri, 2014). The parity relation method transforms the system equations into the frequency domain, checks the equivalence of the mathematical model of the system through the actual measurements of system input and output signals to obtain the residual signals (Kinnaert, 1993).

2.1.2 Knowledge-based Methods

Knowledge-based methods depend on the analysis of priori information about the system operation mechanism, fault characteristics, and the causal relationship

between fault behaviour and its causes. They use logical reasoning to detect and isolate faults. The main methods used include uncertain information theory, qualitative model, expert systems, and graph theory methods. The uncertain information theory method uses fuzzy theory to construct membership functions that represent the relationship between changes in process parameters and the cause of faults to achieve fault detection and diagnosis (Coito *et al.*, 2005). It may also use rough set theory to perform attribute reduction and remove unimportant process variables to achieve simplification of fault feature information and reduction of diagnostic rules (Zarei and Shokri, 2014). However, this type of method relies heavily on prior system knowledge to determine affiliation function or attribute reduction. The qualitative model-based method uses qualitative knowledge of equipment to build models that describe the structure and function of the process. Thus, the predicted qualitative behaviour of the system can be derived, and fault detection and diagnosis can be achieved by comparing it with the actual system behaviour (Zhou *et al*., 2017). This method is only suitable for describing known system faults and cannot accurately diagnose unknown faults. The expert systems method uses the practical expert experience to create a knowledge base and simulates the reasoning and decision-making process for diagnosis (Zinn *et al.*, 2020). However, the method requires a comprehensive knowledge base to build a expert system of the considered process. The graph theory method uses cause–effect relationships between process variables and faults to detect and diagnose faults (Kościelny *et al*., 2022). Although it is straightforward to build for simple systems, the method becomes complex and often ambiguous when used for more complex systems due to the conflicting variable effects.

Common to these qualitative fault diagnosis methods is their ability to establish a qualitative model based on subjective qualitative knowledge by delving deeper into process information such as the internal structure of the process industry, the process mechanism, and the relationships between variables in the process. Evidently, the knowledge-based approach does not require a precise mathematical model and can determine fault propagation through inference and deduction. However, the efficacy of this technique depends on the expert's understanding of the production process and extensive production expertise. As the complexity of the process industry has increased, accumulating expert knowledge has become more challenging, making the knowledge-based approach less adaptable.

2.1.3 Data-based Methods

Data-based methods serve as effective alternative to model-based and knowledge-based methods, particularly when the process to be monitored lacks a precise mathematical- mechanistic model or an adequate expert knowledge base. Modern industrial systems demonstrate high levels of automation and instrumentation, typically accumulating large volumes of process data. Earlier data-based approaches used multivariate statistics, signal processing and pattern recognition to extract and model useful information from the historical process operation data. These techniques are then used to evaluate the current operational state of the process, consequently implementing detection and diagnosis based on the model. With the simplicity and versatility of their models, data-based methods have become one of the most popular research areas in fault detection and diagnosis. In recent years, data mining techniques represented by machine learning and deep learning, have provided robust technical support to data-based approaches, enhancing their performance.

Data-based fault detection and diagnosis methods can be categorised into signal processing-based methods, multivariate statistics-based methods, and machine learning-based methods.

1. Signal Processing Methods

Fault signal analysis and processing is a quantitative fault diagnosis method, which has been quite mature in practical engineering applications. It mainly uses the wavelet transform method (Ravikumar *et al*., 2022), empirical mode decomposition method (Du and Du, 2018) and dynamic time warping theory (Sun *et al*., 2020) to process and transform the monitored signals of the system to obtain the effective information reflecting the fault, i.e., the fault signatures, and uses the deviation degree to realize the fault diagnosis by comparing the actual signals with these fault signatures quantitatively. By comparing the actual signals with these fault signatures quantitatively, the degree of deviation is used for fault diagnosis.

Different faults lead to different characteristics in the spectrum of the measured signal and can thus be diagnosed through spectral analysis of the signal spectrum. In (Drif *et al.*, 2002), the energy spectrum of the stator of an induction motor was analysed for the fault diagnosis of squirrel-cage rotors in motors. The spectral analysis of smooth signals is frequently used in the diagnosing of system faults.

Although this method is intuitive and simple, it is relatively ineffective in detecting system faults.

2. Machine Learning Methods

Machine learning models are divided into statistical machine learning and deep learning models. The statistical machine learning comprises support vector machine (SVM), K-means clustering algorithm and decision trees. Deep learning includes deep belief network (DBN), convolutional neural network (CNN) and recurrent neural network (RNN).

SVM is a supervised machine learning method that projects data samples onto a high-dimensional feature space, maximising the distance between the hyperplane and the nearest observations from differing classes. It is commonly employed for fault classification and has been applied in steel production processes (Russo *et al*., 2021), laser welding processes (Chen *et al*., 2019), optical grinding processes (Zhang *et al.*, 2015), and intermittent distillation towers (Taqvi *et al*., 2018).

The K-means algorithm is an unsupervised learning technique that partitions *n* data points into *k* clusters based on minimizing the sum of squared distances between each data point and its assigned cluster's mean. Originally introduced in signal processing, it is now applied to diverse areas including monitoring of multimodal processes and fault classification. Researchers have proposed PCA-K-means (Gokilavani, and Bharathi, 2021), adaptive K-means (Tong *et al.*, 2013), etc., and applied them to the field of fault detection and identification for continuous and batch processes.

Decision trees represent the relationships between variables in a tress structure in machine learning. It was first used in decision analysis, and was later introduced into fault detection and diagnosis, for instance, diagnosing faults in pressurised water reactors (Mena *et al*., 2022), and also improved decision trees based on SVM (Demetgul, 2013) and decision trees incorporating feature selection methods (Arockia *et al*., 2022) have been applied to process fault diagnosis.

Deep learning models, such as DBN, CNN, and RNN, have also been introduced into the realm of fault detection and diagnosis by numerous researchers. As an early form of deep learning models, DBN are more maturely used in fault detection and diagnosis. In (Shao *et al.*, 2017), a scalable DBN-based fault diagnosis model is proposed which combines two DBN sub-networks for extracting features in the

spatial and temporal domains and using mutual information technology for fault classification based on variable selection. In (Tang *et al.*, 2018), an adaptive learning rate DBN network containing Nesterov momentum is proposed, which was experimentally validated on data sets from gearbox and locomotive bearing test stands, and the results showed that the fault identification rate of the method was significantly improved, demonstrating the accuracy and robustness of the method. Pan et al. (Pan *et al.*, 2019) proposed a mechanical fault diagnosis of high-voltage circuit breakers based on DBN and transfer learning strategy method. DBN was used to achieve deep mining and adaptive extraction of fault features from sample data and combined with the transfer learning method to enhance fault diagnosis by adjusting the weights of auxiliary and target samples to enhance the weights of training samples through the TrAdaboost algorithm. The results show that the method can obtain stronger generalisation capability.

CNN were initially applied to image recognition designs and later applied to fault diagnosis for fault feature extraction and classification. Junior et al. (Junior *et al*., 2022) employed vibration data derived from multiple sensors as input to a multi-headed one-dimensional CNN model in order to diagnose faults in induction motors. The one-dimensional CNN in each head consists of batch normalisation, two convolutional layers, two pooling layers, a fully connected layer, and a SoftMax layer, with each output corresponding to a motor operating condition. The approach produces a diagnosis accuracy of 99.92 % through experimentation and parameter optimization, and the network is quick to train and test. Wang et al. (Wang et al., 2018) used raw vibration signals of each fault pattern to train a one-dimensional CNN model to detect and recognize data features. The acquired features were then utilised to train a Hidden Markov Model (HMM) classifier for fault diagnosis. The classification results were compared with those of CNN, SVM and BP neural networks and showed that the model gave accurate classification for different bearing datasets. In (Zhong *et al.*, 2020), a method is proposed for the intelligent classification of raw vibration signals for fault location and prediction of damage levels. By converting the dataset into a spectrogram, the original information of the starting signal is retained to the maximum extent and later trained by a deep fully convolutional neural network. The results show that this method has a faster convergence rate, higher accuracy, and better generalisation capability. In (Pan *et al.*, 2018), to address the problem of poor diagnostic accuracy due to noisy signals,

a new CNN-based architecture (LiftingNet) was proposed, which includes a segmentation layer, a prediction layer, an update layer, a pooling layer and a fully connected layer, with the main learning processes of segmentation, prediction, update and cycling. The classification performance of the method was validated using the CWRU dataset to improve the diagnostic accuracy at different motor speeds.

RNN-based fault diagnosis is a method that utilizes time sequences as input data and the depth of the network is determined by the length of the input sequence. This approach is commonly used in monitoring dynamic industrial processes. Abed et al. (Abed *et al.*, 2015) used RNN for bearing fault diagnosis. The fault features were first extracted and filtered by discrete wavelet transform, and then used as inputs to RNN for fault classification. The results show that the method can be used for accurate diagnosis of bearing faults even under non-stationary operating conditions. Malhotra et al. (Malhotra *et al.*, 2016) designed a denoising autoencoder network with the help of LSTM to perform non-linear prediction of the next period based on the vibration values of the previous period and anomaly detection based on reconstruction errors, and the algorithm has strong generalisation capability to different fault modes. In (Bie *et al*., 2021), a model based on complete ensemble empirical model decomposition (CEEMD) and LSTM for fault diagnostics of reciprocating pumps was presented. The vibration signal is processed using CEEMD and singular spectral entropy to generate feature vectors for fault diagnostics using an LSTM classifier. In comparison to traditional neural network approaches, the LSTM has the best classification accuracy. In (Liu *et al.*, 2018) a new RNN-based method for bearing fault diagnosis was proposed. The method utilises GRU-based non-linear predictive denoising autoencoders with high generalisation capability. Multiple vibration values for the next cycle are predicted from the previous cycle, and the reconstruction error between the next cycle data and the output data generated by the GRU-based non-linear predictive denoising autoencoders is then used to detect anomalies and classify faults. The results show that the method has excellent performance with strong robustness and high classification accuracy. In (Jiang *et al.*, 2018), CNN and LSTM are combined to build a convolutional bi-directional LSTM using CNN to extract local features from the original data and then combined with the bi-directional LSTM to extract

temporal correlations, and finally stacking a fully connected layer and linear regression layer for RUL prediction.

3. Multivariate Statistical Methods

The multivariate statistical process monitoring (MSPM) method is an approach that leverages large volumes of historical data to understand the performance of system. The advantages of the MSPM method, which is most attractive because of its ease of design and simplicity of implementation, have made the MSPM technique popular for monitoring many industrial processes. The basic idea of MSPM technology is to provide the process engineer with a concise collection of monitoring statistics that describe the desired process behaviour. The method uses a multivariate projection approach to partition the original multivariate data space into a low-dimensional projection subspace consisting of primary meta-variables and a subspace consisting of residual variables. The statistics and thresholds in each of the two low-dimensional subspaces are constructed to indicate if there has been a change in process behaviour. For the monitored sample measured in the process, the sample data is projected into the two subspaces by means of a projection vector to obtain the corresponding monitoring statistics, which are used to monitor the process operation (Yao and Gao, 2009; Qin, 2012; Yin *et al.*, 2014a).

Common MSPM methods include PCA, Partial Least Squares (PLS), Independent Component Analysis (ICA), etc. PCA simplifies the complexity of analysing the original data space by reducing its dimensionality. It applies a linear transformation to explicit process variables to derive a smaller set of implicit variables, thereby retaining the primary information features contained in the original data. The PLS method was proposed in (Wold, 1973), and it is a commonly employed technique for monitoring quality-related processes. It operates by identifying correlations between process and quality variables, accomplishing this by maximizing their covariance. This leads to the formation of orthogonal latent variables, which enable data reduction while preserving crucial information. Consequently, PLS decomposes the process variable space into two orthogonal subspaces: one directly linked to quality and another that is quality independent. By establishing monitoring statistics for each of these subspaces, PLS effectively identifies the emergence of quality-related faults.

Both multivariate statistical methods, PCA and PLS, require data to satisfy the assumption of a Gaussian distribution. However, the real-world industrial process

data often exhibit non-Gaussian characteristics. The ICA-based method is designed to cope with this non-Gaussian. Compared to PCA and PLS, ICA uses the higher order statistics of process data to extract mutually independent non-Gaussian components. This characteristic of ICA allows it to more effectively analyse non-Gaussian process data, as it can capture more complex, higher-dimensional data relationships. However, the ICA-based fault detection method has its drawbacks. Firstly, the ICA algorithm is more complex compared to the solution algorithms of the PCA and PLS models. Secondly, the independent components estimated by the ICA model must have a non-Gaussian distribution (or at most one Gaussian variable), and if there are more than two Gaussian variables, the ICA method will fail.

The three MSPM methods mentioned above have all been developed as a result of the increasing use of MSPM. The introduction of the PCA algorithm to process monitoring was first implemented in (Wise *et al.*, 1990). As the PCA algorithm is a static process monitoring method and cannot address dynamic processes, an improved dynamic PCA algorithm was proposed, and the algorithm was applied to distillation column process monitoring (Ku *et al.*, 1995). Multi-block PCA algorithms (Chen and McAvoy, 1997) have been proposed to address the problem that standard PCA algorithms are too computationally intensive and difficult to perform fault identification when dealing with processes with many variables. The standard PCA algorithm is a linear time-invariant model (Ding *et al.*, 2013; Yin *et al.*, 2014b), but many real industrial systems are slowly time-varying, so adaptive recursive PCA algorithms have been proposed. Since the standard PCA algorithm is a linear algorithm and many real industrial process systems are strongly nonlinear, some scholars have proposed the kernel PCA algorithm (KPCA), while others have used localization methods to obtain nonlinear PCA algorithms (Wang *et al.*, 2008). The literature (Liu *et al.*, 2013) proposes a multi-level PCA (Multi-level PCA) based fault detection and diagnosis method. In (Kresta *et al.*, 1991), the PLS algorithm was first introduced to industrial processes for process monitoring. The standard PLS algorithm assumes a linear relationship between the predictor and response variables. However, in cases where the relationship is nonlinear, the standard PLS method may not provide accurate results. To address this limitation, some scholars have proposed the introduction of the kernel approach to the standard PLS algorithm to obtain the kernel PLS algorithm (KPLS) (Wen *et al.*, 2012), which

has good process monitoring results in strongly non-linear industrial processes. However, in the standard KPLS algorithm, the projected data needs to be centred in the feature space, which can lose some of the features reflected in the original data. Therefore, to address this problem, a KPLS algorithm based on the optimal preference matrix is proposed in the literature (Yi *et al.*, 2017) to solve this problem. Besides, improved algorithms of the total projection to latent structures (T-PLS) algorithm have also been proposed by some scholars. The literature (Zhao and Sun, 2014) gives the T-PLS method in multivariate space and applies this method to online process monitoring. Kano et al. (Kano *et al.*, 2003) were the first to apply the ICA algorithm to process monitoring. The standard ICA algorithm is also linear, so many learned improvements have been made to make ICA a non-linear process. The literature (Zhang and Qin, 2007) focuses on ICA process monitoring methods applied to non-linear systems, and also introduces kernel methods into the ICA algorithm to obtain the kernel ICA algorithm (KICA) as a way to monitor non-linear systems (Zhang *et al.*, 2014).

Among the data-based fault detection and diagnosis methods, multivariate statistical process monitoring methods have the largest number of research papers and application cases. As the most central of these methods, PCA will be described in detail in the following section, along with its principles and the corresponding fault detection statistics.

2.2 Overview of Fault Prognosis Methods

Fault typically exhibit a gradual deterioration process before they trigger severe implications. Once such a fault transpires, the process slowly follows a trajectory to reach a new steady state or gradually deteriorates until it collapses. Traditional fault detection and diagnosis technologies only issue alerts and diagnoses after the monitored statistics exceed their corresponding control limit. However, by this stage, the fault has already escalated significantly. If faults can be detected at an early stage and measures can be taken to prevent them from occurring in the first place, damage to the entire process can be reduced or even avoided. Therefore, to enhance process safety and stability, minimise the damage caused by faults, and prevent faults from beaching the normal operating zone, it is crucial to anticipate the process's future state. As a result, process fault prognosis technology has become a new research trend in process monitoring.

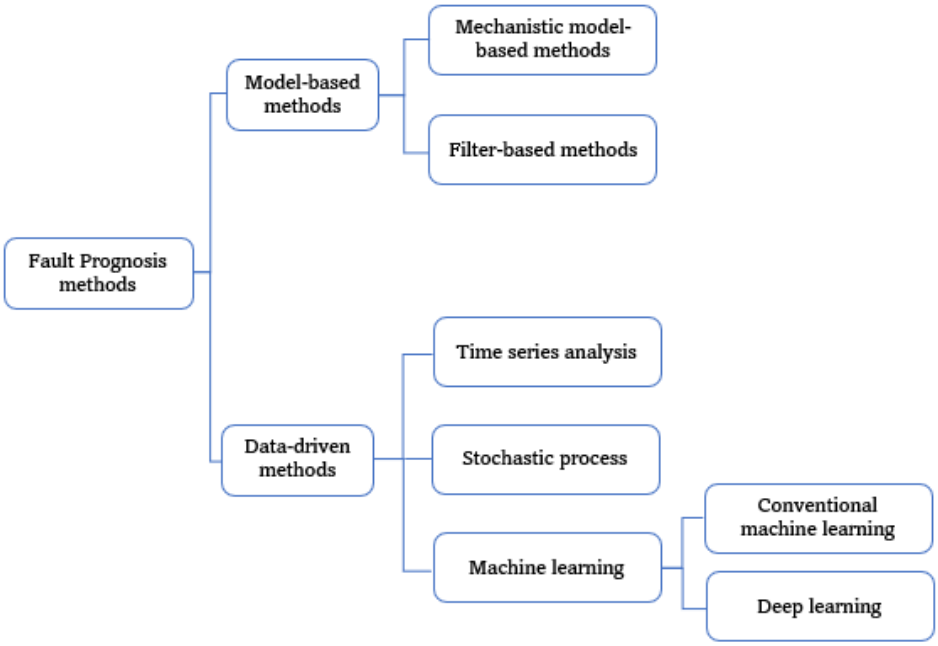

Figure 2.3 Classification of fault prognosis methods

In Figure 2.5, a classification of methods for fault prediction techniques is provided. The advent of fault prediction technology can be traced back to the research conducted for online maintenance systems (Lu and Saeks, 1979). However, the development of fault prognosis, which studies the operational characteristics of the fault's early stage, has been hindered by the small magnitude of faults and the difficulty in detecting system signs. Recently, with advances in fault detection and diagnosis technology, fault prognosis technology has encountered new development opportunities (Reis and Gins, 2017b).

Following the classification of fault detection and diagnosis methods, fault prognosis methods can be divided into model-based methods and data-driven methods (Figure 2.5).

2.2.1 Model-based Methods

Model-based fault prognosis methods often use the physics of the monitored process and process monitoring data to predict the process state. Commonly used methods include mechanistic model-based methods and filter-based methods (Hu *et al.*, 2018).

1. Mechanistic Model-based Methods

The mechanistic model-based approach is focused on examining the internal mechanisms of a system, such as steady-state or transient loads, temperature, as well as other online measurement information obtained from the monitored system. It integrates mathematical representations of the study object's failure mechanisms with macroscopic and microscopic laws of physiochemistry to establish models of performance degradation. These models are then employed to simulate processes for fault prediction. The Paris-Erdogan (PE) model describing the trend of crack expansion was first proposed in the literature (Paris and Erdogan, 1963) and is one of the most applied mechanistic models in mechanical fault prognosis. (Bechhoefer *et al.*, 2008) employed a PE model to construct health indicators for predicting the RUL of a bearing. In (Zhao *et al.*, 2013) the PE model was used to describe the degradation process of a cracked gear and Bayesian estimation was implemented to updates to the model parameters and estimate the RUL of the gear. There are many other physical models besides the PE model, for example, in (Hu *et al.*, 2016a) the Norton creep rate is used to describe the creep degradation process of a turbine and combined it with a stochastic filtering method to achieve RUL estimation. In (Oppenheimer and Loparo, 2002) the Foreman crack growth law is used to predict the RUL of a rotating shaft. In (Jianhui *et al.*, 2003), the model of an automotive suspension system was simulated by using the combining of singular perturbation methods of control theory and dynamic state estimation techniques and the time-averaged mode probabilities are used to predict the RUL. In (El-Tawil and Jaoude, 2013) the stochastic description is used to develop a non-linear damage model for predicting the RUL of pipes. The mechanistic model of fault can clearly show the internal structure and connections of the object of study, so that the corresponding physical meaning can be explained in detail, giving intuitive prediction results with high prediction accuracy. However, a more accurate mechanistic model can only be constructed with a deep understanding of the fault mechanism of the object of study, and with the increasing complexity and uncertainty of modern equipment, the cost of acquiring knowledge of the fault mechanism is significantly high. At the same time, mechanistic models of fault are based on assumptions and simplifications of macroscopic and microscopic laws, which differ to a certain extent from the actual fault process, so this fault prognosis method receives certain limitations in its application.

2. Filter-based Methods

Filter-based methods often require a highly accurate mathematical model of the monitored system, and such methods include mainly Kalman filter-based and particle filter-based methods. Kalman filter-based methods process the observed signals with noise through the state space model of the system to minimise the state estimation error and use the dynamic information of the process to obtain a set of equations for recursive estimation, which is also widely used in the field of forecasting because it also obtains the forecast equations of the system. In (Yang and Liu, 1999), a Kalman filter is used to track and estimate the speed of a DC motor, while a multi-step prediction formula is derived based on a single-step prediction formula to achieve fault prediction of a DC motor. In (Lim *et al.*, 2017) multiple Kalman filters are integrated together to model the degradation process by first dividing the degradation process of the system into stages and then adapting different Kalman filters to the different stages for state prediction and verified the effectiveness of the algorithm with an aero-engine dataset. In (Harrath *et al.*, 2019), the Extended Kalman Filter (EKF) is used to extract several features online in the time, frequency, and time-frequency domains from the vibration data of a real mechanical bearing and evaluated the best features at each instant to obtain an accurate estimate of the RUL. The method has shown good results in prediction over a long-time range.

The particle filter-based prognosis method first models the system dynamics and obtains a minimum variance estimate of the system state by finding a set of random sample approximations that propagate through the state space to represent the probability density and replacing the integration operation with the sample mean. The method can handle non-linear Gaussian systems, which has led to its wide application in the field of audio processing and fault prognosis. In (Orchard and Vachtsevanos, 2009), a non-linear state space model and a particle filter algorithm are used to update the estimate of the current state and calculate the probability density of the RUL of the failed subsystem, and finally validated the method using planetary gear plate fault data. A machine health condition prediction integrating LSTM and particle filter is proposed in (Niu *et al.*, 2018). The LSTM is trained offline on historical data, followed by a Bayesian approach to construct a prognostic model, and finally a particle filter is used to calculate the posterior probability density function and implement state prediction. The validity of the method is demonstrated on experimental vibration data from a ring gearbox. An Improved

Unscented Particle Filter (IUPF) is proposed in (Rathnapriya and Manikandan, 2020). The relationship between the input and predicted faults is first quantified, and later the error and computation time in measuring the distance between actual and observed particles is reduced by introducing relative entropy into the Unscented Particle Filter. The accuracy and efficiency of the forecast is effectively improved.

In summary, model-based process monitoring methods use knowledge of a priori mechanisms such as laws of material and energy conservations in real processes to accurately reflect process changes, with high detection rates, low false alarm rates, high reliability and high interpretability. However, such methods rely heavily on the mechanistic analysis and deep internal knowledge of the modelled object. The process industry often involves multiple variables and multiple states, and the variables are strongly correlated with non-linear behaviour, while the constant changes in the actual process industry and the random disturbances in the external environment make it difficult to establish an accurate process mechanism model. Therefore, the analytical model-based approach has limitations in the process industry and is more suitable for simple industrial equipment where precise mechanisms are easily accessible.

2.2.2 Data-driven Methods

Although model-based fault prognosis methods can accurately predict faults in real time by studying the inherent characteristics and properties of the monitored object, their broad applicability is often compromised due to the complexity of the monitored processes. On the other hand, data-driven fault prognosis methods have gained popularity in recent years due to their wide applicability, high prediction accuracy and easy modelling process. These attributes have led to an accumulation of numerous remarkable research outcomes. Thus, this chapter introduces some of the mainstream data-driven fault prognosis methods: Time Series Analysis methods, Stochastic Process methods and Machine Learning methods.

1. Time Series Analysis Methods

In time series analysis, historical data from the monitored system are collected at certain time intervals to form a time series. A corresponding time series model of the data is constructed and extrapolated to the future for prediction. Additionally, a model can be fitted based on known historical data to enable future value predictions. For any future time point, the forecast value can be estimated using the

fitted model. This method is effective under the assumption that past development patterns will persist into the future, as short-term forecasting is better with this method. A commonly used time series model for fault prediction is the autoregressive (AR) model.

The AR model is the fundamental and practically applicable time series model. Its coefficients coalesce important information about the state of the system. Other commonly used time series models include the moving average (MA) model, the autoregressive moving average (ARMA) model, and the autoregressive integrated moving average (ARIMA) model. These methods build on the dependence between adjacent observations to fit the time series mathematically. A dynamically compensated AR model for fault prognosis in complex dynamic systems was proposed in (Wang and Han, 2019). This model overcomes the limitations of the linear AR model by dynamically compensating for residuals induced by faults. Applying the algorithm to the Tennessee Eastman (TE) Process model demonstrates its efficacy by achieving multi-step advance prognosis and yielding more accurate trend predictions by dynamically compensating for residuals. In (Cai *et al*., 2021) a multi-stage fault prognosis method is presented that combines stage identification with Bayesian networks (BNs) and ARMA models to address multi-stage fault prognosis under complex operating conditions. The method was tested and validated using data from a permanent magnet synchronous motor (PMSM) degradation experiment covering multiple degradation types and stages. The results show that the proposed model performs better in terms of overall prognosis accuracy and stage prognosis accuracy compared to a single model. In (Gómez-Pau *et al*., 2020), a method is presented for predicting the RUL of medium voltage power connectors based on the resistance degradation trajectory. The method involves analysing time series data on electrical resistance using the ARIMA model. Several connectors' experimental results demonstrate the viability and precision of the proposed method for online RUL prediction of power connectors. The method enables the implementation of predictive maintenance schedules and helps to determine when power connectors need to be replaced.

2. Stochastic Process Methods

The stochastic process-based approach refers to the use of mathematical statistics to construct stochastic process models to describe process failure trends. As uncertainty and fluctuating environments can modify equipment’s failure

mechanisms in real time, stochastic process models excel describing the time-varying uncertainties in currently operating equipment's the failure generation mechanisms and operating environments.

Among these methods, the most common one is the Markov chain-based approach, which assumes that the degradation state of the monitored system follows Markov rows. Meaning, the degradation state at the time $t+1$ is only related to the current degradation state at the time *t* and is unaffected by the previous degradation states. In (Yan *et al.*, 2011), degradation state classification was performed using fuzzy C-means, followed by the implementation of fault prognosis using a dynamic multiscale Markov model, and applied to the RUL estimation of rotor equipment. In (Tang *et al.*, 2019), the dynamics of the conditional degradation model are characterised by quantifying the degradation signal through a discrete-time Markov chain and used for RUL estimation of turbofan engines. In many cases, however, the degradation process contains many hidden states that cannot be directly observed. HMM have been introduced into the field of fault prediction for such cases. The method first calculates the current possible degradation states based on performance data, then calculates the mean and variance of each state, and finally sums each state to achieve a RUL estimate. In (Chinnam and Baruah, 2003; Baruah and Chinnam *, 2005), the application of various improved HMM models for fault diagnosis, fault prognosis and estimation of probability density functions for remaining service life is investigated. In (Tobon-Mejia *et al.*, 2011), a degradation model of bearing is established by feature extraction of the sensor data and then using the extracted feature to build a Gaussian mixture HMM for the estimation of the RUL. In (Galagedarage Don and Khan, 2019), a Hidden Markov Model-Bayesian Network hybrid system is combined with a new fault prediction technique to perform fault prognosis. The HMM is trained using normal operating data and then is used to estimate the historical log likelihood of the data and predict future system states. Bayesian network is used to isolate several different industrial faults. The method was validated by predicting and isolating process faults in the TE process. However, the time complexity of failure prediction methods based on HMM and their improvements is large, while a large amount of whole-life performance data is required for state modelling. Furthermore, only the mean and variance of the RUL can be calculated, and the probability density function

distribution of the RUL cannot be obtained, which are not conducive to their application in fault prognosis and health management.

3. Machine Learning Methods

Machine learning based fault prognosis methods are an important branch of the field of data-driven fault prognosis, enabling more accurate predictions of fault trends and RUL. Depending on the depth of the machine learning model structure, the approach can be divided into two parts: the conventional machine learning approach and deep learning approach.

The conventional machine learning approach for fault prediction aims to use the original measurement data or the features extracted from the original measurement data as the input to the neural network, continuously adjust the structure and parameters of the network through certain training algorithms, and use the optimised network to predict the process fault trend and remaining life online, without any a priori information in the prediction process and based entirely on the prediction results obtained from the monitoring data. Current methods based on conventional machine learning mainly include Back Propagation (BP) Neural Network, Radial Basis Function (RBF) Neural Network and Extreme Learning Machine (ELM).

BP neural network is a multilayer forward network with unidirectional propagation. In (Shao and Nezu, 2000), an early study of RUL estimation methods based on BP neural networks was carried out and compared with the ARMA model. The experimental results verified that the RUL estimation methods based on BP neural networks have better long-term prediction capabilities than the ARMA model. In (Gebraeel *et al.*, 2004), BP neural network-based prediction models for individual bearings and batch bearings are developed. Using the vibration monitoring information during the life cycle and fusing the parameters of each bearing indicator regression model with certain weights, an online parameter updating method is proposed, which effectively ensures the accuracy of the RUL prediction. The time series analysis was able to predict the trend of data changes, and the BP neural network adjusted the prediction error in real time to ensure the accuracy of the RUL prediction. In (Gebraeel *et al.*, 2004), a genetic algorithm-BP neural network approach for the prediction of RUL is investigated, and the weights of the BP neural network are globally optimized using the genetic algorithm, and the results showed that the genetic algorithm-BP neural network method was superior to the

conventional BP neural network method. The artificial neural network was trained with accelerated life data at different temperatures and the experimental analysis showed that the life data predicted by the artificial neural network were in good agreement with the actual life data.

RBF neural networks are three-layer feedforward networks with a single hidden layer and are capable of approximating any continuous nonlinear function with arbitrary accuracy (Powell, 1987; Bors, 2001). The main difference between RBF neural networks and BP neural networks is that the output of each hidden neuron is a function of the distance between the input vector and the centre of the RBF of this hidden neuron, rather than the weighted sum of the input vector. In recent years RBF neural networks have been successfully applied in signal processing, system modelling, process control, fault diagnosis and fault prognosis, and have shown extraordinary advantages. It is proposed in (Liu *et al.*, 2010) that the key to RBF neural network models lies in the correct selection of suitable RBF centres, and the number and locations of RBF centres in the hidden layer directly affect the approximation capability of the network. In (Chen *et al.*, 2016), a multivariate grey RBF hybrid model was proposed for the RUL estimation of industrial equipment, incorporating the advantages of the grey model and RBF neural network, which effectively ensures the prediction accuracy and has practical engineering application value.

ELM is a novel learning algorithm for single hidden layer feedforward neural networks. The basic idea of the training process lies in randomly selecting the input weights and the bias values of the hidden layer, manually selecting the number of neurons in the hidden layer based on practical engineering experience and determining the output weights through the least squares method to achieve rapid determination of the network structure and parameters. Because ELM has the characteristics of fast learning speed and strong generalization ability, it has now attracted widespread attention from scholars and engineers in the fields of fault diagnosis, fault prediction and reliability assessment of engineering equipment. In (Liu *et al.*, 2015), the features reflecting the bearing degradation process were extracted by the joint approximate diagonalisation method of the two-layer feature matrix, and the extracted features were input into the ELM model to achieve accurate prediction of the RUL of the bearing. The advantage of the ELM method is that it enables fast fault prognosis and reduces the model training time, while not

using the gradient descent algorithm and avoiding the problem of falling into local minima easily. However, due to the random generation of input weights and hidden layer bias values, there is no guarantee that the ELM network training will have a stable effect.

In recent years, subject to the increase of computing power, the increase of big data and the development of effective algorithms, deep learning techniques combined with brain-like cognitive mechanisms for data processing have opened the bottleneck of traditional neural networks in practical applications. As a new representation learning tool, deep learning has been widely used in the fields of computer vision and natural language processing (Young *et al.*, 2018; Karthikeyan and Priyakumar, 2021). Deep learning uses multi-hidden layer networks to extract information from input data in a layer-by-layer learning manner, and its deep architecture allows it to form high-level representations, attributes or categories through multiple levels of abstraction (Chang, 2015), which has led to an increasing interest in deep learning methods for process monitoring. Neural networks have achieved more extensive research and application in the field of industrial process monitoring due to their self-learning and self-adaptive characteristics (Sohaib *et al.*, 2017). However, whether the fault sample data is sufficiently complete and representative, as well as the convergence of deep learning algorithms, training speed and real-time diagnosis are all constraints on the development of traditional neural network-based fault diagnosis techniques. Therefore, how to combine deep learning algorithms to design new data-driven process monitoring methods has become a hot research topic in the field of process control in recent years. Current methods based on deep learning method mainly include Deep Belief Network (DBN), Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

In (Zhang and Zhao, 2017), high prediction accuracy for the RUL was achieved by using DBN to extract features from the capacity decay of Li-ion batteries and using the extracted features as the inputs to a correlation vector machine. In (Zhang *et al.*, 2017), a multi-objective evolutionary algorithm was combined with a traditional DBN training technique to simultaneously evolve multiple DBNs into two conflicting objectives, and then the evolved DBNs were used to construct an integrated model for RUL estimation. Yu et al. (Yu *et al.*, 2020) proposed a DBN model consisting of improved condition restrict Boltzmann Machines (ICRBMs) to

predict the RUL of a hydraulic pump by adding timing linkage factors and constraint variables to nodes on the same layer enhanced the original RBM. Following Bispectral analysis, the normalised data of the vibration signal's features were utilised for the training and testing of the DBN model. Comparative experiments show that the improved DBN model can achieves higher prediction accuracy than the original DBN, BP neural network, and SVM models. A DBN-based model for backlash error prediction was proposed by (Li *et al.*, 2017). The DBN is first constructed by stacking RBMs, and later the backlash error is used as model input to predict the future state. The method was verified to be superior by comparing with streamwise learning, support vector machines and back propagation neural networks. In (Deutsch *et al.*, 2017; Deutsch and He, 2018), a method is proposed for predicting the RUL of a rotating equipment by fusing DBN and feedforward neural networks (FNN), combining the feature extraction capability of DBN with the predictive performance of FNN. Based on this, DBN and particle filtering are effectively combined to obtain the probability distribution of the RUL, which further improves the prediction accuracy. However, DBN can only give good results in short-term forecasting and its long-term forecasting performance is poor. As can be seen from the above, the fault diagnosis method based on DBN networks has certain advantages for the processing of unlabelled data of low dimensional data of industrial processes due to its simple structure and self-learning during unsupervised training. The layer-by-layer training process also helps to extract higher order non-linear features of complex sample data and avoids the diffusion problem of deep networks. However, due to the complex stacking of the network structure, the training speed and training efficiency are relatively low.

In (Cheng *et al.*, 2018), a new online data-driven framework is proposed to predict the RUL of bearings using CNN. The training data were first processed using the Hilbert-Huang transformation and a new non-linear degradation metric was constructed. A CNN was then used to identify hidden patterns between the degradation metric and the vibration data, resulting in an automatic estimation of bearing degradation. In (Ren *et al.*, 2018), a CNN is applied to predict the RUL of a bearing. By combining the extracted features of the bearing degradation process into an image through the spectral principal energy vector, the CNN can give accurate prediction of the RUL and achieve the prognosis. In (Mazaev *et al.*, 2021), a Bayesian convolutional neural network-based fault prognosis method for solenoid

valves is proposed. Significant physical features extracted from the electromechanical data (valve current signal images) are used as network inputs to improve the accuracy of fault prediction. The CNN-based fault prognosis method is suitable for long-term prognosis of large amounts of data, where the size of the convolutional kernel in the convolutional layer controls the extraction of locally spatially relevant features of the input information, which can enhance certain features of the original signal while reducing the effect of noise; the pooling layer replaces single point values with statistics of features in adjacent regions, reducing the amount of data processing while retaining valid information. CNNs can therefore effectively target industrial data from multiple sources (time series, spectrograms, monitoring images, etc.). Convolutional computation of local links is also useful for learning and describing local association information for strongly correlated industrial process data. However, at the same time, the complexity of CNNs requires significant training time and a large amount of network inputs with labelled 2D data, increasing the upfront workload.

The basic idea of the RNN-based fault prognosis method is to use RNN to model the development of faults. In (Heimes, 2008), RNN-based prediction of the RUL is implemented and the model is trained by the back propagation through time (BPTT) algorithm and an extended Kalman filtering method. The experiments showed that the performance of the proposed method was significantly better than that of the multilayer perceptron -based method. In (Peng *et al.*, 2017), a fault prognosis method for transmission system gearboxes using the noise signal ratio of the current signal as a system health indicator is proposed. An RNN model was developed, and a real time recurrent learning algorithm was used to implement online prognosis. System health thresholds are used to determine the system health status and to develop appropriate maintenance strategies. Li et al. (Li *et al.*, 2019) predicted the deterioration tendency of rolling bearings using RNN and reinforcement learning. The singular spectral entropy of the vibration signal was used to characterise the bearing's deterioration, and the trend of deterioration was separated into multiple phases by moving average noise reduction. Then, reinforcement learning is used to optimise the number of hidden layers and the number of nodes within each hidden layer of the RNN model. In terms of prediction accuracy and convergence speed, the reinforcement-learning RNN model outperforms ELM, SVM, and the original RNN model.

RNNs have been applied to a wide range of sequential data problems with excellent results. Because of the length of the sequence, the gradients of the model error with respect to network weights W (the weight matrix from the input layer to the hidden layer) and U (the weight matrix from the hidden layer outputs) during network training decay or amplify exponentially as time goes forward, resulting in vanishing gradient (the gradient is close to zero and the network parameters cannot be updated) and gradient explosion (the gradient tends to infinity and the model parameters are unstable and cannot converge). Thus, while simple recurrent networks can theoretically establish dependencies between states at long intervals, in practice only short-term dependencies can be learned due to the presence of exploding or vanishing gradients, which is known as the long-term dependency problem. The gradient dispersion and gradient explosion problems were first discovered by (Hochreiter and Schmidhuber, 1997) and have been effectively solved due to the increasing research and application of deep neural network structures in recent years. LSTM networks and gated recurrent unit networks avoid the long-term dependency problem by introducing gates that enable each recurrent unit to adaptively capture dependencies at different time scales (Cho *et al.*, 2014). In contrast to the recursive calculation of the system state established by the RNN, the three gating units of the LSTM establish a self-loop in the internal state of the network: the input gate determines the update of the internal state by the input at the current time and the system state at the previous time; the forgetting gate determines the updating of the internal state at the previous time to the current internal state; and the output gate determines the update of the internal state to the system state. In view of the different contributions of different gates to the learning ability, GRU network simplifies the structure of the neural network and improves the learning efficiency by removing the gates with small contributions and their corresponding weights, and evolves into two gates, the update gate and the reset gate. The function of the reset gate is similar to that of the input gate of the LSTM unit, while the update gate implements the functions of both the forget gate and the output gate. In (Zhang *et al.*, 2018), a RUL prediction model for Lithium batteries was developed using LSTM, which has more stable performance in long-term prognosis than SVM and conventional RNN. The probability distribution of the RUL was also obtained using Monte Carlo simulation method. In (Long *et al.*, 2022), a fault prognosis method based on gated recursive units is proposed, where noise and outliers are eliminated by data pre-processing. The prognosis results

using back propagation neural network and LSTM were compared with those of GRU. It is shown that GRU gives more accurate predictions for different prediction starting points. The method was validated using a dataset of hydrogen fuel cells.

RNN-based fault prognosis methods can be trained by combining the predicted state of the previous moments with the predicted state for the current moment, which not only improves the accuracy and stability of the prognosis, but also speeds up the convergence of the algorithm, and these make it play an important role in the field of fault prognosis and RUL estimation. Moreover, its memorability and chain-connected patterns facilitate the extraction and representation of non-linear features of industrial process dynamics. Also, for the variable length and irregular sampling of industrial processes, RNNs have a more stable performance in learning and testing different sequence lengths.

2.3 A Simulated Continuous Stirred Tank Reactor Process

In this thesis, a simulated CSTR process with feedback control (Yoon and MacGregor, 2001) is used to demonstrate the feasibility of the proposed fault diagnosis and fault magnitude prediction method. An exothermic irreversible first-order reaction ($A \rightarrow B$) takes place in the reactor. The process diagram is shown in Figure 2.8.

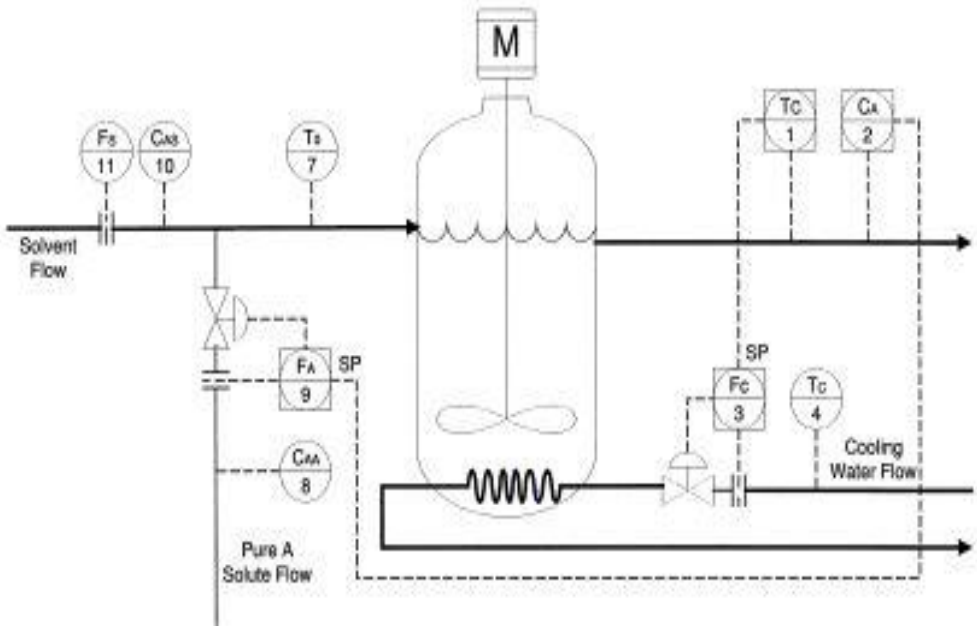

Figure 2.4 A continuous stirred tank reactor process

The simulation model can be described by the following differential equations, which are developed from mass and energy balances:

$$\frac{dC_A}{dt} = \frac{F}{V}C_{A0} - \frac{F}{V}C_A - k_0 e^{-\frac{E}{RT}}C_A \tag{2.44}$$

$$V_\rho C_P \frac{dT}{dt} = \rho C_P F(T_0 - T) - \frac{aF_c^{b+1}}{F_C + \frac{aF_C^b}{2\rho_C C_{PC}}}(T - T_{cin}) + (-\Delta H_{rxn})Vk_0 e^{-\frac{E}{RT}}C_A \tag{2.45}$$

where C_A is the product concentration, T is the temperature of the reaction, T_C is the cooling water temperature, T_0 is the inlet temperature, and C_{A0} is the inlet concentration which is presented as two parts, the reactant and the solvent ($C_{A0}F_0 = C_{AA}F_A + C_{AS}F_S$). Through the empirical relationship ($UA = aF_C^b$), there is a correlation between the heat transfer coefficient and coolant flow rate. Random process noise stem from poisoning of the reaction and fouling of cooling coils are introduced as pre-coefficients (a_1 and a_2) for reaction constants and heat transfer coefficients.

$$k = a_1 k_0 e^{-\frac{E}{RT}};\ \ UA = a_2 a F_C^b \tag{2.46}$$

The form of process noise is unified as the following first-order function:

$$x_t = x_{t-1} + \sigma_e e_t \tag{2.47}$$

where t represents the current sampling time, $e_t \sim N(0,1)$, and $\sigma_e e_t$ is the white noise contained in the system.

In the simulation, the measured process variables are the input concentrations (C_{AA}, C_{AS}), the inlet temperature (T_0), the product concentration (C_A), the reactor temperature (T), the inlet flow rates (F_A, F_S), the cooling water flow rate (F_c), and the temperature of cooling water (T_C).

The product concentration (C_A) and reactor temperature (T) are controlled variables and are controlled by reactant flow rate (F_A) and cooling water flow rate (F_C) respectively. Measurement noise and process noise are added to the system measurement variables and process variables, respectively, to simulate real situations. At steady state, $C_A = 0.265\, kmol/m^3$, $T = 394K$.

The measured process variables in the system are:

$$x = [T\ \ C_A\ \ F_C\ \ F_A\ \ T_0\ \ C_{AA}\ \ C_{AS}\ \ T_C\ \ F_S]^T$$

The other variables are constants during the simulation process, and the process is controlled by PI controllers. The system variables and initial conditions are shown in Table 2.1.

Simulation parameters	$F = 1m^3/min; V = 1m^3; \rho = 10^6\, g/m^3;$ $k_0 = 1 \times 10^{10} min^{-1}; E/R = 8330.1K;$ $C_{PC} = 1\, cal/(gK); \rho_c = 10^6\, g/m^3;$ $-\Delta H_{rxn} = 130 \times 10^6\, cal/(kmol); C_P = 1cal/(gK);$ $b = 0.5; a = 1.678 \times 10^6\, (cal/min)/K$
Controller parameters	$K_C(T) = 2; T_i(T) = 3; K_C(C_A) = -3; T_i(C_A) = 2.72$
Initial conditions	$T_0 = 323K; T_{cin} = 365K; T = 368.25K;$ $F_C = 15\, m^3/min; C_A = 0.8\, kmol/m^3;$ $F_A = 0.1\, m^3/min; F_S = 0.9\, m^3/min;$ $C_{AA} = 19.1\, kmol/m^3; C_{As} = 0.1\, kmol/m^3$

Table 2.1 Parameters and conditions in CSTR model

2.4 Conclusion

In conclusion, process monitoring proves critical to the management of industrial operations. It discovers and corrects faults in order to ensure safety, efficiency, and product quality. As the complexity of industrial processes escalates, process monitoring becomes increasingly important, given that faults can lead to significant impacts in such highly integrated and intensified processes. This chapter elucidates the fundamental tasks, components, and implementation stages of industrial process monitoring, along with the numerous sorts of failures that can arise in industrial processes.

Additionally, an overview of various fault detection and diagnosis methods, particularly in the context of industrial processes has been provided and delves into different techniques in data-based approaches such as signal processing, multivariate statistical-based methods, and machine learning-based methods (including deep learning techniques), comparing the advantages and limitations of these fault detection and diagnosis methods based on a literature review. Following

that, because previous process monitoring methods have primarily focused on fault detection and diagnosis, there is a scarcity of research on fault prediction in process systems, a review of fault prediction methods for industrial processes is presented to emphasise the importance of data-driven methods.

Moreover, a simulated CSTR process has been introduced, which serves as a case study to illustrate the feasibility of the proposed fault diagnosis and magnitude prediction method in the following chapters. This model offers an exothermic irreversible first-order reaction scenario, incorporating elements of feedback control. This simulation is characterized by a set of developed differential equations based on mass and energy balances and includes factors such as product concentration, reactor temperature, cooling water temperature, and inlet concentration. Several failure modes will be simulated to demonstrate the robustness and applicability of the methods discussed in the context of real-world industrial processes.

Chapter 3 Technical Preliminaries and Algorithm Descriptions

In the preceding chapter, an extensive literature review was delved, highlighting the significant advancements, and identifying the knowledge gaps in the realm of data-based process monitoring. This chapter aims to build on that foundation by presenting a detailed mathematical exposition of the computational techniques that are pivotal in fault diagnosis and severity prediction. It serves as a crucial link between the theoretical underpinnings reviewed in Chapter 2 and the practical applications and case studies explored in Chapter 4. Here, we begin with a comprehensive discussion on Principal Component Analysis (PCA), a cornerstone technique in data analysis and reduction, laying out its mathematical structure and significance in process monitoring. Subsequent sections will further elaborate on other critical algorithms such as AR model, ELM, and LSTM networks, each playing a unique role in advancing fault diagnosis and prediction methodologies.

3.1 Multivariate Statistical Process Monitoring

3.1.1 Principal Component Analysis

Principal component analysis (PCA) aims to identify a smaller set of principal components (also referred to as latent variables) that represent the underlying structure of larger set of variables by reducing the dimensionality of the data. Considering a data matrix with n observations of m variables with certain dependencies, a smaller number of composite variables (principal components) are created. These new variables allow the information contained in the original m variables to be captured more concisely. The technique involves determining the maximum variance direction in the data and extracting each principal component in decreasing order of variance. These principal components (PCs) are orthogonal i.e., they are uncorrelated and independent. Generally, each PC is a linear combination of the original variables, and their number is typically much smaller than that of the original variables. The PCs retain most of the information of the original variables and are uncorrelated with each other.

The prerequisite for performing PCA on process variables is the acquisition of sample data within the normal range of variation, from which to construct a PCA statistical model that reflects the characteristics of normal production conditions. Suppose the matrix $X^o \in R^{n\times m}$ contains process data collected from the process under normal operation conditions, with n representing the number of samples, m

representing the number of observed variables. As different variables can have considerably different magnitudes, each column $X^o(i)$ of the measurement data X^o must first be standardized to eliminate any unjustified effects that may arise from the different magnitudes:

$$X(i) = \frac{X^o(i) - E\left(X^o(i)\right)}{\left(Var\left(X^o(i)\right)\right)^{\frac{1}{2}}} \qquad i = 1,2,\cdots,m \tag{3.1}$$

This equation transforms the original data set X^o into a standardised data set X, with mean of 0 and a variance of 1 for each column.

The matrix X can be decomposed into the sum of the outer product of the m vectors:

$$X = t_1 p_1^T + t_2 p_2^T + \cdots + t_m p_m^T = TP^T \tag{3.2}$$

Here, $p_i = [p_{i1}, p_{i2}, \cdots, p_{im}]^T \in R^{m\times 1}$ represents the loading vector, and $t_i = [t_{i1}, t_{i2}, \cdots, t_{in}]^T \in R^{n\times 1}$ is the score vector. Additionally, $P = [p_1, p_2, \cdots, p_m] \in R^{m\times m}$ is the loading matrix, and $T = [t_1, t_2, \cdots, t_m] \in R^{n\times m}$ is the score matrix, representing the projection of X in the load directions.

Each score vector is orthogonal to each other, which means that for any i and j, $t_i^T t_j = 0$ holds true when $i \neq j$. Each loading vector is also orthogonal to each other, and the length of each loading vector is 1:

$$\begin{aligned} p_i^T p_j &= 0 \quad i \neq j \\ p_i^T p_j &= 1 \quad i = j \end{aligned} \tag{3.3}$$

Multiplying both sides of equation (2.2) by p_i, we get:

$$t_i = X p_i \tag{3.4}$$

The above equation shows that the score vector is the projection of the data matrix X in the direction of its corresponding loading vector. The PCA of matrix X involves an eigenvector analysis of the covariance matrix of X, $X^T X$. The loading vector of matrix X is the eigenvector of $X^T X$. If the score vectors are arranged in order of their length as follows:

$$\|t_1\| > \|t_2\| > \cdots > \|t_m\| \tag{3.5}$$

then, the loading vector p_1 represents the direction of the largest variations in the data matrix X, p_2 represents the second largest variations in the data matrix X, and p_m represents the direction of the smallest variations in the data matrix X. If there

is a notable linear correlation among the variables in the matrix X, the variation be principally in the direction of the first k ($k < m$) loading vectors. The projection of X on the remaining $(m - k)$ loading vectors will be relatively small, mainly due to the measurement noise.

The PCA model splits the monitoring space into two orthogonal subspaces, which are principal component subspace (PCS), where normal data variation is captured, and the residual subspace (RS), where abnormal variation and noise are captured (Joe Qin, 2003; Elshenawy and Mahmoud, 2018), Figure 3.1 display this division. Each data vector can subsequently be decomposed into two orthogonal vectors by projecting onto these subspaces.

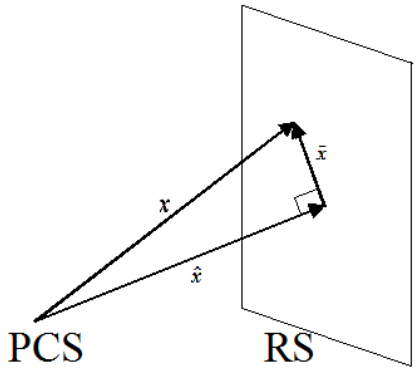

Figure 3.1 Projection of data using PCA

Let $P = [p_1 \;\; p_2 \;\; \cdots \;\; p_k]$, $\tilde{P} = [p_{k+1} \;\; p_{k+2} \;\; \cdots \;\; p_m]$, $T = [t_1 \;\; t_2 \;\; \cdots \;\; t_k]$ and $\tilde{T} = [t_{k+1} \;\; t_{k+2} \;\; \cdots \;\; t_m]$, with k representing the number of the retained PCs. The loading matrix P is an orthogonal matrix, meaning that $PP^T = I$, therefore, the following equation is valid.

$$\begin{aligned} X = \hat{X} + \tilde{X} = TP^T + \tilde{T}\tilde{P}^T \\ = XPP^T + X\tilde{P}\tilde{P}^T = XPP^T + X(I - PP^T) \end{aligned} \tag{3.6}$$

Equation (3.6) reveals that the original data matrix X can be decomposed into two sections, $\hat{X}$ and $\tilde{X}$, which are projections of data matrix X on the subspace PCS and RS, respectively.

In many practical applications, k is often much smaller than m. If the residual matrix $\tilde{X}$, which is mainly caused by the measurement noise, is neglected, it can have the effect of eliminating the noise without causing any significant loss of useful information in the data. Thus, the data matrix X can be approximated as:

$$X \approx \hat{X} = TP^T \tag{3.7}$$

There are primarily two methods for solving the loading vector in PCA:

Firstly, the eigen decomposition of the covariance matrix of the data. Suppose the covariance matrix of the matrix X is denoted as S:

$$S = \frac{1}{n-1} X^T X \tag{3.8}$$

The equation above reflects the eigenvalue decomposition of S:

$$S = [P \quad \tilde{P}] \begin{bmatrix} \widehat{\Lambda} & 0 \\ 0 & \widetilde{\Lambda} \end{bmatrix} [P \quad \tilde{P}]^T \tag{3.9}$$

where the eigenvalue matrix contains the real non-negative eigenvalues ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$) in decreasing order along its diagonal, $\widehat{\Lambda} = diag(\lambda_1, \cdots, \lambda_k)$ and $\widetilde{\Lambda} = diag(\lambda_{k+1}, \cdots, \lambda_m)$. The orthogonal column vectors in the matrix P serve as the loading vectors.

The singular value decomposition (SVD) method provides another way to compute the loading vectors of the matrix X, represented as:

$$X = U \Sigma V^T \tag{3.10}$$

The parameters involved in above equation are given as follows:

$$\begin{gathered} U = [u_1 \quad u_2 \quad \cdots \quad u_n] \in R^{n \times n} \\ V = [v_1 \quad v_2 \quad \cdots \quad v_m] \in R^{m \times m} \\ \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ & \cdots & & \\ 0 & 0 & \cdots & \sigma_m \\ & \cdots & & \\ 0 & 0 & \cdots & 0 \end{bmatrix} \in R^{n \times m} \end{gathered} \tag{3.11}$$

where $(\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m)$ are the singular values of matrix X. The singular values of the data matrix X are the square roots of the eigenvalues of its covariance matrix $\Lambda \in R^{m \times m}$, which contains non-negative real eigenvalues of decreasing magnitude $(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0)$, and the loading vector p_i is given by each column vector v_i in the matrix V.

Geometrically, the essence of PCA is to translate and rotate the original coordinate system so that the origin of the new coordinates coincides with the centre of gravity of the data group points. The first axis of the new coordinate system corresponds to the largest direction of variation in the data, the second axis of the new coordinate system is standardly orthogonal to the first axis and corresponds to the second

largest direction of variation in the data, and so on. These new axes are the loading vectors, $p_i, i = 1, 2, \ldots, m$. If the first k axes of the new coordinate system describe the variation of the original data very effectively, then the original data group points can be approximated by a projection onto the new *k*-dimensional space.

In three dimensions, the geometric interpretation of the PCA is given by Figure 3.2, where (a) is the distribution of observations in the measurement space, P_1 in (b) is the first load direction and P_2 in (c) is the second load direction. The data are mostly distributed in the same plane, so that only 2 PCs are needed to describe the set of data effectively.

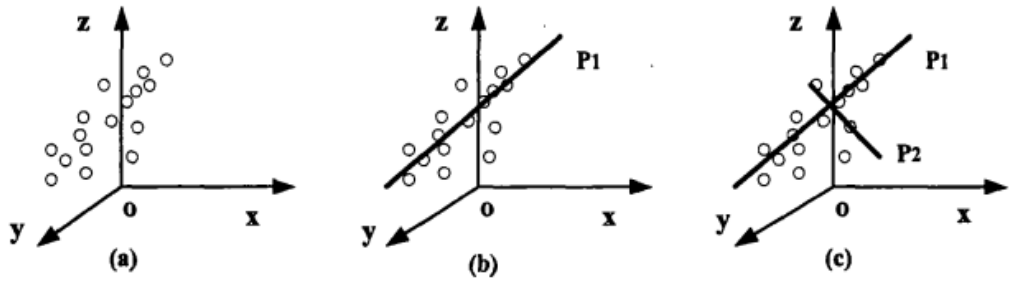

Figure 3.2 Geometric interpretation of the PCA

3.1.2 Fault Detection based on PCA

The PCA based process monitoring model describes the correlation among process variables under normal operating conditions due to constraints such as material balance, energy balance and operational limitations. This method projects the process data vectors onto two orthogonal subspaces (PCS and RS), creating separate statistics on the corresponding subspaces to perform hypothesis testing for determining the process operation. Two commonly used fault detection statistics include the Hotelling's T^2 statistic and squared prediction error (SPE) statistic, which represent the degree of variation in the data projection in the principal component subspace and the residual subspace, respectively. By comparing them with the corresponding control limits, it is possible to determine whether a fault has occurred in the process.

a. Hotelling's T^2

The Hotelling's T^2 monitoring index is the weighted sum of squares of the score vectors. It represents the extent to which each sample deviates from the model in terms of trend and magnitude and is a measure of the variability within the model,

which measures the variations of the sample in the PC subspace by the estimated value $\hat{S}$ of the correlation matrix. The definition of T^2 statistic is:

$$T^2 = X^T \hat{S}^{-1} X = X^T P \hat{\Lambda}^{-1} P^T X \tag{3.12}$$

where P is the loading matrix containing the first k loading vectors and S is a diagonal matrix composed of the first k major eigenvalues of the covariance matrix (the number of PCs kept in the PCA model) and $\tilde{k} = m - k$ represents the number of neglected PCs. T^2 reflects changes in multivariate variables through fluctuations in the principal vector modes within the principal component model.

The control limits of T^2 statistic can be calculated using the F distribution as follows:

$$\delta_T^2 = \frac{(m-1)(m+1)k}{m(m-k)} F_\alpha(k, m-k) \tag{3.13}$$

where m is the number of variables, k is the number of PCs kept in the PCA model, and $F_\alpha(k, m-k)$ is the function of the F distribution with degrees of freedom k and $m-k$ at a confidence level of $1-\alpha$.

b. Squared prediction error

The squared prediction error of the PCA model is also known as the Q-statistic, which represents the variation of data in the RS. The SPE at the sample i is a scalar measure of the deviation of the measurements, $x(i)$, from its PCA model prediction, $\hat{x}(i)$, and is a measure of the variability of the data outside the model. The definition of SPE statistic is:

$$\begin{aligned} SPE &= \left\|\tilde{X}\right\|^2 = X^T \tilde{P} \tilde{P}^T X \\ &= [(I - PP^T)X]^T (I - PP^T)X \end{aligned} \tag{3.14}$$

The SPE statistic represents the variation in the data that is not accounted for by the PCA model. When the SPE is too large, it indicates that the current measured data sample does not obey the PCA model developed from the normal process operation data and, hence, deviates from normal process operation. The control limit of SPE is calculated as:

$$\delta_{SPE}^2 = a(b + cz_\alpha)^d \tag{3.15}$$

where α is the significance level, z_α is the standard normal distribution with a confidence level of $(1-\alpha) \times 100\%$, and other parameters are as follows.

$$
\begin{aligned}
a &= \sum_{i=k+1}^{m} \lambda_i \\
b &= 1 + [\theta_2 h_0 (h_0 - 1)]/a^2 \\
c &= \left(\sqrt{2\theta_2 h_0}\right)/a\,; \quad d = 1/h_0 \\
\theta_2 &= \sum_{i=k+1}^{m} {\lambda_i}^2\,;\ \theta_3 = \sum_{i=k+1}^{m} {\lambda_i}^3 \\
h_0 &= (1 - 2a\theta_3)/(3\theta_2^2)
\end{aligned}
\tag{3.16}
$$

Here λ_i is the i^{th} largest eigenvalue of the covariance matrix S. The SPE statistic measures the deviation between X and PCS, indicating the deviation from the main correlation between the variables. Therefore, a large SPE means that the variable relationship under normal operating conditions is destroyed, and a process failure has occurred.

After the PCA model is established, the multivariate process can be monitored by monitoring the T^2 and SPE statistics and the process is considered under normal operation if the following conditions are satisfied:

$$T^2 = \left\| \widehat{\Lambda}^{-(1/2)} P^T X \right\|^2 \leq \delta_T^2 \tag{3.17}$$

$$SPE = \|(I - PP^T)X\|^2 \leq \delta_{SPE}^2 \tag{3.18}$$

Here, δ_T^2 and δ_{SPE}^2 are the control limits of the T^2 and SPE respectively.

Although both T^2 and SPE are applied to process monitoring, they are different ways of describing the degree of change in a process. The SPE statistic reflects the extent to which the data deviates from the PCA model in the residual subspace, measuring changes that break the correlation of normal processes, often implying anomalies. The T^2 statistic, on the other hand, reflects the extent to which the data deviates from the PCA model in the principal component subspace, measuring the distance from the centre of the model for changes in the principal directions. Because the principal component subspace contains normal process variations, which have large variance and usually represent signals, and the residual subspace contains variations with small variance, which usually represent noise, the control limits of T^2 are larger than those of SPE in terms of the magnitude of the control limits. If the T^2 control limit is exceeded, the magnitude of the fault must be relatively large, whereas the SPE control limit only contains the noise part of the normal process, so even if the magnitude of the fault is small, it can be detected by

SPE. If the T^2 statistic of a sampled data exceeds the control limits, but the SPE statistic does not, then this means that the data did not break the normal correlation between the variables, but simply shifted in the main PCA subspace. This could be a fault, or it could be a change in the operational range.

3.2 Data-driven Modelling Techniques for Fault Prediction

3.2.1 Autoregressive Model

The AR model is a special case of the ARMA model, and it is important to first understand the structure of the ARMA model. Assuming that the response of a system at time t is a stationary time series $\{x_t\}$, if the value of x_t is related not only to the response at the previous q moments $(x_{t-1}, x_{t-2}, \cdots, x_{t-q})$ but also to the disturbances at the previous p $(w_{t-1}, w_{t-2}, \cdots, w_{t-p})$, the general structure of ARMA (q, p), following the idea of multiple regression, is as follows:

$$\begin{cases} x_t = \sum_{i=1}^{q} \varphi_i x_{t-i} - \sum_{j=1}^{p} \theta_j w_{t-j} + w_t \\ w_t \sim NID(0, \sigma^2) \end{cases} \tag{3.19}$$

where q and p represent the orders of the autoregressive and moving average parts respectively. The parameter φ_i $(i = 1, 2, \cdots, n)$ represents the regression coefficients, the parameter θ_j $(j = 1, 2, \cdots, m)$ represents the moving average coefficients, and w_t represents a normal independent distribution with mean 0 and variance σ^2 (white noise).

If the x_t relates only to the response at the previous moment, then:

$$\begin{cases} x_t = \sum_{i=1}^{q} \varphi_i x_{t-i} + w_t \\ w_t \sim NID(0, \sigma^2) \end{cases} \tag{3.20}$$

The above equation is then the mathematical structure of the AR (q) model.

The process of building the AR model is the process of linear parameter estimation, which aims to select the appropriate parameters so that the residuals w_t of the model are white noise series. Therefore, the data needs to be pre-processed before using the AR model. The AR model is only valid if the time series data needs to meet the steadiness requirement. The judgement is based on whether the autocorrelation coefficient ρ_q of the time series data can rapidly converge to zero as the value of q increases. If so, the steadiness requirement is met. The

autocorrelation coefficient is a measure of the degree of correlation between any two elements of the series x_t. For stochastic processes, the autocorrelation coefficient between two elements at two time instances is defined as follows:

$$\rho_q = \frac{cov(x_t, x_{t+q})}{\sqrt{varx_t varx_{t+q}}} \tag{3.21}$$

For non-stationary data, the difference method is usually used to adjust the smoothness of the data, and the expression for the first order difference is:

$$x_t = x_{t+q} - x_t \ (q > 1) \tag{3.22}$$

Repeat the differential transformation until a stationary time series is obtained.

The next step is to determine the order of the AR model. For model fitting of a stationary random series, the estimation of the model order is an important issue. An improper choice of order can seriously affect the accuracy of the fitted results. The appropriate order of the model is usually determined using a criterion that considers the accuracy of the model fitting and the number of model parameters. There are two commonly used criteria.

The first one is the Akaike Information Criterion (AIC) criterion, which was proposed by the Japanese scholar Akaike in 1973 and is also known as the minimum information criterion. For a stationary time series $\{x_t\}$ fitted to an AR model, the AIC criterion function is defined by the formula:

$$AIC(q) = \ln(\hat{\sigma}_q^2) + 2q/N \tag{3.23}$$

where q is the order of the model, N is the length of the time series and σ_q^2 is the maximum likelihood estimates of the noise variance for the q^{th} order model. $AIC(q)$ decreases as the order q increases where the $\ln(\hat{\sigma}_q^2)$ term decreases. Therefore, the order corresponding to the smallest value of $AIC(q)$ is taken to be the optimal order.

The second is the Bayesian Information Criterion (BIC), which was proposed by Schwarz and is defined as follows:

$$BIC(q) = \ln(\hat{\sigma}_q^2) + \frac{q \ln N}{N} \tag{3.24}$$

In the information criterion approach, there is a preference for models that produce a minimum value criterion, which can be used as a basis for model selection by comparing the AIC and BIC values of different models.

Once the sample series has been smoothed and the model order has been determined, the parameters of the AR model can be estimated. There are various methods, and this section only uses least squares estimation to estimate the model parameters.

When the optimal order of the model is q, the AR (q) model can be expressed as:

$$\begin{cases} x_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_q x_{t-q} + w_t \\ w_t \sim NID(0, \sigma^2) \end{cases} \tag{3.25}$$

By parameter estimation, the parameters φ_i $(i = 1,2, \cdots q)$ and σ^2 are estimated from the known observations $\{x_t\}$ $(t = 1,2, \cdots N)$. For the residuals $\{w_t\}$ of the model, the above equation gives:

$$w_t = x_t - \varphi_1 x_{t-1} - \varphi_2 x_{t-2} - \cdots - \varphi_n x_{t-q} \tag{3.26}$$

According to the least squares estimation, the residuals equation of the q^{th} order autoregressive model has the following linear expansion form when minimising the sum of squares of the model residuals:

$$\begin{cases} w_{q+1} = x_q \hat{\varphi}_1 + x_{q-1} \hat{\varphi}_2 + \cdots + x_1 \hat{\varphi}_q - x_{q+1} \\ w_{q+2} = x_{q+1} \hat{\varphi}_1 + x_q \hat{\varphi}_2 + \cdots + x_2 \hat{\varphi}_q - x_{q+2} \\ \vdots \\ w_N = x_{N-1} \hat{\varphi}_1 + x_{N-2} \hat{\varphi}_2 + \cdots + x_{N-q} \hat{\varphi}_q - x_N \end{cases} \tag{3.27}$$

Then the following matrix can be obtained:

$$W = \begin{bmatrix} w_{q+1} \\ w_{q+2} \\ \vdots \\ w_N \end{bmatrix} \quad \hat{\varphi} = \begin{bmatrix} \hat{\varphi}_1 \\ \hat{\varphi}_2 \\ \vdots \\ \hat{\varphi}_q \end{bmatrix} \quad X = \begin{bmatrix} x_q & x_{q-1} & \cdots & x_1 \\ x_{q+1} & x_q & \cdots & x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_{N-2} & \cdots & x_{N-q} \end{bmatrix} \quad Y = \begin{bmatrix} x_{q+1} \\ x_{q+2} \\ \vdots \\ x_N \end{bmatrix} \tag{3.28}$$

Then the residuals equation can be rewritten as:

$$W = X\hat{\varphi} - Y \tag{3.29}$$

The least squares solution of the estimated model parameters matrix $\hat{\varphi}$ is then: (Chatfield *et al*., 2019)

$$\hat{\varphi} = (X^T X)^{-1} X^T Y \tag{3.30}$$

3.2.2 Extreme Learning Machine

In the traditional gradient based neural network training algorithms, the network weights are obtained through multiple iterations of the learning algorithm. This training process is not only time-consuming, but also computationally intensive, resulting in low network training efficiency. In the implementation of the gradient based neural network training algorithms, the training process can easily be trapped in local minima. To address the above problems, Huang et al. (2006) proposed the ELM neural network. In contrast to previous gradient based neural network training algorithms, the ELM randomly selects weights and biases in the hidden layer, and then calculates the weights in the output layer by a regularized linear regression method.

An ELM neural network is basically a single hidden layer neural network shown in Figure 3.3, where the numbers of neurons in the input, hidden and output layers are I, l, O, respectively.

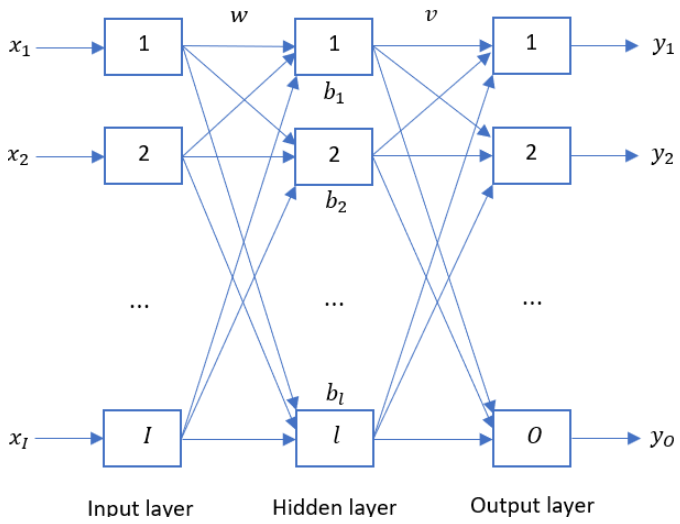

Figure 3.3 Structure diagram of the Extreme Learning Machine Neural Network

Let the connection weights between the input layer to the hidden layer and the hidden layer to the output layer be *w* and *v* respectively:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_l \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1I} \\ w_{21} & w_{22} & \cdots & w_{2I} \\ \cdots & \cdots & \ddots & \vdots \\ w_{l1} & w_{l2} & \cdots & w_{lI} \end{bmatrix}_{l \times I} \tag{3.31}$$

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_l \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1I} \\ v_{21} & v_{22} & \cdots & v_{2I} \\ \cdots & \cdots & \ddots & \vdots \\ v_{l1} & v_{l2} & \cdots & v_{lO} \end{bmatrix}_{l \times O} \tag{3.32}$$

The hidden layer bias is:

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_l \end{bmatrix} \tag{3.33}$$

The activation function is $g(x)$, then for an output Y with N samples can be expressed as:

$$Y = Hv \tag{3.34}$$

where, H is the hidden layer output matrix:

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & g(w_2 \cdot x_2 + b_2) & \cdots & g(w_l \cdot x_1 + b_l) \\ g(w_1 \cdot x_2 + b_1) & g(w_2 \cdot x_2 + b_2) & \cdots & g(w_1 \cdot x_2 + b_l) \\ \cdots & \cdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & g(w_2 \cdot x_N + b_2) & \cdots & g(w_1 \cdot x_N + b_l) \end{bmatrix}_{N \times l} \tag{3.35}$$

The following theorems were proposed by Huang et al. (Huang *et al.*, 2006):

1. Given a single hidden layer forward neural network (SLFN) with l hidden layer neurons, N distinct samples (x_i, y_i), where $x_i = [x_{i1}, x_{i2}, \cdots, x_{il}]^T \in R^I$ and $y_i = [y_{i1}, y_{i2}, \cdots, y_{iO}]^T \in R^O$, and an infinitely differentiable function $g(x)$, then for any assignment $w_i \in R^I$ and $b_i \in R$, there is a hidden layer output matrix H that is invertible and satisfies $\|Hv - Y\| = 0$.
2. Given any small error $\varepsilon(\varepsilon > 0)$, and an infinitely differentiable function $g(x)$, for N distinct samples (x_i, y_i), there exists a SLFN with M $(M \leq N)$ hidden layer neurons that has a hidden layer output matrix H that is invertible and satisfies $\|Hv - Y\| < \varepsilon$ for any assignment $w_i \in R^I$ and $b_i \in R$.

From the above two theorems, if the number of hidden layer neurons l is greater than or equal to the number of training samples N, and the input weight w and biases b are randomly assigned, then the ELM network can approximate the training samples with zero error. Thus, training for the SLFN model is also equivalent to solving the equation (3.34) for the least-and-multiply solution $\hat{v}$.

$$\left\|H\left(\hat{w}_1, \cdots, \hat{w}_M, \hat{b}_1, \cdots, \hat{b}_M\right)\hat{v} - Y\right\| = \min_v \|H(w_1, \cdots, w_M, b_1, \cdots, b_M)v - Y\| \tag{3.36}$$

However, when N is large, an excessive number of hidden layer nodes imply a larger computational effort, so the size of the hidden layer nodes can be reduced to M so that the training error can be approximated by an arbitrarily small ε. In this

case, the Moore-Penrose generalised inverse matrix H^+ of H can be used to solve for equation (3.36).

$$\hat{v} = H^+ Y \tag{3.37}$$

The most common method currently used to solve the generalized inverse matrix H^+ is the singular value decomposition method, regardless of whether $H^T H$ is a singular or non-singular matrix.

In contrast to traditional single hidden layer feedforward neural networks, ELM randomly selects weights and biases in the hidden layer and then calculates the weights of the output layer by a regularised linear regression method (Huang *et al.*, 2004). Even though the weights of the hidden layer are randomly generated, ELM maintains the universal approximation capability of SLFN. Therefore, from the perspective of learning efficiency, ELM networks are not only simple to operate, but also learn faster and have better global search capability, which can overcome the problem of traditional neural networks falling into local optima.

3.2.3 Long Short-term Memory Neural Network

RNN usually uses the backpropagation through time algorithm for training, but as the time information stored in the RNN network increases, the gradient tends to disappear or explode (Pascanu *et al.*, 2013). To solve this problem, in (Hochreiter and Schmidhuber, 1997) a long and short-term memory network structure containing memory units is proposed. A typical LSTM neural network cell is shown in Figure 3.4. Both LSTM and RNN are trained with similar parameters on the neural network structure, with the main difference being on the recurrent neuron nodes.

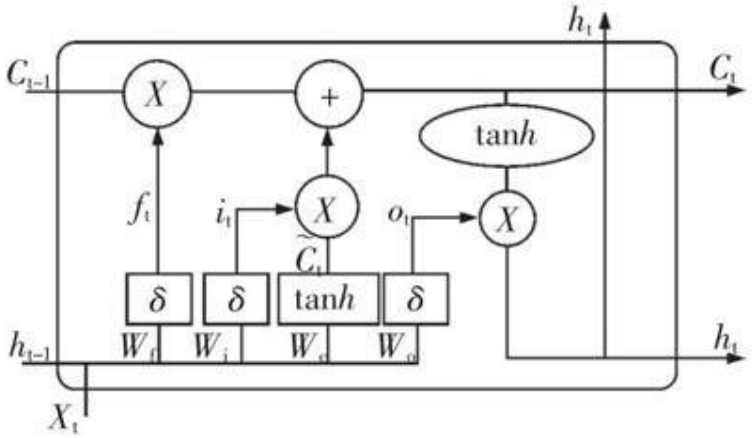

Figure 3.4 LSTM cell architecture

In Figure 3.4, h_t is the hidden state, which represents short-term memory, and C_t is the cell state, which represents long-term memory. Compared to the RNN basic recurrent neurons, the LSTM has an extra state or memory for storing long-term memory and another for storing short-term memory. The long-term state is updated slowly and stores mainly long-term dependent information, while the short-term state is updated faster and changes more under different moments of the neuron. The LSTM also introduces three gated cyclic units to control the updating and forgetting of memories, ensuring that important information is always remembered, and less important information is forgotten, and enabling the storage and flow of memories in the hidden layer units. In Figure 3.4, f_t, i_t, and o_t are, respectively, the forget gate unit, input gate unit, and output gate unit, which are used to implement information discarding, updating, and status updating respectively, and W_f, W_i, W_c, W_o are weight matrices. The specific internal process of LSTM can be summarized into the following three stages.

First is the forgetting phase, where for the LSTM neural unit at time *t*, the input information received includes the input X_t at the current time, the cell state C_{t-1} and hidden state h_{t-1} at the previous time step. However, the LSTM discards some information from the cell state to make room for new information from the input of current time. Here the forget gate unit uses the $sigmoid$ activation function to selectively retain the important information, by outputting a value between 0 and 1, where 1 indicate complete retention and 0 indicates complete discard. The forget gate unit can be calculated using the following equation.

$$f_t = sigmoid\big(W_f[h_{t-1}, x_t] + b_f\big) \tag{3.38}$$

Next comes the memory selection phase, where input gate unit i_t sifts through the input X_t and short-term memory h_{t-1} at the current time to determine how much information will be stored in the cell state and updates the cell state. This step is divided into two main stages, first using the $sigmoid$ layer to update the input values, then using the $tanh$ layer to create a candidate state $\tilde{C}_t$ and update the cell state by multiplying the candidate state with the input values. The specific implementation is as follows.

$$i_t = sigmoid(W_i[h_{t-1}, x_t] + b_i) \tag{3.39}$$

$$\tilde{C}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3.40}$$

The updated cell state can be expressed as

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.41}$$

The last is the output stage. After the first two phases have finished updating and filtering the information, it will be up to the output gate unit to decide what information is to be output at the next time step. The $sigmoid$ layer is first used to determine which part of the cell state is to be output, and then the output of the $tanh$ and $sigmoid$ layers are multiplied together to obtain the remaining state values:

$$o_t = sigmoid(W_o[h_{t-1}, x_t] + b_o) \tag{3.42}$$

$$h_t = o_t tan\, h(C_t) \tag{3.43}$$

where, b_f, b_i, b_c, b_o are the bias vectors of each layer.

The LSTM model training process uses the backpropagation through time (BP) algorithm, which can be roughly divided into 4 steps:

1. The output value of each neuron is calculated using equations (3.38) to (3.43).
2. The error term for each neuron is calculated. Like the RNN, the backpropagation of the LSTM error term consists of two directions: one is the backpropagation along time starting from the current time t, where the error term is computed at each time period; the other direction of propagation of the error term is towards the upper layer.
3. The gradient of model error with respect to each weight is calculated according to the corresponding error term.
4. Apply a gradient-based optimisation algorithm to update the weights.

There are many different gradient-based optimisation algorithms, such as stochastic gradient descent (SGD) (Amari, 1993), Adagrad (Duchi *et al.*, 2011), RMSProp (Yeung *et al.*, 2018) and others. This thesis uses the adaptive moment estimation (Adam) algorithm (Kingma and Ba, 2014), which is an effective gradient-based stochastic optimisation method that combines the advantages of the Adagrad and RMSProp algorithms to calculate adaptive learning rates for different parameters and uses less storage resources and is used in practical applications. The overall performance is better.

Overall, the LSTM network enables a shift in the information transfer pattern of the original RNN, and the gating mechanism introduced produces paths where

gradients continue to flow for long periods of time, thus fundamentally solving the gradient disappearance or explosion in RNN.

3.3 Conclusion

This chapter has provided a comprehensive understanding of the mathematical algorithms and computational techniques fundamental to fault diagnosis and fault magnitude prediction. This chapter began with a detailed examination of PCA, its mathematical framework, and its application in fault detection and reconstruction of industrial process. It then expanded into a discussion of other vital algorithms such as autoregression, Extreme Learning Machines (ELM), and Long Short-Term Memory (LSTM) networks for fault magnitude prediction.

This chapter has bridged the theoretical concepts introduced in the previous chapter with their practical application, which will be further explored in the following chapters. The algorithms and techniques detailed here are essential for developing robust fault diagnosis and prognosis systems, and their implications extend far beyond theoretical applications, setting the groundwork for real-world implementations and evaluations in complex industrial processes.

Chapter 4 An Improved Reconstruction Approach for Process Fault Diagnosis

4.1 Introduction

As the industrial sector continues to advance, the development of efficient and robust fault diagnosis methods for sensor and process faults has become increasingly critical. A variety of innovative approaches have been proposed to address the challenges inherent in continuous time-varying chemical processes. For instance, an improved method for Residual Based Contribution (RBC) plots to diagnose sensor faults with reduced computational demands introduced in (Elshenawy and Mahmoud, 2018). Similarly, a reconstruction method employing Bayesian Lasso for regression analysis has been put forward in (Yan *et al.*, 2018), enhancing the robustness of abnormal process data reconstruction. The utilization of PCA has been particularly prominent, evidenced by the work in (Bose *et al.*, 2005) leveraged principal component score contributions to analyse and reconstruct data from failed sensors in a closed-loop controlled CSTR. This theme continues with the application of PCA in various settings, from monitoring the effects of faults in high-temperature gas-cooled reactors (Uren *et al.*, 2015; Uren *et al.*, 2016) to detecting and diagnosing faults in nuclear research reactors (Penha *et al.*, 2001) and screw-type water-cooled chiller sensors (Hu *et al.*, 2016b).

This chapter proposes an improvement to the process fault reconstruction method. To validate the practicability of the proposed method, a simulated CSTR process with feedback control is analysed as a case study. A PCA model is constructed using normal operating history data, and the Hotelling T^2 statistic and squared prediction error (SPE) statistic are used for fault detection. PCA models constructed with historical fault data for different faults are used to extract the fault directions. The fault direction of each fault is obtained from the first loading vector under that fault. As a result, the process fault direction matrix can be quickly extracted by combining the major loading vectors of historical fault data for different faults. Based on this fault direction matrix, process fault identification and estimation of fault magnitude can be achieved.

The chapter is organized as follows: Section 4.2 introduces the fault direction matrix and fault detectability. Section 4.3 reviews current fault diagnosis methods and suggests enhancements to the reconstruction-based approach. Section 4.4

details the steps of the proposed fault diagnosis method. Section 4.5 applies the method to a simulated CSTR process, and Section 4.6 concludes with a summary of the chapter's key insights.

4.2 Fault Detectability in PCA based Method

By establishing a PCA model under normal operating conditions, fault detection is carried out by projecting the newly collected process data into the principal component and residual subspaces, establishing the corresponding monitoring statistics T^2 and SPE, and comparing them with their control limits. A fault is detected if one or both monitoring statistics exceed the corresponding control limits. Since PCA analysis does not make use of the information from the process mechanism model, it is difficult to carry out a theoretical analysis of the fundamental issues such as fault detectability and identifiability. With the introduction of the concept of fault direction matrices by Qin and co-workers, these important issues have been partially solved and have provided new ideas for further development of the theory (Dunia and Joe Qin, 1998a; Qin *et al.*, 2001).

4.2.1 Fault Direction Matrix

For the monitored process, the known $J \in Z^*$ system faults can be defined as a fault ensemble $\{F_i\}$ where $i = 1,2,\cdots,J$, based on the process information recorded in the historical process operation data. By analysing the propagation pattern of the faults, each of these faults can be expressed as a subspace in R^m, where m is the number of monitored variables in the system. Denote $\{\Xi_i\}$ as the ensemble of fault direction matrix corresponding to the set of faults $\{F_i\}$, where the fault direction matrix Ξ_i is standard orthogonal. For faults that affect only a single variable or dimension of the data, the direction matrix is a column vector. Conversely. for faults that impact multiple variables or dimensions of data at the same time, the direction matrix becomes a multi-dimension matrix $dim(\Xi_i) = l_i > 1$.

The fault direction matrix characterizes how the fault influences the monitored variables in the process. When fault F_i is present, the resulting sample vector x can be represented by the following:

$$x = x^* + \Xi_i f \tag{4.1}$$

where $x \in R^m$ denote a sample vector of m variables under fault operation conditions, x^* represents the sample vector under normal operating conditions,

which is unknown when a fault occurs, and f is the fault parameter, which is a scalar for one-dimensional faults and a vector for multi-dimensional faults, which may change over time, depending on how the actual fault develops, and $\|f\|$ represents the magnitude of fault.

The underlying assumption in this study is the superposition of faulty data onto normal operating data, which stems from the premise that the process variables' correlation structure can be effectively approximated by a limited number of principal components. This approximation is generally valid in steady-state conditions where process variables exhibit relative stability. Under such conditions, it is postulated that changes in the correlation structure, potentially induced by faults, are sufficiently minor and thus can be encapsulated within the principal components.

The focus of this research is predominantly on faults emerging post-attainment of the process's steady state. Given that these faults arise during a period of assumed stability, the correlation structure of the process variables is unlikely to undergo significant alterations. Consequently, the superposition assumption maintains its validity within this context.

However, it is acknowledged that certain fault types, particularly non-linear faults or those inducing considerable system disturbances, may challenge this assumption. Similarly, the presumption may falter in scenarios where process variables exhibit insensitivity to faults or when faults remain undetectable within these variables. These scenarios present avenues for future research, aiming to refine and expand the applicability of the proposed methodologies.

4.2.2 SPE-based Fault Detectability

The problem of fault detectability based on SPE statistics is a critical aspect in process monitoring. SPE quantifies the deviation of a process variable vector, denoted as x, from its projection onto the PC subspace. When the fault F_i occurs, it influences this deviation, and this influence is captured by the fault direction matrix Ξ_i. This matrix maps the effect of the fault on the process variables.

The SPE statistic corresponding to a sampled data vector x is mathematically expressed as:

$$SPE(x) = \|\tilde{x}\|^2 = \|(I - PP^T)x\|^2 \tag{4.2}$$

where $\tilde{x}$ is the projection of x onto the residual subspace and P represents the matrix of principal component loadings.

Upon the introduction of a fault, the sample vector x is represented as equation (4.1), and substituting equation (4.1) into (4.2) yields:

$$SPE(x) = \|\tilde{x}\|^2 = \|(I - PP^T)(x^* + \Xi_i f)\|^2 = \left\|\tilde{x}^* + \widetilde{\Xi}_\iota f\right\|^2 \qquad (4.3)$$

where $\widetilde{\Xi}_\iota = (I - PP^T)\Xi_i$ represents the projection of Ξ_i onto the residual subspace and $\tilde{x}^* = (I - PP^T)x^*$ represents the projection of x^* onto the residual subspace. If $\widetilde{\Xi}_\iota = 0$, it indicates that the fault has no impact on the residual subspace. In this situation, and thus, the SPE statistical would not register any deviation due to the fault.

However, when $\widetilde{\Xi}_\iota \neq 0$, the fault impacts the residual subspace. Applying the triangle inequality to the SPE equation, which gives:

$$\|\tilde{x}\| = \left\|\tilde{x}^* + \widetilde{\Xi}_\iota f\right\| \geq \left\|\widetilde{\Xi}_\iota f\right\| - \|\tilde{x}^*\| \qquad (4.4)$$

This leads to the equation:

$$\|\tilde{x}\| \geq \left\|\widetilde{\Xi}_\iota f\right\| - \|\tilde{x}^*\| \qquad (4.5)$$

From here, it follows that for the SPE statistic to detect a fault, its value must exceed a predetermined threshold, signifying that the deviation under fault conditions is sufficiently large to be distinguished from normal operational variations. If $\left\|\widetilde{\Xi}_\iota f\right\| - \|\tilde{x}^*\| > \delta_{SPE}$, then $\|\tilde{x}\| > \delta_{SPE}$ must hold and the fault can be detected by the SPE statistic. Furthermore, since $\|\tilde{x}^*\| \leq \delta_{SPE}$, a sufficient condition for a fault to be detectable based on the SPE statistic is:

$$\left\|\widetilde{\Xi}_\iota f\right\| > 2\delta_{SPE} \qquad (4.6)$$

where δ_{SPE} is the predetermined threshold for detectability.

4.2.3 T^2-based Fault Detectability

A study of fault detectability based on T^2 statistics using the concept of fault direction matrix is given by (Dunia and Joe Qin, 1998a). When fault F_i occurs, the value of the T^2 statistic corresponding to the sampled data vector x is given by:

$$T^2 = x^T P \Lambda_k^{-1} P^T x = \left\|\Lambda_k^{-(1/2)} P^T x\right\|^2 \qquad (4.7)$$

where $\Lambda_k \in R^{k\times k}$ is the diagonal array of eigenvalues corresponding to the first k PCs.

Substituting equation (4.1) into equation (4.7):

$$\begin{aligned} T^2 &= \left\|\Lambda_k^{-(1/2)} P^T (x^* + \Xi_i f)\right\|^2 \\ &= \left\|\Lambda_k^{-(1/2)} P^T P P^T (x^* + \Xi_i f)\right\|^2 \\ &= \left\|\Lambda_k^{-(1/2)} P^T \left(\hat{x}^* + \hat{\Xi}_i f\right)\right\|^2 \\ &= \left\|\Lambda_k^{-(1/2)} P^T \hat{x}^* + \Lambda_k^{-(1/2)} P^T \hat{\Xi}_i f\right\|^2 \end{aligned} \tag{4.8}$$

Let $\Theta_i = \Lambda_k^{-(1/2)} P^T \hat{\Xi}_i$. When $\Theta_i = 0$, then $\left\|\Lambda_k^{-(1/2)} P^T x\right\|^2 = \left\|\Lambda_k^{-(1/2)} P^T \hat{x}^*\right\|^2 \leq \delta_T^2$. From equation (4.8) the fault corresponding to the fault direction matrix Ξ_i is not detectable by the T^2statistic value at this point, no matter how large the fault magnitude $\|f\|$ is. Therefore, the necessary condition for a fault to be detectable by the T^2 statistic is:

$$\Theta_i \neq 0 \tag{4.9}$$

From here, we can deduce a condition for fault detectability:

$$\left\|\Lambda_k^{-(1/2)} P^T x\right\| \geq \|\Theta_i f\| - \left\|\Lambda_k^{-(1/2)} P^T \hat{x}^*\right\| \tag{4.10}$$

This implies that for the T^2 statistic to detect a fault, the deviation caused by the fault must be sufficiently large compared to normal operational variations. If $\|\Theta_i f\| - \left\|\Lambda_k^{-(1/2)} P^T \hat{x}^*\right\| > \delta_T$, then $\left\|\Lambda_k^{-(1/2)} P^T x\right\| > \delta_T$ must hold and the fault is detectable by the T^2 statistic. Furthermore, since $\left\|\Lambda_k^{-(1/2)} P^T \hat{x}^*\right\| \leq \delta_T$, a sufficient condition for a fault to be detectable based on the T^2 statistic is:

$$\|\Theta_i f\| > 2\delta_T \tag{4.11}$$

4.3 Fault Diagnosis through Reconstruction

In (Dunia and Joe Qin, 1998b) a geometric analysis method based on the fault direction matrix is proposed and applied to solve the PCA-based fault reconstruction problem in process fault identification. This fault reconstruction method is based on an optimisation approach to find the optimal estimate of the fault magnitude, which eliminates the effect of the fault on the data and estimates the normal value of the data. Through fault reconstruction, the severity (magnitude) of the fault can be estimated, and the type of fault can be identified so that

corresponding measures can be taken to effectively eliminate the impact of the fault on the industrial process.

a. SPE-based fault reconstruction

Fault reconstruction is an intermediate part of PCA based monitoring and is important for the determination of fault types, as well as for the separation and identification of faults. Fault reconstruction is the process of estimating the normal value of data by eliminating the influence of faults on the data. Once the fault direction matrix has been obtained, the normal part x^* of the original fault data x can be reconstructed using the PCA model. Suppose that when a fault F_i occurs, the reconstruction of x along the fault direction matrix Ξ_i is:

$$x_i = x - \Xi_i f \tag{4.12}$$

where, x_i represents the reconstructed normal operation data, f is an estimate of the actual fault parameters in the Ξ_i direction. Theoretically the best reconstruction is to minimize $\|x_i - x^*\|$ for a certain fault size f. However, $x^*$is unknown when the fault occurs, so this idea is not feasible. From equations (4.5) and (4.12), if x_i approaches x^* along the fault direction matrix Ξ_i, the SPE statistic for x_i will gradually decreases to below the corresponding control limit. Thus, an estimate of the fault parameters f in the direction of the fault Ξ_i can be obtained if the SPE value ($SPE(x_i) = \|\tilde{x}_i\|^2 = \left\|\tilde{x} - \tilde{\Xi}_i f\right\|^2$) of the reconstructed data is kept below its control limit.

Then the estimated of the fault parameter on residual subspace is:

$$f = \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T \tilde{x} \tag{4.13}$$

And $\|f\|$ is the estimation of the actual fault magnitude along the direction of Ξ_i.

Substituting (4.13) into (4.12) gives the reconstructed normal sample vector on residual subspace:

$$\tilde{x}_i = \left(I - \tilde{\Xi}_i \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T\right) \tilde{x} \tag{4.14}$$

b. Reconstruction-based fault identification

Once a fault has been detected in the process using SPE statistics, the actual fault type can be identified from the set of possible faults $\{F_i, i = 1,2,\cdots,J\}$ by means of a fault reconstruction method based on SPE statistics.

When the SPE statistics exceed its control limit then a fault is detected. Before performing fault diagnosis, since the root cause of the fault is not known in advance, each fault in the set of possible faults $\{F_i, i = 1,2,\cdots,J\}$ is first assumed to be a real fault in turn. The data will be reconstructed for each fault in the given set of possible faults of the system $\{F_i, i = 1,2,\cdots,J\}$ one by one for the corresponding fault direction Ξ_i.

Suppose that fault F_j occurs, the data x is reconstructed using the subspace Ξ_i corresponding to the fault F_i, i.e., for the i^{th} fault, $i \in [1,J]$:

$$x_{i/j} = \left(I - \Xi_i \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T\right) x \tag{4.15}$$

Projection of $x_{i/j}$ onto the residual subspace:

$$\tilde{x}_{i/j} = \left(I - \tilde{\Xi}_i \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T\right) \tilde{x} \tag{4.16}$$

Then the projection $\tilde{x}$ of the sampling vector x onto the residual subspace when the fault F_j occurs can be expressed in terms of the fault direction matrix Ξ_j as:

$$\tilde{x} = \tilde{x}^* + \tilde{\Xi}_j f \tag{4.17}$$

Substitute (4.17) into (4.16), the reconstructed vector $\tilde{x}_{i/j}$ can be related to the fault-free vector $\tilde{x}^*$ as follows:

$$\tilde{x}_{i/j} = \left(I - \tilde{\Xi}_i \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T\right) \tilde{x}^* + \left(I - \tilde{\Xi}_i \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T\right) \tilde{\Xi}_j f \tag{4.18}$$

From the above equation, it can be found that $\left(I - \tilde{\Xi}_i \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T\right) \tilde{\Xi}_j f = 0$ if the reconstruction of F_i used for the hypothesis is exactly the actual faulty F_j, i.e. when $\Xi_i = \Xi_j$. At this point, the projection of the reconstructed value in the residual subspace is not related to f:

$$\tilde{x}_{i/j} = \left(I - \tilde{\Xi}_i \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T\right) \tilde{x}^* \tag{4.19}$$

Since $\left(I - \tilde{\Xi}_i \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T\right) \tilde{x}^*$ is the projection matrix of $\tilde{x}^*$, we have $\left\| \left(I - \tilde{\Xi}_i \left(\tilde{\Xi}_i{}^T \tilde{\Xi}_i\right)^{-1} \tilde{\Xi}_i^T\right) \tilde{x}^* \right\| \leq \|\tilde{x}^*\|$, then the SPE value of the sampled vector obtained by the reconstruction should satisfy the following conditions:

$$SPE(x_{i/j}) = \left\|\tilde{x}_{i/j}\right\|^2$$
$$= \left\|\left(I - \tilde{\Xi}_i\left(\tilde{\Xi}_i{}^T\tilde{\Xi}_i\right)^{-1}\tilde{\Xi}_i^T\right)\tilde{x}^*\right\| = SPE(x^*) \le \delta_{SPE}^2 \tag{4.20}$$

From the above equation, when the reconstruction is performed using the fault direction matrix corresponding to the true fault, the SPE value of the reconstructed data will be within the control limits. If $\Xi_i \neq \Xi_j$, then this indicates that the fault direction matrix used in the reconstruction does not correspond to the true fault, which also indicates that $\left(I - \tilde{\Xi}_i\left(\tilde{\Xi}_i{}^T\tilde{\Xi}_i\right)^{-1}\tilde{\Xi}_i^T\right)\tilde{\Xi}_j f \neq 0$ and that the SPE will exceed the control limit under the condition that f is sufficiently large. Based on the above description, the following fault identification index can be defined:

$$FIS_{i/j} = \frac{SPE(x_{i/j})}{SPE(x)} \tag{4.21}$$

The values of this index are distributed between 0 and 1. If it is close to 1, then the hypothetical fault F_i used for reconstruction cannot be the real fault, because it does not serve to correct the SPE statistic for the original data. If the index is close to 0, then the hypothetical fault F_i used for reconstruction is a real fault ($j = i$) and the fault is diagnosed. In practice, a diagnostic threshold r can also be defined based on engineering experience, and if $FIS_{i/j} \le r$ then the fault direction matrix used for reconstruction is of the same type as the real fault. If any of the faults in the set of fault direction matrix have a discrimination index greater than the diagnostic threshold for the current data reconstruction, then it is likely that a new unidentified fault has emerged. At this point, process knowledge can be combined to determine if this is a new fault type. Once identified, the fault direction matrix can be added to the existing set of fault direction matrix, so this method has some learning capability.

c. Improved fault direction matrix extraction

In reconstruction-based fault identification studies, the premise is that the fault direction matrix of each type of fault is known, whereas in practice, the fault direction matrix is unknown and needs to be extracted from historical data based on the characteristics of the fault. The literature (Dunia and Joe Qin, 1998b) first proposed a sensor and process fault reconstruction method based on the fault direction matrix with PCA model for the general process fault. Then, in (Yue and Qin, 2001) a combined index is used to more effectively determine the direction of

failure for individual and multiple sensors. In the case of sensor faults on non-control loops, the fault direction matrix is easier to obtain because the faults only occur in the sensor itself and causing the measured values of the variables to deviate from their normal values, but do not directly cause anomalies in the remaining state variables in the process. Therefore, the fault direction matrix for the single sensor fault reconstruction, the direction matrix is generally expressed as a column vector, for example:

$$\Xi_{is} = [0 \quad 1 \quad 0 \quad \cdots]^T \tag{4.22}$$

The above equation represents the fault vector corresponding to the second sensor fault in the sampling vector x.

When multiple sensors fail at the same time, the fault direction matrix is represented by a matrix, for example:

$$\Xi_{is} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \end{bmatrix}^T \tag{4.23}$$

The above equation represents the subspace of faults corresponding to the simultaneous faults of the second and third sensors in the sampling vector x.

For process faults, as there is a certain correlation between measured variables in the production process, so when a process fault occurs, it will cause a set of sensor measurements with a strong correlation to change at the same time. At this time, the fault direction matrix determination method for sensor faults cannot be used. In this case, fault direction matrix could be extracted from the historical process data of various types of faults.

The direction matrix can also be projected to the PCS and RS subspaces as:

$$\Xi_i = \hat{\Xi}_i + \tilde{\Xi}_i \tag{4.24}$$

where $\hat{\Xi}_i = PP^T\Xi_i$ and $\tilde{\Xi}_i = (I - PP^T)\Xi_i$.

In practical fault reconstruction, the residuals of faults are much larger than the residuals of normal values, so reconstruction-based fault identification often uses the projection of the fault direction matrix Ξ_i onto the residual subspace $\tilde{\Xi}_i$, then the extraction process of the fault direction matrix Ξ_i can be transformed into the extraction of $\tilde{\Xi}_i$ by the following method.

If the fault F_i occurs, then the projection of the data vector x in the residual subspace can be expressed as:

$$\tilde{x} = \tilde{x}^* + \tilde{\Xi}_i f \tag{4.25}$$

Since the normal process operation data used to construct the PCA model is scaled to zero mean and unit variance, and the normal part of fault data projected on the residual subspace $\tilde{x}^*$ is small, therefore, it is reasonable to disregard its contribution when the fault magnitude is relatively large in comparison to the normal variations ($\|\tilde{x}^*\| \ll \|\tilde{\Xi}_i f\|$):

$$\tilde{x} \approx \tilde{\Xi}_i f \tag{4.26}$$

Selecting q consecutive observations from the sample data corresponding to the fault F_i:

$$X_i^T = [x(1), x(2), \cdots, x(q)] \tag{4.27}$$

According to equation (4.26) the projection of X_i in the residual subspace is:

$$\tilde{X}_i^T = \tilde{\Xi}_i [f(1), f(2), \cdots, f(q)] \tag{4.28}$$

This show that $\tilde{\Xi}_i$ and $\tilde{X}_i^T$ have the common subspace. Then we apply SVD on $\tilde{X}_i^T$:

$$\tilde{X}_i^T = U_i D_i V_i^T \tag{4.29}$$

where the matrix U_i is a unitary matrix, matrix D_i is a diagonal matrix containing the singular values ($\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m$) in descending order along its diagonal.

This gives the projection of fault direction matrix in the residual subspace for process faults as.

$$\tilde{\Xi}_i = U_i \tag{4.30}$$

However, as mentioned by (Yue and Qin, 2001), in practice, it is necessary to use the singular vector $U_i(:, 1)$ corresponding to the highest singular value as the initial fault direction matrix for reconstruction. If the reconstruction can bring the process to a normal operation region, as indicated by the SPE index, then $U_i(:, 1)$ suitable as the fault direction matrix $\tilde{\Xi}_i$. Otherwise, the singular value vector corresponding to the next highest singular value is incorporated into the fault direction matrix used for the preceding reconstruction. This process is repeated up to J times, until the SPE statistics of the reconstructed data fall within the normal operational range. At this point, $\tilde{\Xi}_i = U_i(:, 1{:}J)$ is chosen as the fault direction matrix.

However, this is sufficient for sensor faults reconstruction, but still cannot be effectively used for process fault reconstruction. Since there is a strong correlation

between process variables, process faults can have different effects on multiple process measurements, which can be characterized by monitoring the affected measurements during abnormal operation. However, when multiple variables are affected, it is difficult to reconstruct the fault data and distinguish all fault categories. In order to successfully reconstruct the faults, the features of the considered faults need to be discovered. Due to the correlation among process variables, a process fault typically causes process measurements to move in a particular direction in the measurement space. The first loading vector P_f of the PCA model for faulty data can effectively extract fault characteristics to represent the fault direction and plays a role in assisting fault reconstruction. Therefore, the fault direction matrix for process faults will be extended as:

$$\tilde{\Xi}_{ip} = U_i P_f \tag{4.31}$$

where P_f is the first loading vector of the loading matrix in PCA fault modelling. This makes it possible to derive the fault direction matrix for process faults intuitively and quickly.

4.4 PCA-based Fault Detection and Diagnosis Process

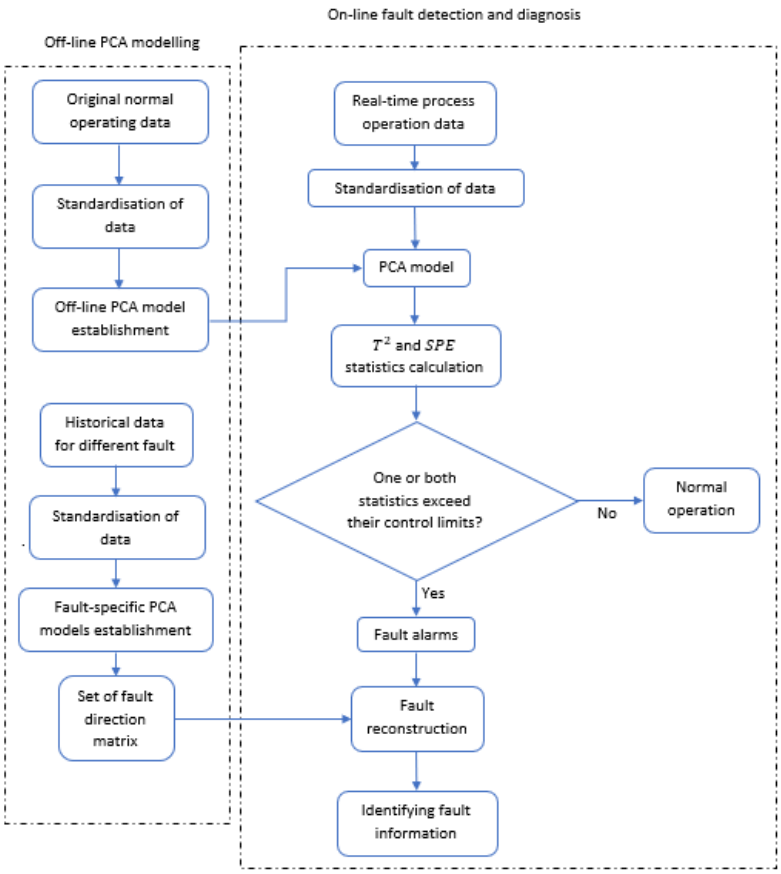

Figure 4.1 Flow diagram of PCA-based fault detection and diagnosis process

Fault detection and diagnosis based on PCA consists of establishing a PCA model offline, performing fault detection and fault diagnosis online as shown in Figure 4.1.

In the offline PCA modelling phase, standardised historical process data from normal operations are used to establish a baseline PCA model, ensuring zero mean and unit variance. The optimal number of principal components (PCs) is selected based on this model. Moreover, for enhanced fault detection, PCA is also applied separately to historical fault data when available. This process yields fault specific PCA models, from which fault direction matrices are extracted for each fault type. These matrices are critical for the accurate reconstruction and diagnosis of faults, allowing for a comprehensive fault detection system.

During online monitoring, the SPE and T^2 statistics of the current sample are calculated and compared with their control limits. If one or both monitoring statistics exceed their control limits, then it is detected that the process is under faulty operation and fault diagnosis needs to be carried out. The fault reconstruction method is used to diagnose the fault.

4.5 Fault Detection and Reconstruction

In this study, a variety of failure modes can be simulated during the process and only 7 faults are considered, as listed in Table 4.1.

Fault No.	Fault description	Type of Fault
1	Decrease in heating medium temperature causes a slow decrease of inlet temperature T_0	Process fault (Incipient fault)
2	Exposure of cooling water pipes to external heat sources or high external ambient temperatures, resulting in a slow increase in the cooling water temperature T_C	Process fault (Incipient fault)
3	The control valve stuck in the fixed position causes a constant inlet solvent flow rate F_S	Process fault (Abrupt fault)

4	Deactivation of the catalyst leads to a reduction in the rate of reaction, resulting a slow increase in the flow of inlet reactant flow rates F_A	Process fault (Incipient fault)
5	Incorrect calibration of the temperature sensor leads to a bias in measurement of output temperature T	Process fault (Abrupt fault)
6	Inconsistent mixing or stirring resulting in a slow increase of inlet reactant concentration C_{AA}	Process fault (Incipient fault)
7	Electrical interference causes a drift in measurement of inlet temperature T_0	Sensor fault (Incipient fault)

Table 4.1 Process and sensor faults

The simulation of different types of faults is implemented by representing them in the following way:

$$X_f = C_1 X_n + C_2 X_n t + C_3 \tag{4.32}$$

where X_n and X_f represent the normal and faulty states of individual process variables, respectively, with t donates the time since the fault onset. Parameters C_1, C_2, and C_3 specify fault characteristics and magnitude. The values of C_1, C_2, and C_3, are chosen to represent abrupt and incipient faults. For example, an incipient fault can be expressed as, $C_1 = 1$, C_2 indicating the rate of change of the fault magnitude, and $C_3 = 0$.

For the faults designed in this study, we select one of the magnitudes of each fault to demonstrate its impact on the process variables. Table 4.3 displays the parameters for each designed fault, and each set of fault data comprises 900 samples. Among the faults listed in Table 4.2, faults 3 and 5 are abrupt faults while others are incipient faults. The parameters chosen for the simulation of faults in the CSTR, as listed in Table 3.1, are based on theoretical considerations and operational norms of CSTR systems (Venkateswarlu *et al.*, 1992; Yunus and Zhang, 2010; Hu *et al.*, 2017; Wang, 2017; Ji *et al.*, 2019). In each of the cases, the fault was introduced to the process at the 201^{st} sampling point and the process variables under these faults are shown in Figures 4.2 to 4.8.

Table 4.2 Parameters of each fault

Fault No.	C_1	C_2	C_3
1	1	$-2.36 * 10^{-6}$	0
2	1	$1.28 * 10^{-5}$	0
3	0	0	0.42
4	1	$6.54 * 10^{-6}$	0
5	1	0	5.91
6	1	$9.49 * 10^{-5}$	0
7	1	$3.69 * 10^{-6}$	0

Table 4.3 Parameters of each fault

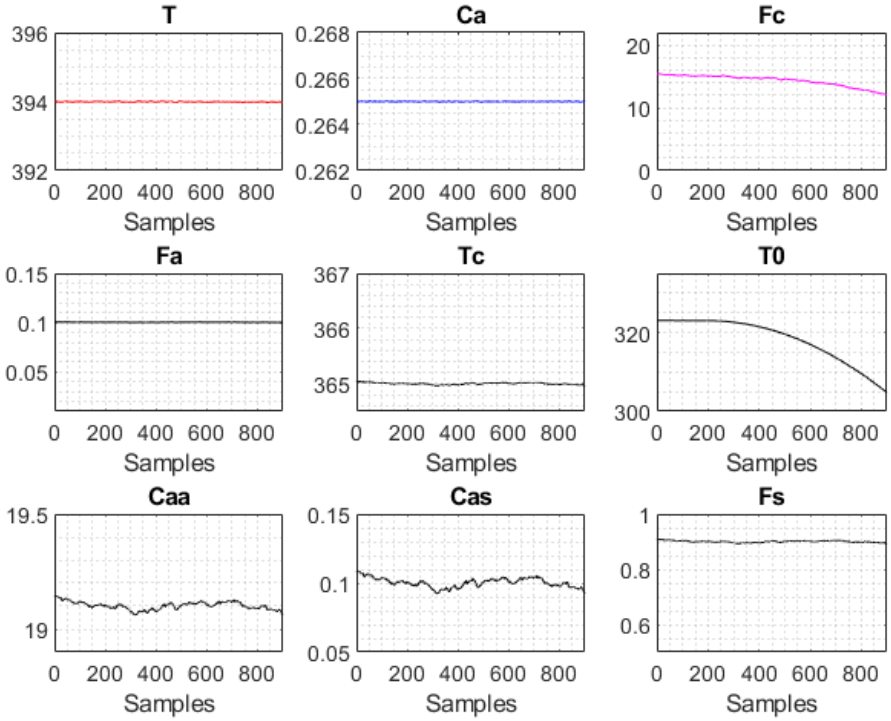

Figure 4.2 Measurements of process variables under Fault 1

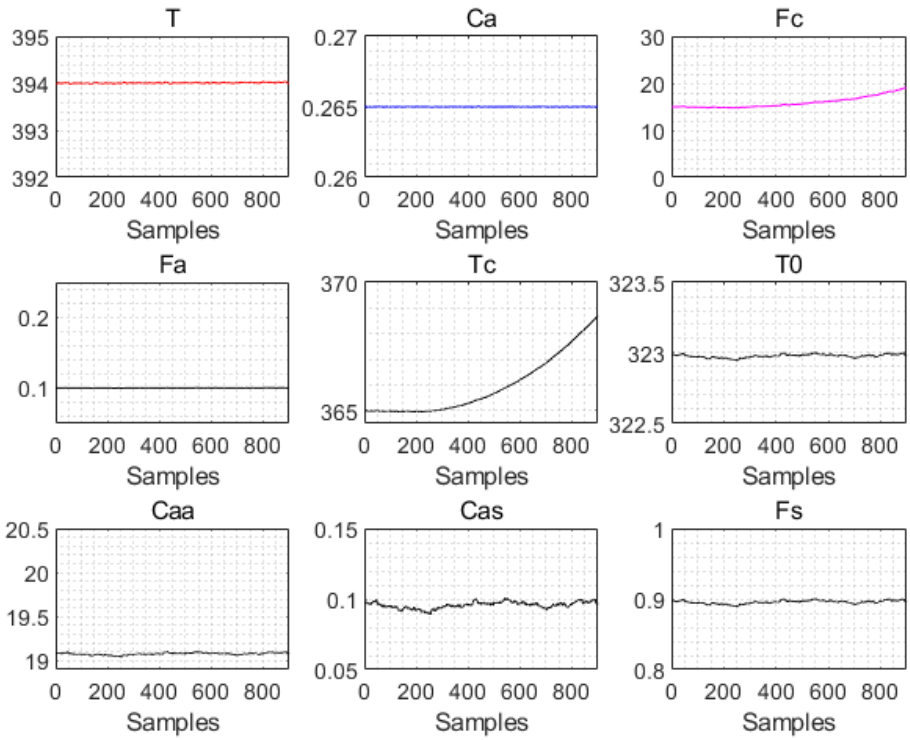

Figure 4.3 Measurements of process variables under Fault 2

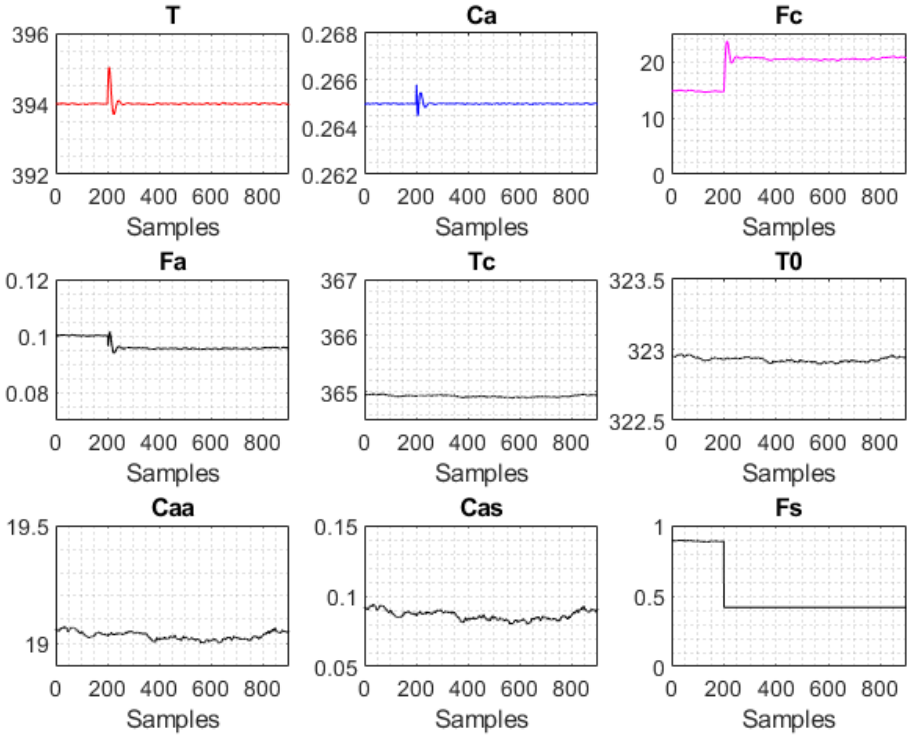

Figure 4.4 Measurements of process variables under Fault 3

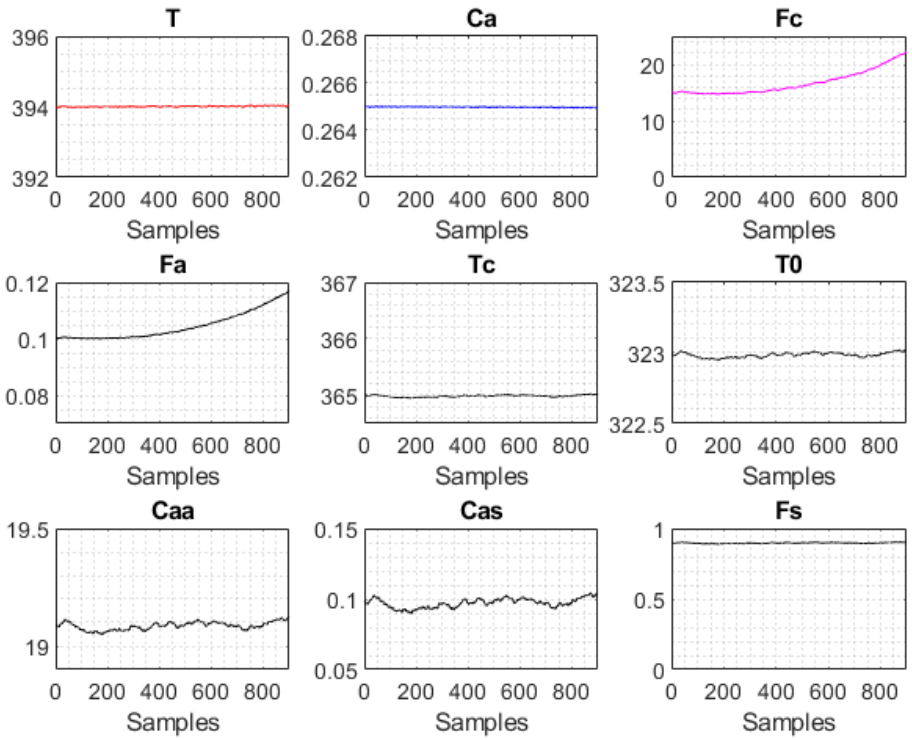

Figure 4.5 Measurements of process variables under Fault 4

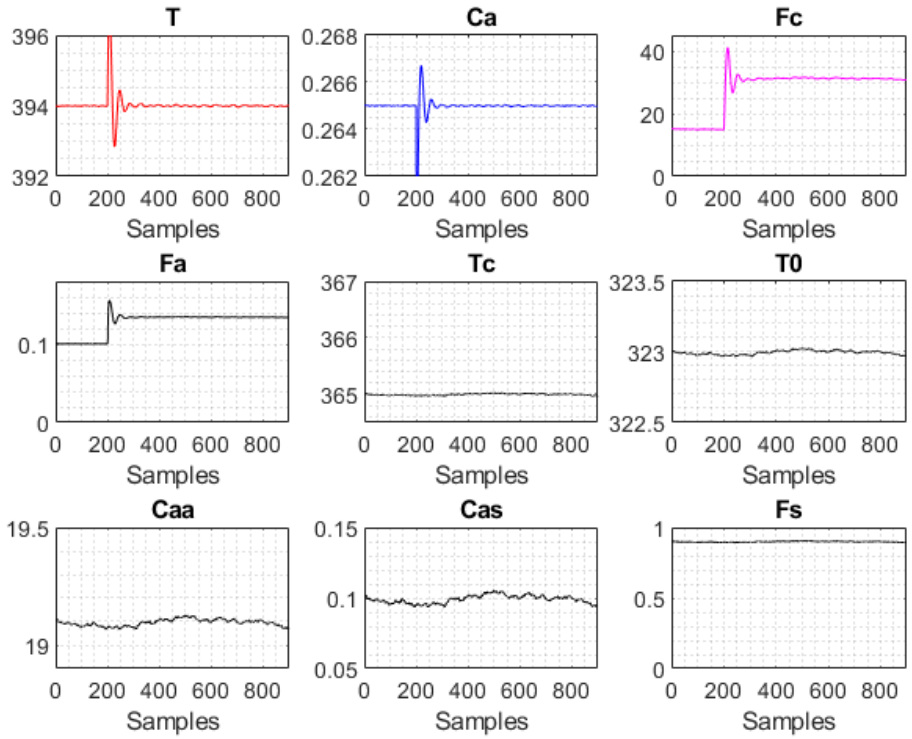

Figure 4.6 Measurements of process variables under Fault 5

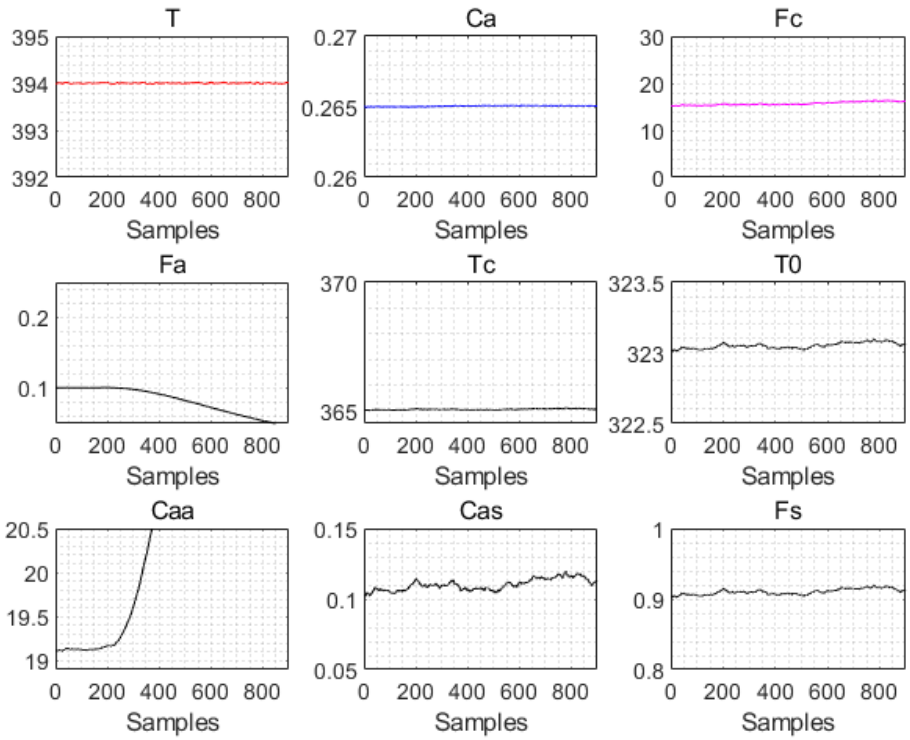

Figure 4.7 Measurements of process variables under Fault 6

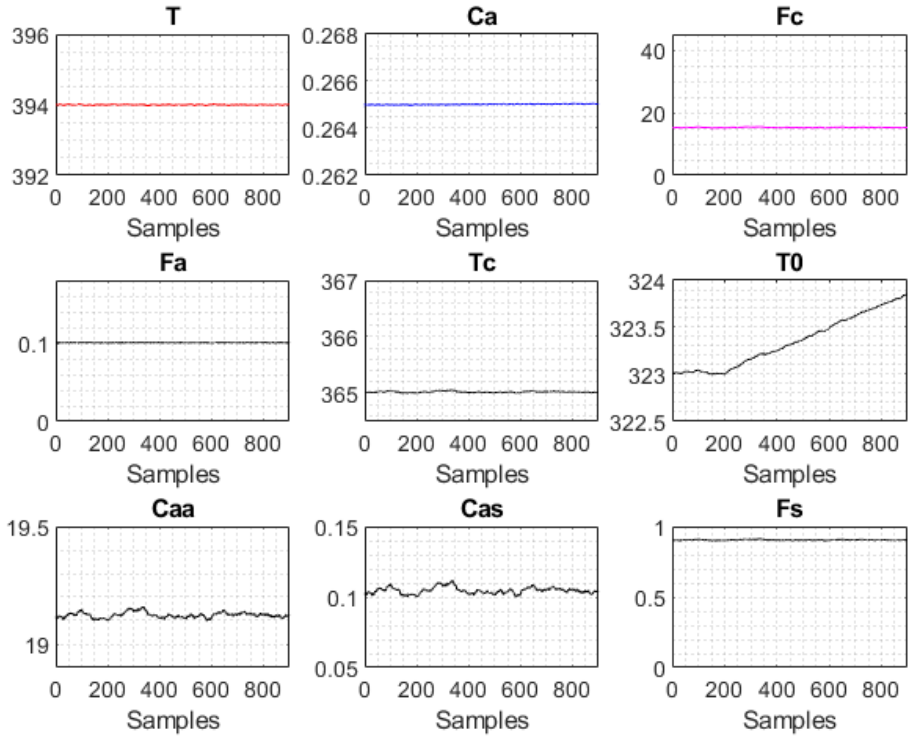

Figure 4.8 Measurements of process variables under Fault 7

The considered process faults can affect at least two process variables, as shown by the monitoring plots of the process variables presented above. Despite the fact that introduced faults could be identified by examining the classical univariate plots, this method has several limitations, such as its inability to capture the correlation between multiple process variables, its inability to detect incipient faults or anomalies, and a lack of scalability and adaptability. Hence, PCA-based fault detection and reconstruction-based fault diagnosis methods are used in this study to enhance fault diagnosis performance.

Pre-processing of the data is required prior to performing PCA. The data collected under normal process operation conditions, comprising of 900 samples with 9 process variables, is arranged in a data matrix with each variable scaled to zero mean and unit variance. The PCA-based fault detection method relies on the principle that normal data should encapsulate normal variations to accurately represent the common cause variation of a process. PCA aims to identify the directions, or PCs, which exhibit the most significant variability. These components enable a more concise description of the covariance structure of the normal data, making the selection of the right number of PCs crucial for both detection and diagnostic performance. The appropriate number of PCs is a critical model parameter, and several methods are available for its determination, including scree plot, Kaiser criterion, cross-validation, average eigenvalue, imbedded error function, and cumulative percent variance (CPV), among others (Berbache et al., 2019). In most applications, the CPV index is the most widely utilized method for selecting the optimal number of PCs due to its effectiveness and simplicity of implementation.

CPV is calculated from the eigenvalues of the covariance matrix, which are identical to the principal component variances of the data matrix. CPV is expressed as:

$$CPV = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \tag{4.33}$$

where, k represents the number of PCs and λ_i represents the sample variance of the i^{th} principal component. CPV is used to represent the proportion of the total data variation explained by the first k PCs, hence providing a measure of how effectively these components capture the underlying data variation. Figure 4.9 and Table 4.3 show the explained variance of each principal component, as well as the

cumulative explained variance, obtained after using PCA on the normal operating condition data, and it can be concluded that with three PCs we can capture over 99% of the variance.

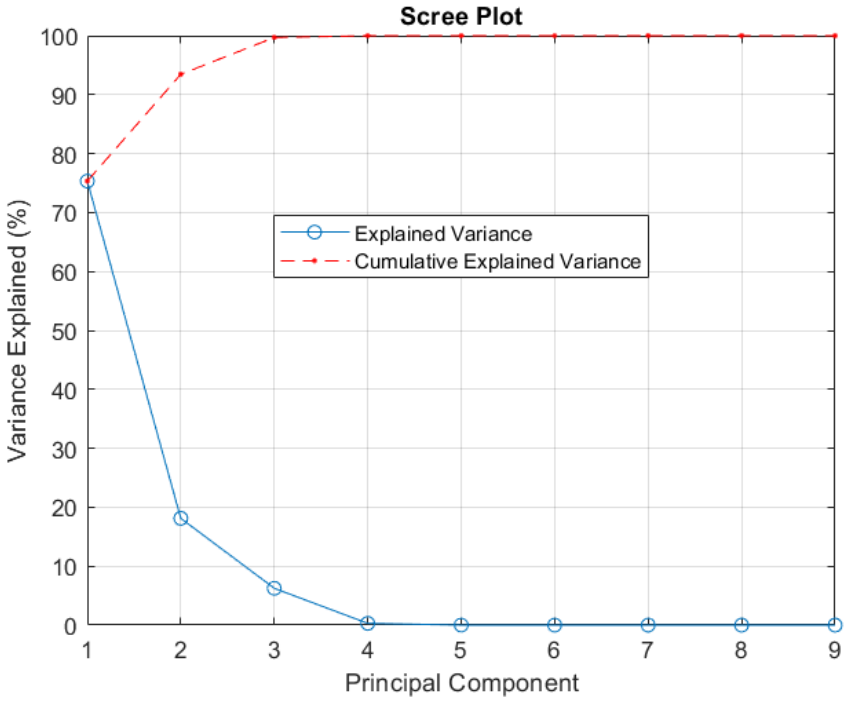

Figure 4.9 Explained and cumulative explained variances for PCA

Number of Principal Components	1	2	3	4	5
CPV (%)	75.345	93.451	99.694	99.995	100

Table 4.4 Explained variance of each principal component

In general, a higher CPV threshold indicates that the PCs capture more of the total data variation. However, we cannot arbitrarily set the CPV threshold to 99%, as this could capture noise in the data. There is not universally accepted CPV threshold, but some general rules can be used as a guidance. Choosing a threshold that explains between 95% and 99% of the total data variance, for instance, guarantees that most of the key information is retained while reducing the dimensionality of the data. Another strategy involves incrementally adding principal components and observing the additional variance they explain. This process is continued until the increase in explained variance by adding an additional principal component

becomes marginal. This marginal increase can be quantitatively defined as a threshold, for example, when the addition of a new principal component contributes less than 1% to the cumulative explained variance. This threshold is a balance between capturing most of the variability in the data and avoiding the inclusion of components that mainly capture noise or redundant information. Setting this threshold helps in determining the optimal number of principal components, ensuring each added component contributes significantly to the model's performance without overcomplicating the model (Abdi and Williams, 2010). The exact percentage can vary based on the dataset characteristics and the specific requirements of the analysis.

The aim of this study is to improve the reconstruction efficiency with as much accuracy as possible, and later in this section, the appropriate CPV threshold will be discussed according to the impact of different numbers of PCs on reconstruction performance. Here, the threshold for CPV is first tentatively set at 99%, which would suggest that the number of PCs is chosen to be three.

With the above information, fault detection can be achieved by comparing the monitoring statistics with their control limits. As the incipient faults have a more pronounced trend in fault magnitude, the results for faults 1, 2, 4 and 6 are given in Figures 4.10 to 4.13 respectively.

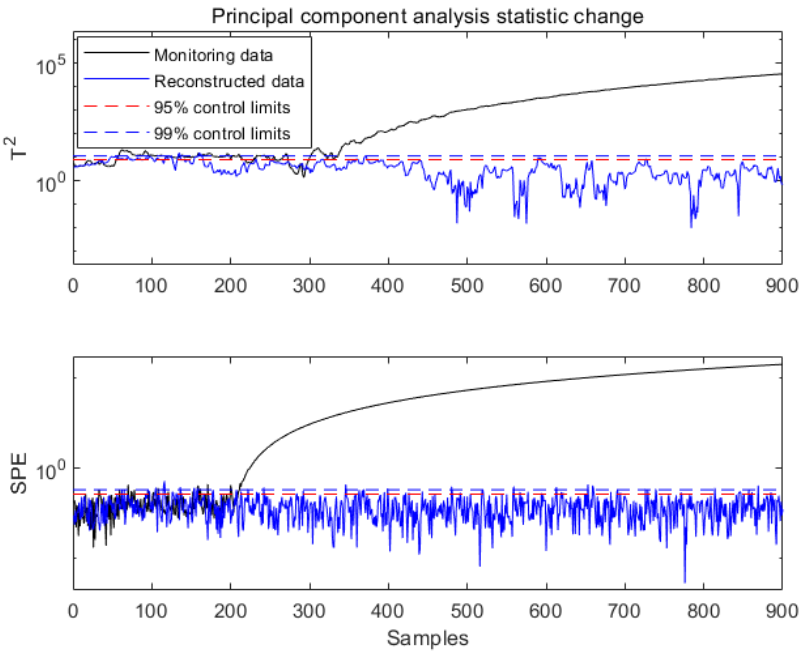

Figure 4.10 Monitoring statistics of process data and reconstructed normal data under Fault 1

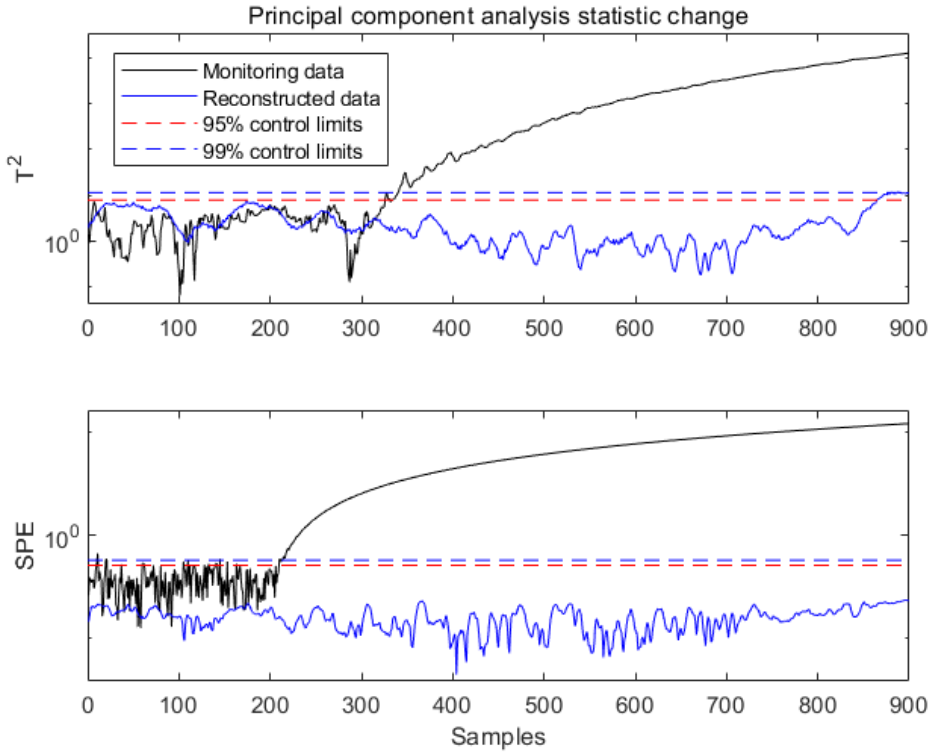

Figure 4.11 Monitoring statistics of process data and reconstructed normal data under Fault 2

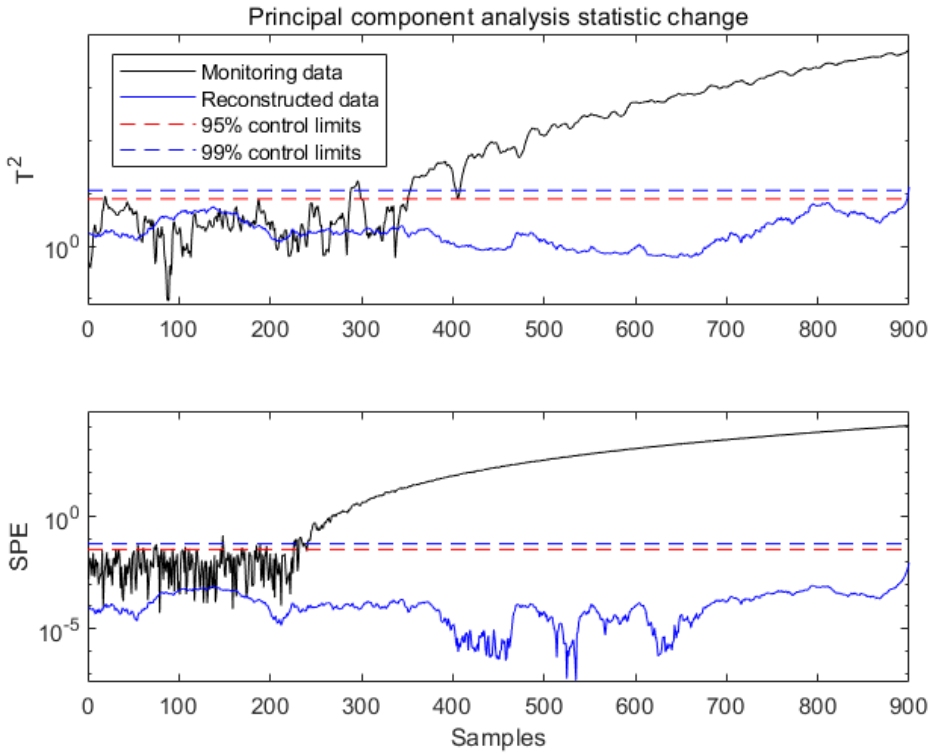

Figure 4.12 Monitoring statistics of process data and reconstructed normal data under Fault 4

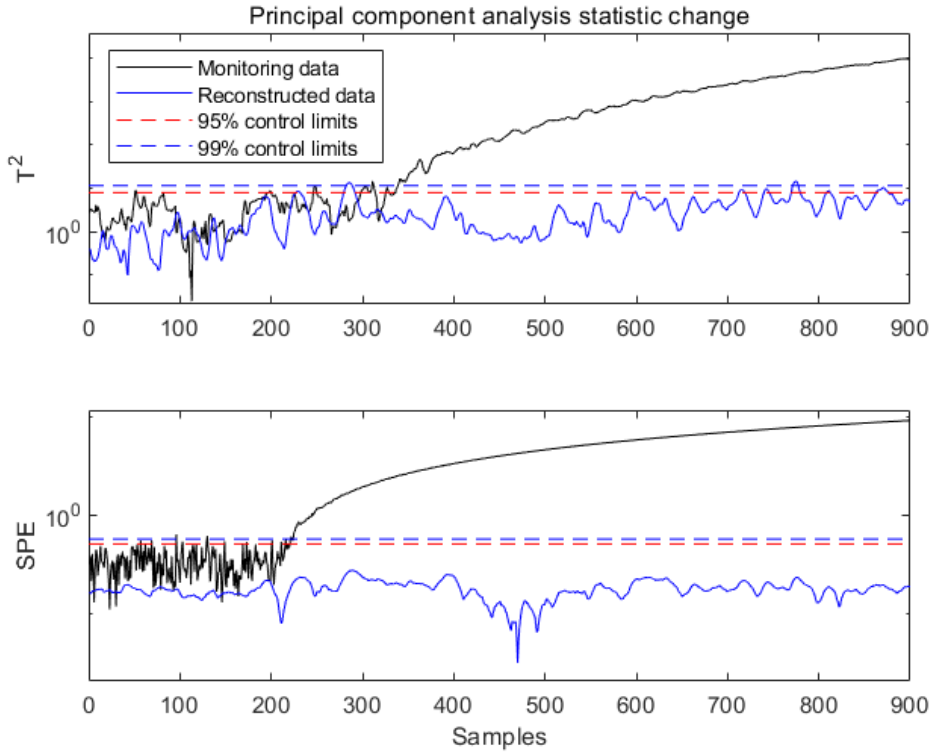

Figure 4.13 Monitoring statistics of process data and reconstructed normal data under Fault 6

In these figures, the black curves represent monitoring statistics for the corresponding faults, the blue curves represent the monitoring statistics after fault reconstruction, and the red and blue horizontal dashed lines represent the 95% and 99% control limits respectively. As mentioned previously, all faults were introduced at the 200th sampling point and as can be seen from the detection times obtained for different faults case given in Table 4.4, both statistics can detect the occurrence of faults, but the SPE is more sensitive to incipient faults compared to T^2, as it measures the change in correlation between the variables in the sampled data.

Fault type	Detection time with T^2 (samples)	Detection time with SPE (samples)
Fault 1	113	9
Fault 2	122	18
Fault 4	132	9
Fault 6	106	39

Table 4.5 Detection time for different faults

Once faults are detected, PCA models are built by using historical fault data and these PCA models for various faults are used to extract fault direction matrix $\tilde{\Xi}_{ip}$ according to equations (4.25) - (4.31). A reconstruction of the process data affected by the fault is then achieved by estimating the normal data using a fault direction matrix. If the statistics of the reconstructed normal data are below the control limits, then the corresponding fault is considered as the fault occurred in the process. If the statistics of the reconstructed normal data are above the control limits, then the corresponding fault is not the correct fault and reconstruction is continued using the next fault direction matrix until the statistics fall below the control limits. The solid blue lines in Figures 4.10 to 4.13 represent the statistics of the normal data estimated by the reconstruction, which are below the control limits, indicating the correct faults being diagnosed.

However, it can be observed that the monitoring statistics of the reconstructed normal data do not particularly follow the actual data prior to the onset of the fault. This may be because, firstly, the CSTR process is a dynamic process that is influenced by various interacting factors, such as reaction kinetics, mass and heat transfer, and, secondly, it may be that the random noise introduced into the simulation to simulate the real process results in the process not being completely consistent from run to run, thus affect the quality of the reconstructed normal data.

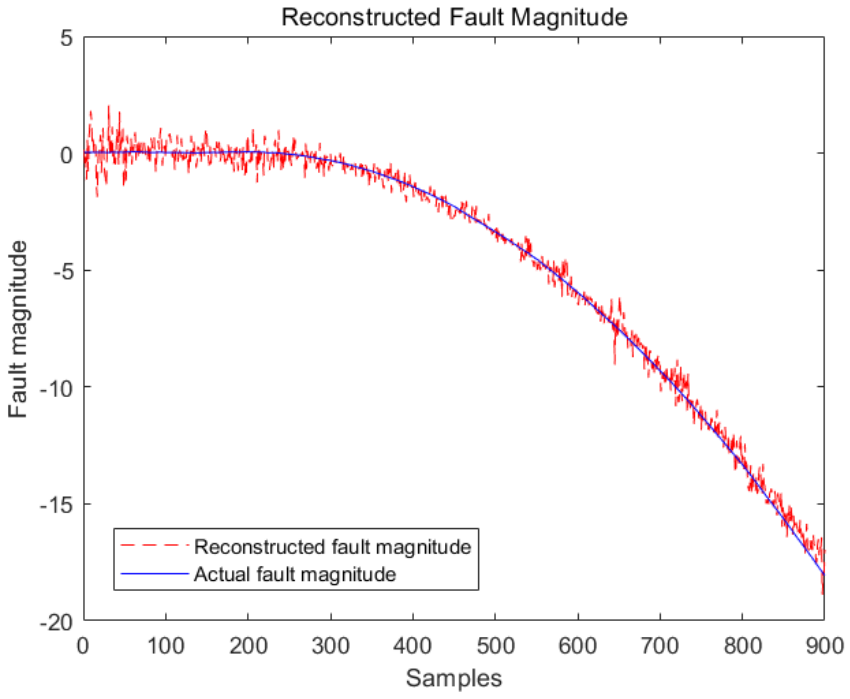

Figure 4.14 Comparison of estimated and actual fault magnitude for Fault 1

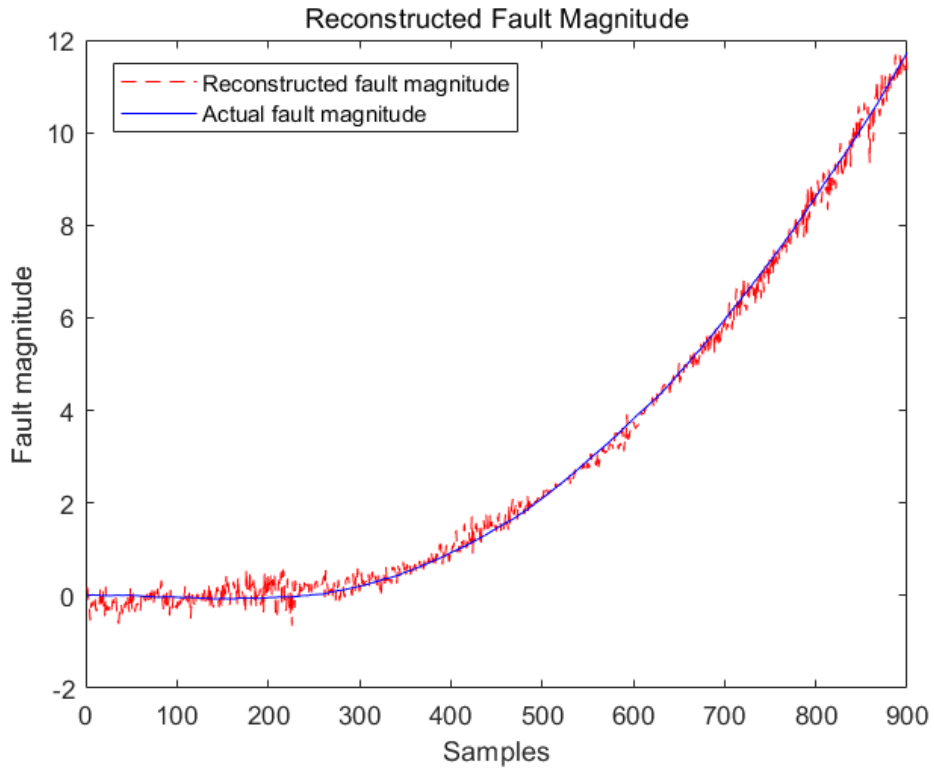

Figure 4.15 Comparison of estimated and actual fault magnitude for Fault 2

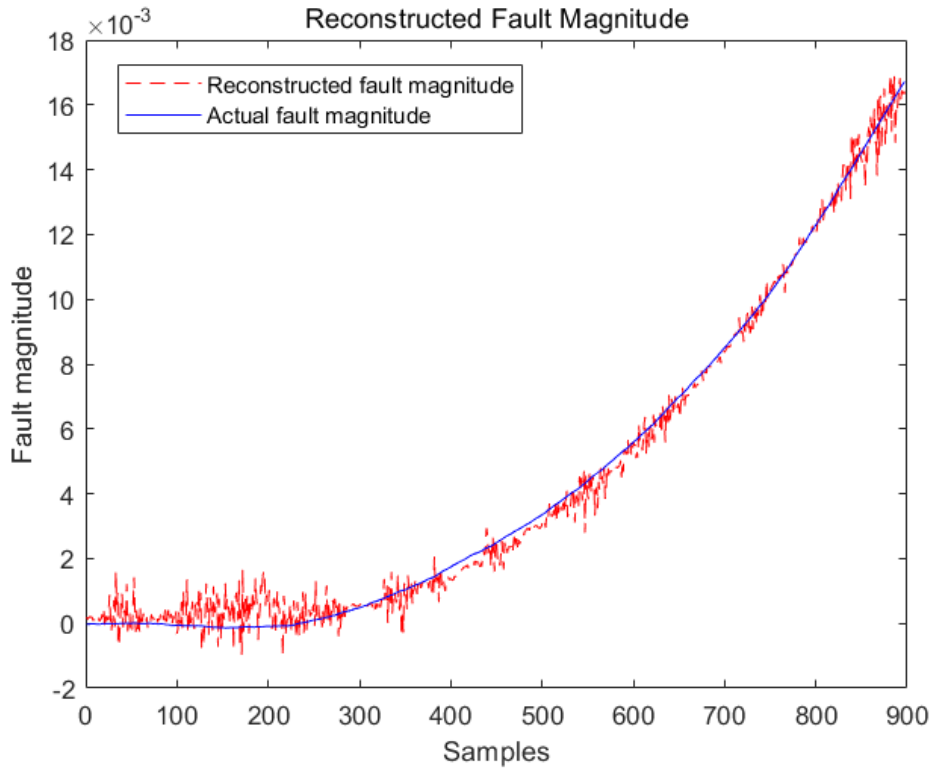

Figure 4.16 Comparison of estimated and actual fault magnitude for Fault 4

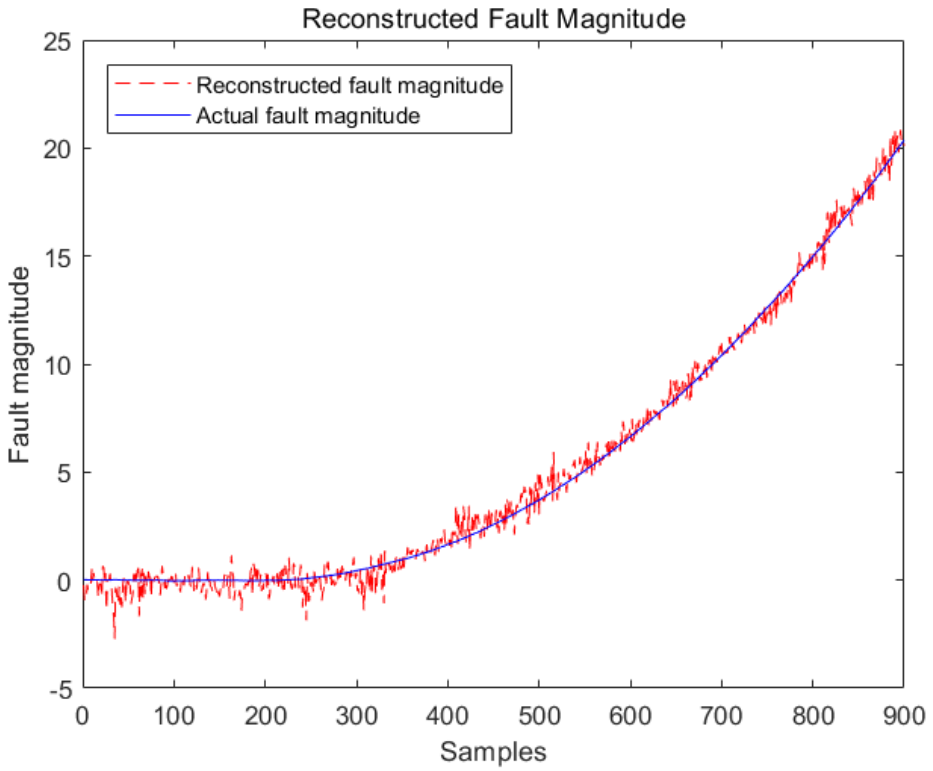

Figure 4.17 Comparison of estimated and actual fault magnitude for Fault 6

Meanwhile, further estimation of the fault magnitudes is performed based on the reconstruction method, as shown in Figures 4.14 to 4.17, where the blue and red line represent the actual and reconstructed fault magnitude respectively. The actual fault magnitude was determined by comparing the measurements taken at the time of the fault with the expected measurements under normal operation conditions. Specifically, the expected measurements were generated using our process model under normal operation conditions, serving as a reference point. Although there is some fluctuation in the estimated magnitude of faults, the trend of the estimated magnitude of faults remains consistent with the actual trend. With the results presented above, it can be demonstrated that the PCA-based approach enables the detection of process faults and the estimation of fault magnitude for diagnosis, followed by a discussion of the effect of the selection of the principal components of the PCA model on the diagnostic performance of the model. The reconstruction-based fault diagnosis relies on the fault direction matrix, which is derived from the projection of the fault data to the residual subspace, and the number of PCs is a crucial factor in determining the dimension of the PCA subspace.

Choosing more PCs can capture a larger percentage of the total variance in the fault data, which can certainly better represent the fault. However, using too many PCs

may introduce the effect of noise and thus reduce the fault magnitude estimation accuracy, whereas using too few PCs would reduce the model's sensitivity to noise in the data, allowing it to focus more on the variation caused by the faults, but may result in a loss of information in the data. To clarify the fault magnitudes designated as M1 to M5 represent varying degrees or scales of faults in the industrial process. Specifically:

- M1: Represents minor deviations or the earliest stage of a fault.
- M2: Indicates a fault of low to moderate severity, with noticeable impact on the process.
- M3: Represents a moderate to high severity fault, significantly impacting process performance.
- M4: Designates a high severity fault, leading to major process disruptions.
- M5: Indicates a critical fault level, where the process is at a near-failure or failure stage.

Fault 1	M1	M2	M3	M4	M5	Average MSE
Five PCs	0.0400	0.2084	0.5903	0.6861	1.7601	0.65577
Four PCs	0.0628	0.5992	0.6234	0.6524	2.3199	0.85154
Three PCs	0.2482	0.6806	**0.5853**	**0.5949**	**1.5652**	0.73484
Two PCs	0.1090	0.4944	0.6169	0.6431	1.9294	0.63511

Table 4.6 MSE of the estimated fault magnitude for fault 1

Fault 2	M1	M2	M3	M4	M5	Average MSE
Five PCs	0.0343	0.0448	0.0445	0.0547	0.0635	0.04436
Four PCs	0.0666	0.1275	0.0461	0.0590	0.0617	0.07218
Three PCs	0.0323	0.1301	**0.0435**	**0.0467**	**0.0590**	0.06232
Two PCs	0.0265	0.2203	0.045	0.0238	0.0699	0.0771

Table 4.7 MSE of the estimated fault magnitude for fault 2

Fault 4	M1	M2	M3	M4	M5	Average MSE
Five PCs	$1.79*10^{-7}$	$2.26*10^{-7}$	$5.82*10^{-7}$	$6.14*10^{-7}$	$5.14*10^{-7}$	$3.8300*10^{-7}$
Four PCs	$3.95*10^{-7}$	$4.95*10^{-7}$	$5.76*10^{-7}$	$6.56*10^{-7}$	$6.56*10^{-7}$	$5.5560*10^{-7}$
Three PCs	$4.04*10^{-7}$	$5.62*10^{-7}$	**$5.41*10^{-7}$**	**$4.43*10^{-7}$**	**$4.89*10^{-7}$**	$5.0780*10^{-7}$
Two PCs	$3.07*10^{-7}$	$4.20*10^{-6}$	$4.36*10^{-7}$	$4.76*10^{-6}$	$5.28*10^{-6}$	$2.9966*10^{-6}$

Table 4.8 MSE of the estimated fault magnitude for fault 4

Fault 6	M1	M2	M3	M4	M5	Average MSE
Five PCs	0.0266	0.0131	0.0587	0.4630	0.5361	0.1955
Four PCs	0.0490	0.0743	0.0672	0.4526	0.6548	0.25958
Three PCs	0.0197	0.0436	**0.0498**	**0.4025**	**0.4999**	0.2710
Two PCs	0.0692	0.1444	0.1099	0.4947	0.5883	0.2813

Table 4.9 MSE of the estimated fault magnitude for fault 6

Tables 4.5 to 4.8 give the mean squared error (MSE) between the estimated fault magnitude and the actual fault magnitude for faults 1, 2, 4 and 6. For each fault case, PCA models were constructed for each of the five different magnitude data (M1-M5) to examine the accuracy of the reconstruction using different numbers of PCs.

The magnitudes of the faults increase progressively from M1 to M5 in the table above. As the magnitudes of the faults increase, the estimation accuracy might decrease due to various factors, including the complexity of the underlying relationships, increased noise in the data, or non-linearity of the data. It is challenging to determine the optimal number of PCs from the smaller fault magnitudes (M1 and M2), nonetheless, it can be observed that the MSE values calculated using three PCs (in bold) are consistently lower for larger fault magnitudes (M3, M4, and M5) than those calculated using other numbers of PCs. This would suggest that the selection of three principal components, aiming for a 99% CPV threshold, was intended to achieve a balance between capturing significant data variation and minimizing the impact of potential noise in fault cases.

This threshold aligns with standard PCA practices, ensuring that major data features are included while disregarding minor fluctuations possibly representing noise or irrelevant variations. However, the point raised about white noise contamination merits consideration. White noise may influence the variance captured by PCs. In situations where measurements include white noise, the required number of PCs for achieving the same CPV could increase, potentially affecting PCA model performance.

Acknowledging that this study assumed relatively clean measurement data, it's pertinent to note the inevitability of measurement noise in practical scenarios. Future research should, therefore, examine how different noise levels, particularly white noise, influence PC selection in PCA. Conducting sensitivity analyses to assess the impact of white noise on CPV and PCA performance would provide valuable insights, leading to more robust criteria for PC selection under various noise conditions.

4.6 Conclusion

This chapter presents a comprehensive study of the application of PCA-based process fault reconstruction methods. Most previous research has focused on reconstruction methods for diagnosing sensor faults in systems without feedback control and less on process faults. In this study, the diagnosis of individual process faults affecting multiple variables and the extraction of fault magnitudes is achieved by improving the extraction method of the process fault direction matrix. This chapter demonstrates the feasibility and effectiveness of the proposed method by simulating multiple fault types in CSTR processes.

Fault detection is achieved by comparing the monitoring statistics with their control limits. The results show that the SPE monitoring statistics allow the detection of faults at a relatively early stage, making the method more reliable for the detection of incipient faults than directly observing variable changes from a classical univariate chart. After a fault has been detected, fault identification is accomplished by extracting fault direction matrices from historical fault data and reconstructing normal data. The results suggested that the proposed method could effectively identify fault types and provide estimations of fault magnitudes. Although there is some fluctuation in the estimated magnitudes due to noise and imperfect representation of the actual fault behaviour, the trend in fault magnitudes is not

overly biased and can still be used in the next prediction stage. In addition, this chapter emphasises the importance of selecting optimal number of PCs (CPV threshold) balancing the need to capture a larger percentage of the total variance in fault data while minimizing the impact of noise on fault magnitude estimation accuracy.

Overall, this chapter has demonstrated that the proposed method can be a valuable tool for diagnosing process faults. However, there are some parts that need to be improved, such as the fluctuations in the estimated magnitudes. First, this may be because the fault direction matrix was extracted from historical fault data and may not perfectly represent the actual fault behaviour of the process. Secondly, it may be that the actual process behaviour is masked by the random noise introduced in the simulation, making fluctuations in the estimated fault magnitudes likely to come from variations in the noise, affecting the accuracy of the fault magnitude estimates. Further research could explore ways to improve the generalisability of the models by increasing the variety of historical faults to refine the fault direction matrix, optimising the selection of principal components, or incorporating other advanced techniques to improve fault detection and reconstruction performance.

Chapter 5 Fault Magnitude Prediction based on Autoregressive Model and Extreme Learning Machine

5.1 Introduction

The focus of research on fault prognosis varies across four main stages: the fitting and prediction of the fault development trend, the implementation of early warnings for faults, the computation of the RUL of equipment, and the establishment of a fault prognosis and health management system. This thesis primarily concentrates on the prediction of fault magnitude trends. Central to this research are fault prognosis techniques, which play a pivotal role in anticipating the potential future states of the system. These techniques involve extracting information about the possible development of system faults in future time periods, primarily based on historical data. Typically, this extracted information is consolidated and formulated into a fault prognosis model. With the acceleration of advancements in data acquisition, network communication, and data storage technologies, a surfeit of data is continuously amassed in the process industry. A significant portion of this data comprises high-dimensional, non-linear, and non-Gaussian time series data, spurring the widespread application of statistical and neural network time series prediction methods in industrial fault prognosis.

The commonly used statistical methods are the autoregressive (AR) model and the autoregressive moving average (ARMA) model. These two models are mainly applied to stationary linear time series. In (Yan *et al.*, 2004), an ARMA model is integrated with a logistic regression analysis based on a maximum likelihood estimate to evaluate the equipment condition and RUL, and the results show that the equipment failure process may be dynamically anticipated and updated. While traditional methods like the AR and ARMA models are effective for short-term, stationary time series prediction, they often fall short with non-linear or non-stationary industrial process data. In contrast, neural network algorithms, despite their inherent complexity, offer a more robust solution for such data. Their ability to model complex, non-linear relationships makes them extensively used in industrial fault prognosis. It's important to note that while neural networks are complex, their adaptability and advanced learning capabilities outweigh the challenges posed by their complexity, especially in the context of intricate industrial datasets.

Neural networks are pivotal in fault prediction due to their unique advantages. These include their superior nonlinear fitting ability, essential for modelling complex systems, and their simple yet effective learning rules. This makes them particularly suited for scenarios where traditional linear models fail. Among neural networks, back propagation (BP) networks are the most widely used, but BP networks suffer from long training time and can fall into local minima during training. In (Huang *et al.*, 2006), a new single hidden layer feedforward neural network learning algorithm, generally known as ELM, is proposed. In an ELM, the connection weights between the input layer and the hidden layer, as well as the biases of the hidden layer, are randomly set and not changed during training. The output layer weights are obtained analytically in one step based on the training data. Thus, ELM can be trained very quickly and can have better generalization performance compared to some classical neural networks. In (Yang *et al.*, 2016b), an ELM-based method for RUL prediction is proposed and compared with BP artificial neural networks. It is shown that the ELM-based model is slightly inferior to the BP artificial neural network-based model in terms of prediction accuracy and stability, but it can significantly reduce the training time.

In this chapter AR models are developed as a baseline for comparison, and to cope with the non-linearity of many industrial processes, ELM models are proposed in this chapter for fault prognosis. These AR and ELM models can be trained using time series data of fault magnitudes. However, such time series data for fault magnitudes are usually not available in industry. Therefore, the fault reconstruction approach described in Chapter 3 can be used to restructure fault magnitude data for the development of fault prognosis models.

This chapter is organised as follows. Section 5.2 presents an overview of fault prognosis strategies, including direct and iterative method. Section 5.3 gives details of the establishment of the AR and ELM based fault magnitude prognosis models. Section 5.4 shows the fault magnitude prognosis results for the AR and ELM models and the corresponding analysis. Finally, Section 5.5 summarises the conclusions from this chapter.

5.2 Fault Magnitude Prediction Strategies

5.2.1 Single-Step prediction

For a fault magnitude varying with time t, the discrete set composed of its observations constitutes a time series $X = \{x_t | t = 1, 2, \cdots, N\}$, where N is a positive integer representing the total number of observations in the time series and x_t donates the observations at time step t. The aim of time series prediction is to estimate unknown future states based on past and present observations. For one-step ahead prediction, the model can be expressed as $\hat{x}_{t+1} = f(X)$, where $\hat{x}_{t+1}$ represents the predicted value at the time $t + 1$ and f symbolizes the prediction model. While one-step predictions provide immediate insights into the near future, they have limitations in capturing long-term trends. Therefore, the exploration of multi-step ahead prognosis methods becomes essential as it offers a broader perspective on future trends, which is crucial for effective process management (Hu *et al.*, 2005).

5.2.2 Multi-Step Ahead Prediction

For industrial processes requiring longer-term fault trend analysis, multi-step ahead prognosis becomes crucial. Common approaches for multi-step prognosis strategies include the direct method and the iterative method (Liu *et al.*, 2016b).

1. Direct Method

The direct method was proposed by Cox (Cox, 1961), with this method, a distinct prediction model is developed for each prediction time step, which means *H* different models are used to predict h steps ahead. (Franses and Legerstee, 2009)

Assume the prediction step size of multi-step prediction is h, then h distinct prediction models would need to be developed. The first step in this process involves partitioning the original observed series into h subsets. Each subset is then used to train a corresponding model denoted as f_h:

$$y_{t+h} = f_h(x_t) \tag{5.1}$$

where, $x_t \in \{x_t, x_{t-1}, \cdots, x_{t-d+1}\}, y_{t+h} = \{x_{t+1}, x_{t+2}, \cdots x_{t+h}\}$, $t \in \{d, d + 1, \cdots, N - h\}$, d is the embedding dimension of the input variable (the number of past values taken used to predicting a future value), N is the number of observations and $h \in \{1,2, \cdots, H\}$.

Next, after the learning process, a prediction model is built for each data subset, and the prognosis output of a multi-step prognosis with step size H can be expressed as:

$$\hat{x}_{t+h} = \hat{f}_h(x_t, x_{t-1}, \cdots, x_{t-d+1}) \quad h = 1,2, \cdots, H \tag{5.2}$$

where f_h represents the prognosis model built on the data subset. As can be seen from the above equation, the multi-step ahead prediction based on the direct method involves building separate models for each future time step. Consequently, there is no direct correlation between the models, and the predictions made by each model are independent, which gives this approach some limitations such as lack of correlation between forecast values, high functional complexity, and large computational time for multiple models.

2. Iterative Method

Compared with the direct method, the iterative prognosis method is simpler and more intuitive. The model prediction at time t+1 is fed back to the model input to further predict the model output at time t+2, and so on. A general d^{th} order time series model can be represented as:

$$\hat{x}_{t+1} = f(x_t, x_{t-1}, \cdots, x_{t-d+1}) \tag{5.3}$$

Then, the multi-step ahead prognosis based on the iterative method is given as:

$$\hat{x}_{t+h} = \begin{cases} f(x_t, x_{t-1}, \cdots, x_{t-d+1}) & if \;\; h = 1 \\ f(\hat{x}_{t+h-1}, \cdots, \hat{x}_{t+1}, x_t, \cdots, x_{t-d+h}) & if \;\; h \in \{2, \cdots, d\} \\ f(\hat{x}_{t+h-1}, \cdots, \hat{x}_{t+h-d}) & if \;\; h \in \{d+1, \cdots, H\} \end{cases} \tag{5.4}$$

where, f represents the prognosis model, t is the discrete time, $h \in \{1,2, \cdots, H\}$is the prediction horizon of multi-step prediction, and d is the embedding dimension or model order.

Compared with the direct method, the recursive approach to multi-step ahead prediction can be considered better than the direct approach, mainly due to its efficiency in requiring only one model for all prediction horizons. This advantage provides significant savings in computational resources, training time and model maintenance, as well as scalability and model consistency across time steps. Furthermore, when compared to the direct method, where the best model must be selected and refined at each time step, the recursive approach streamlines the model selection process. Therefore, the iterative methods are selected as the multi-step prediction strategy here.

5.3 Fault Magnitude Prediction

5.3.1 Fault Magnitude Prediction based on AR Model

The fault magnitude prognosis in this section is based on the fault magnitude data obtained from fault reconstruction in the previous chapter. After data pre-processing, model order selection and parameter estimation, a suitable AR model can be obtained, which can then be used to predict future fault magnitudes. The L-step prognosis of the AR(q) model then refers to the estimation of future values at moment $t+L$ based on actual observations at the current moment t and previous moments.

The steps for the L-step prognosis are to first calculate the prediction for step L=1, then calculate the prediction for $L=2$ based on that prediction, and so on, until the prediction for step L is found, with the following recursive process:

$$\begin{cases} \hat{x}_t(1) = \varphi_1 x_t + \varphi_2 x_{t-1} + \cdots + \varphi_q x_{t-q+1} \\ \hat{x}_t(2) = \varphi_1 \hat{x}_t(1) + \varphi_2 x_t + \cdots + \varphi_q x_{t-q+2} \\ \vdots \\ \hat{x}_t(q) = \varphi_1 \hat{x}_t(q-1) + \varphi_2 \hat{x}_t(q-2) + \cdots + \varphi_q x_t \\ \vdots \\ \hat{x}_t(L) = \varphi_1 \hat{x}_t(L-1) + \varphi_2 \hat{x}_t(L-2) + \cdots + \varphi_q \hat{x}_t(L-q) \ \ (L > q) \end{cases} \tag{5.5}$$

where $\hat{x}_t(i)$ $(i = 1,2,\cdots,L)$ is the predicted value, φ_i $(i = 1,2,\cdots,q)$ is the model parameter, and x_t is the historical observation.

The one-step ahead prognosis from the AR model is an estimate of the fault magnitude value at moment $t+1$ based on the actual observation at the current moment t and past moments $x_t, x_{t-1}, x_{t-2}, \cdots$. Then, in the iterative multi-step prediction, a single model is trained to predict one step ahead, and then the predicted value is recursively fed back into the model to generate prediction for further prediction horizons.

First, the assumed maximum order q_{max} of the AR model is set and the matrices X and Y are obtained according to equation (3.28).

$$X = \begin{bmatrix} x_q & x_{q-1} & \cdots & x_1 \\ x_{q+1} & x_q & \cdots & x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_{N-2} & \cdots & x_{N-q} \end{bmatrix}$$
$$Y = \begin{bmatrix} x_{q+1} \\ x_{q+2} \\ \vdots \\ x_N \end{bmatrix} \tag{5.6}$$

where, $q = 1,2,\cdots q_{max}$. After which the corresponding matrix of model coefficients $\hat{\varphi}$ at order q is calculated by (3.30) and the mean square deviation σ can be calculated according to the following equation:

$$\sigma^2 = E\left(x_t - \sum_{j=1}^{p} \varphi_j x_{t-j}\right)^2 \tag{5.7}$$

The appropriate model order q can then be determined according to the AIC criterion, and finally the predicted data can be obtained according to equation (3.25). The training progress of AR model is shown in Figure 5.1.

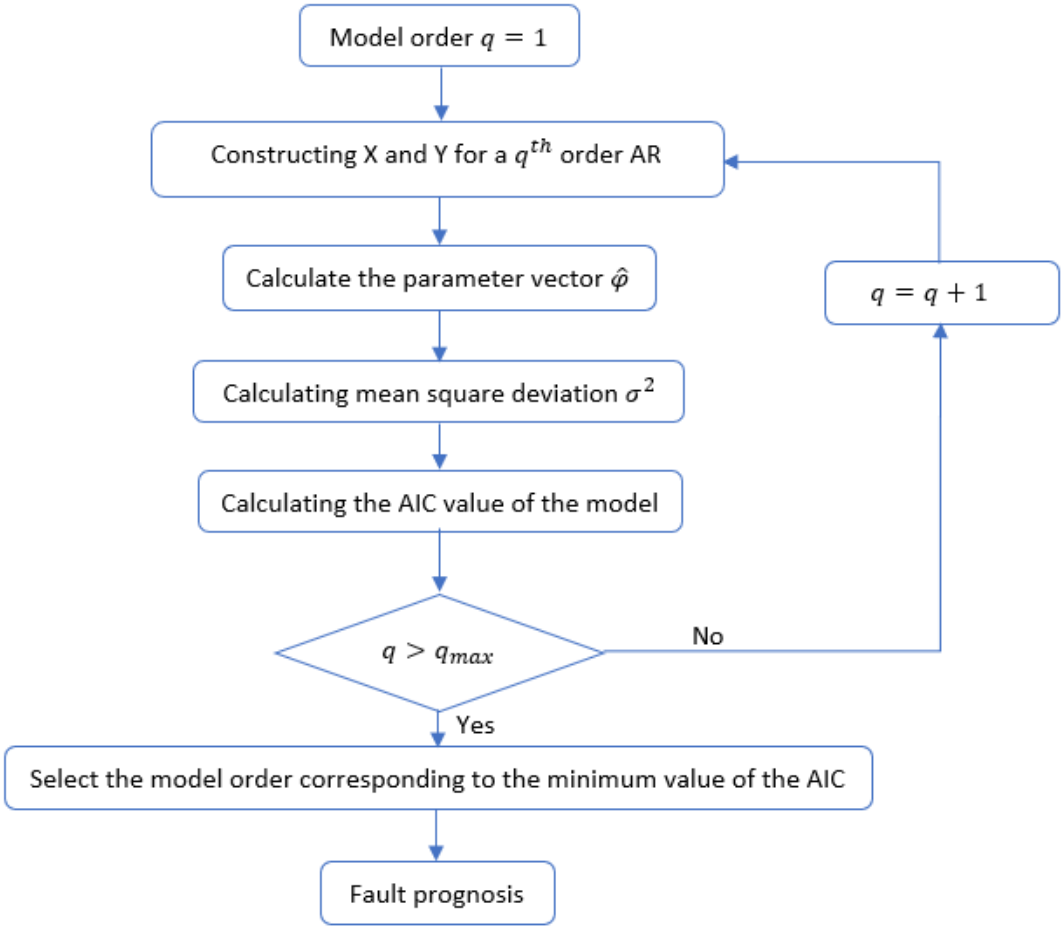

Figure 5.1 AR model algorithm flow

5.3.2 Fault Magnitude Prediction based on ELM

When applying ELM to prognosis, the correct selection of model order and model parameters is crucial to the outcome of the prediction. The model order of the nonlinear time series model refers to the number of past values used in the model inputs, while the model parameters include the network weights and the number of

hidden layer nodes. Appropriate selection of model order and model parameters will enable effective prediction of future fault magnitudes.

The ELM modelling process is as follows:

1. The dataset was first scaled to zero mean and unit variance. It was then divided into training, testing and validation sets. Using the training set to set up the input matrix and the corresponding desired output matrix. For a sequence of fault magnitudes with N samples, assuming the input sequence dimension (model order) is q, then the input and output matrices can be expressed as follow:

$$X = \begin{bmatrix} f_1 & f_2 & \cdots & f_q \\ f_2 & f_3 & \cdots & f_{q+1} \\ \vdots & \vdots & \vdots & \vdots \\ f_{N-q} & f_{N-q+1} & \cdots & f_{N-1} \end{bmatrix} \quad Y = \begin{bmatrix} f_{q+1} \\ f_{q+2} \\ \vdots \\ f_N \end{bmatrix} \tag{5.8}$$

2. Set the number of neurons in the hidden layer and their activation function.
3. Randomly generate the input and hidden layer connection weights and bias.
4. Calculate the hidden layer output matrix based on equations (3.34) and (3.35).
5. Calculate the optimal output layer weights of the network using equation (3.37).

The training process of the ELM neural network is convenient in terms of obtaining the network weights with fast-learning speed and the training accuracy is also relatively high.

To predict the fault magnitude value at the next time step, it is only necessary to substitute the relevant time series into the trained prediction model as described above.

5.4 Result and Analysis

Based on the reconstructed fault magnitudes given in chapter 4, it can be seen from Figures 5.16 and 5.19 that the fault magnitudes do not change significantly in the first half of the period and show a rapid upward trend from roughly the 400th sampling point, so we used the last 500 sampling points for the development of prediction models of the fault magnitudes data for different faults.

The training data set contains the reconstructed fault magnitude data at six different fault development speeds (6 × 500 = 3000 samples) and the testing data set contains

the reconstructed fault magnitude data at two different fault development speeds (2 × 500 = 1000 samples). The validation data set also contains the reconstructed fault magnitude data at one different fault development speeds (500 samples). Table 5.1 gives the fault developing speeds of these data sets.

	Fault 1	Fault 2	Fault 4	Fault 6
Training datasets	-0.12×10^{-6}	0.92×10^{-5}	3.09×10^{-6}	0.47×10^{-5}
	-1.03×10^{-6}	1.28×10^{-5}	4.22×10^{-6}	1.56×10^{-5}
	-2.36×10^{-6}	3.62×10^{-5}	5.13×10^{-6}	3.84×10^{-5}
	-3.39×10^{-6}	5.26×10^{-5}	6.54×10^{-6}	5.23×10^{-5}
	-4.66×10^{-6}	7.49×10^{-5}	7.92×10^{-6}	7.58×10^{-5}
	-6.42×10^{-6}	9.33×10^{-5}	8.15×10^{-6}	9.49×10^{-5}
Testing datasets	-5.45×10^{-6}	2.85×10^{-5}	2.01×10^{-6}	6.74×10^{-5}
	-7.14×10^{-6}	4.42×10^{-5}	1.14×10^{-6}	8.31×10^{-5}
Validation dataset	-8.28×10^{-6}	1.52×10^{-5}	0.33×10^{-6}	1.85×10^{-5}

Table 5.1 Fault magnitudes under different fault development speeds in the datasets

5.4.1 AR Model

As a single AR model can only predict for one time series, separate AR prediction models are required for each fault. To present the results more clearly, the AR models designed for the following incipient process faults, faults 1, 2, 4 and 6. These models are designed with orders ranging from 1 to 10, and their respective AIC values are shown in Table 5.2.

Model Order	AIC for Fault 1	AIC for Fault 2	AIC for Fault 4	AIC for Fault 6
1	-4.3929	-3.9274	-4.6603	-4.8069
2	-4.6995	-4.1856	-4.9353	-4.9993
3	-4.8228	-4.3143	-5.0470	-5.1034
4	-4.8972	-4.3343	-5.0597	-5.1277
5	**-4.9038**	-4.3371	-5.1086	-5.1351
6	-4.8998	-4.3597	-5.1058	**-5.1396**
7	-4.8990	**-4.3617**	-5.1404	-5.1363
8	-4.8941	-4.3584	-5.1378	-5.1361
9	-4.8913	-4.3535	**-5.1593**	-5.1314
10	-4.8914	-4.3499	-5.1585	-5.1299

Table 5.2 AIC values of AR (1) to AR (10) models for incipient process faults

It can be seen that the AR models constructed for faults 1, 2, 4 and 6 yield the smallest AIC determination values at order 5, 7, 9 and 6 (in bold), respectively. The optimal order for each AR model is the order corresponding to the smallest AIC value. The AR model parameters for faults 1, 2, 4 and 6 are then obtained from equations (3.19) - (3.30) as:

$$\hat{\varphi}_{F_1} = [0.1800 \quad 0.2320 \quad 0.2327 \quad 0.2529 \quad 0.1004]$$
$$\hat{\varphi}_{F_2} = [0.2455 \quad 0.1908 \quad 0.2297 \quad 0.0777 \quad 0.0274 \quad 0.1429 \quad 0.0814]$$
$$\hat{\varphi}_{F_4} = [0.2908 \quad 0.1862 \quad 0.2062 \quad 0.0186 \quad 0.1254 \quad 0.0055 \quad 0.2072 \quad 0.0278 \quad 0.0581]$$
$$\hat{\varphi}_{F_6} = [0.4444 \quad 0.1197 \quad 0.1898 \quad 0.1236 \quad 0.0325 \quad 0.0887]$$

Using these models, one-step ahead and multi-step ahead predictions of the fault magnitude of the testing data set can be made based on equations (5.3) and (5.4), and the results are shown in Figures 5.2 to 5.5, where the prediction results for the testing set corresponding to each fault are shown. In these figures, the solid black line represents the actual reconstructed fault magnitudes, the dashed blue line represents the one-step ahead predicted fault magnitudes, and the dotted red line represents the multi-step ahead predicted fault magnitudes.

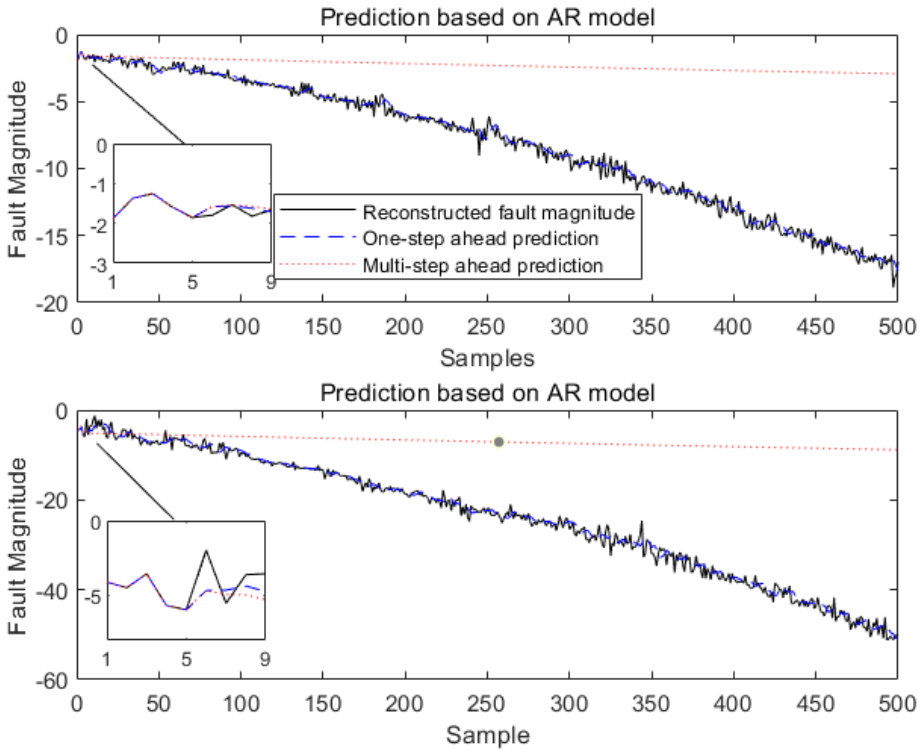

Figure 5.2 Prediction of fault magnitude for fault 1 based on the AR model

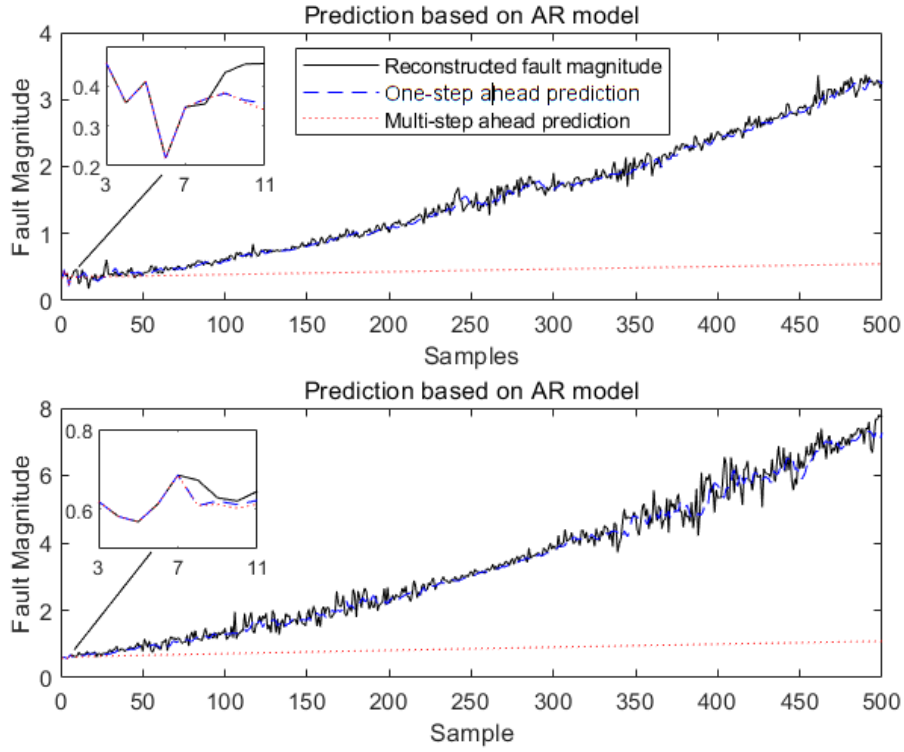

Figure 5.3 Prediction of fault magnitude for fault 2 based on the AR model

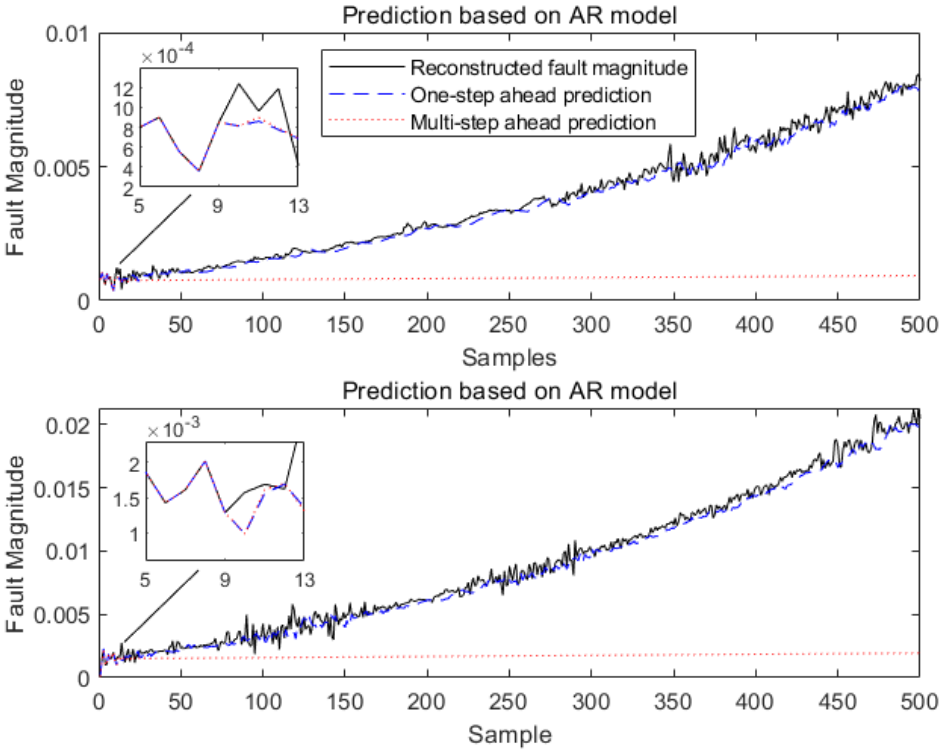

Figure 5.4 Prediction of fault magnitude for fault 4 based on the AR model

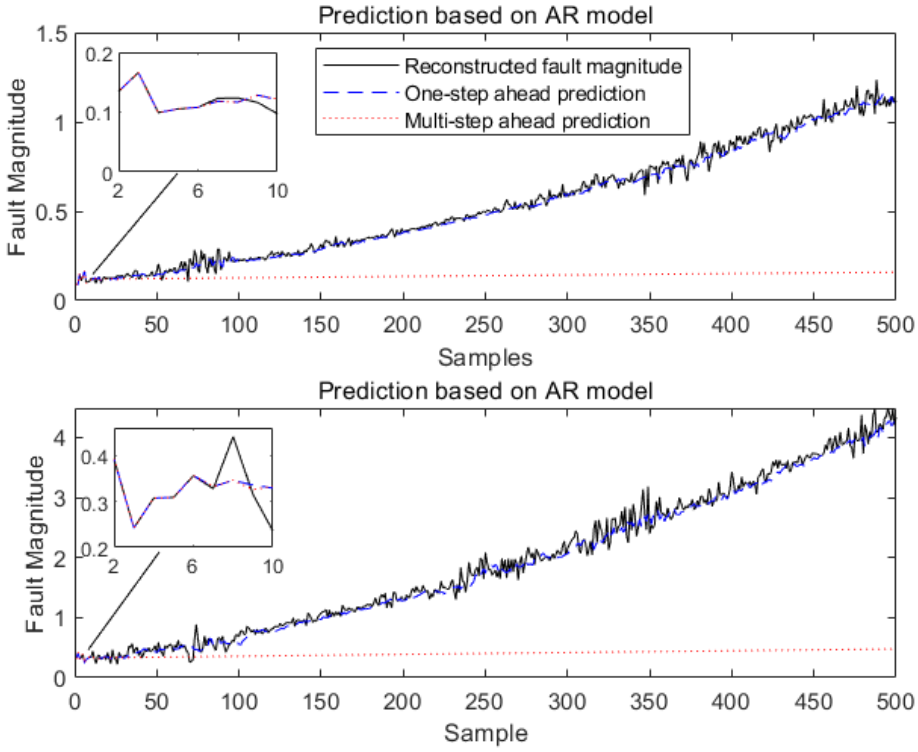

Figure 5.5 Prediction of fault magnitude for fault 6 based on the AR model

When using the AR model for time series prediction, the model uses a linear combination of past values to capture the linear dependence of the time series. As

can be seen from Figures 5.2 to 5.5, the one-step ahead prediction based on the AR model performs well for fault magnitude predictions of different magnitudes for various faults. This is because the one-step prediction values of the AR model are based on the closest observations to the current prediction moment, which can reasonably capture the dynamics of the time series over a short period.

However, for multi-step ahead prediction, the predictions are generated based on the model's own previous predictions. The process begins at a specific time step, determined by the optimal model orders obtained through the Akaike Information Criterion (AIC), as shown in Table 5.2. For instance, for fault type 1, with an optimal model order of 5, the prediction starts at time step 5, using the previous 5 time steps of historical data. The prediction for each subsequent time step is iteratively generated by combining both historical data and the model's own previous predictions. As the prediction moves further into the future, the predicted value becomes increasingly dependent on its own predictions and model coefficients rather than on past observations. This iterative process is consistently applied to different fault types, as determined by their respective optimal model orders. The ability to capture trends deteriorates progressively as the model inputs are entirely replaced by the predicted values. This is one of the reasons why the performance of the AR model declines sharply and exhibits a flat line after multiple prediction steps equal to the order of the model. Furthermore, when discussed from a data perspective, a fundamental assumption of the AR model is that the data is required to be stationary, meaning that the statistical properties of the data (mean, variance and autocorrelation) should remain constant over time. Failure magnitude data, on the other hand, are usually non-linear time series data with trend in nature, and even after they have been detrended and normalised, all non-stationary components may not be completely eliminated, which may also be the reason for the unsatisfactory multi-step prediction results. Owing to the subpar performance of the AR model for multi-step predictions, only its root means square error (RMSE) and mean absolute percentage error (MAPE) for one-step predictions on the testing sets are acquired here as shown in Tables 5.3 to exhibit the model's predictive capabilities.

The RMSE and MAPE are defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y(k) - \hat{y}(k)\right)^2}$$
$$MAPE = \frac{100}{N}\sum_{k=1}^{N}\left|\frac{y(k) - \hat{y}(k)}{y(k)}\right| \quad (5.9)$$

where, $y(k)$ is actual value at sampling k, $\hat{y}(k)$ is prediction value at sampling k, and N is length of time series. These metrics indicate the deviation of the predicted values from the actual values, with lower values signifying better model predictions.

Testing sets	Fault 1		Fault 2		Fault 4		Fault 6	
	First testing set	Second testing set	First testing set	Second testing set	First testing set	Second testing set	First testing set	Second testing set
RMSE	0.4380	1.2636	0.0819	0.2772	2.4511×10^{-4}	6.3382×10^{-4}	0.0331	0.1371
MAPE (%)	5.1106	6.3738	5.0066	6.6027	5.5746	7.3772	5.3269	7.2324

Table 5.3 One-step prediction accuracy of the AR model for test sets across different types of faults

5.4.2 ELM Model

In order to address nonlinear challenges in the fault magnitude prediction, ELM models are developed for fault prediction. Here the same reconstructed fault magnitude datasets of varying severity as in the previous section were used to construct the ELM models, and the number of hidden nodes in each ELM model is determined by trial and error. The data for ELM modelling is split into training data set, testing data set, and validation data set.

In this thesis, the ELM neural network is employed for fault magnitude prediction. To prepare the data for ELM modelling, the data are first normalized to have zero mean and unit variance. This ensures that the input features have the same scale and improves the model's performance. The hidden layer uses the sigmoid activation function, and the output layer uses a linear activation function. The optimal number

of hidden nodes and input sequence dimensions, i.e., model order, for each ELM model is determined through trial and error. The possible range for the input sequence dimension is set to integers between 1 to 40, while the number of hidden layer nodes corresponding to each time length of the dimension can vary from 10-1000. The performance of the ELM models with different input sequence dimensions and hidden layer nodes is evaluated using RMSE and MAPE on the validation set. By analysing the RMSE and MAPE results for different ELM models with varying input sequence dimensions and hidden layer nodes, the optimal model configuration can be identified.

The process of determining the optimal input sequence dimension and the number of hidden layer nodes for each fault involves training and validating several ELM models with different configurations. This means that for each combination of input sequence dimension and hidden layer neuron nodes, an ELM model is trained using the training data and its performance is evaluated using the validation data. The error metrics, such as RMSE and MAPE, are calculated for each of these ELM models to assess their prediction accuracy, and the optimal parameters that allowed for the best accuracy of the ELM prediction models for faults 1,2,4 and 6 are given in Table 5.4.

Fault number	Fault 1	Fault 2	Fault 4	Fault 6
Input sequence dimension	36	38	40	39
Hidden layer nodes	310	620	330	280
RMSE	0.5038	0.3319	1.7749×10^{-4}	0.0482
MAPE (%)	4.7425	7.4580	3.5929	7.2038

Table 5.4 Optimal model parameters configuration for each fault type using the ELM model

The ELM neural network is known for its fast-training capabilities as it randomly generates the connection weights between the input and hidden layers and the bias of the hidden layer neurons. These parameters are not adjusted during the training process, which leads to the instability of its prediction results, and even after several trials to select the best combination of input sequence dimension and hidden layer neuron nodes. To address the instability issue, 10 distinct ELM models were built for each test set of fault magnitude during testing phase. These models share

commonalities in their input vector dimensions and the number of hidden layer nodes, both of which are the optimal combination of parameters previously identified. The differences lie in the weights and biases of the hidden layers, which are randomly selected for each model.

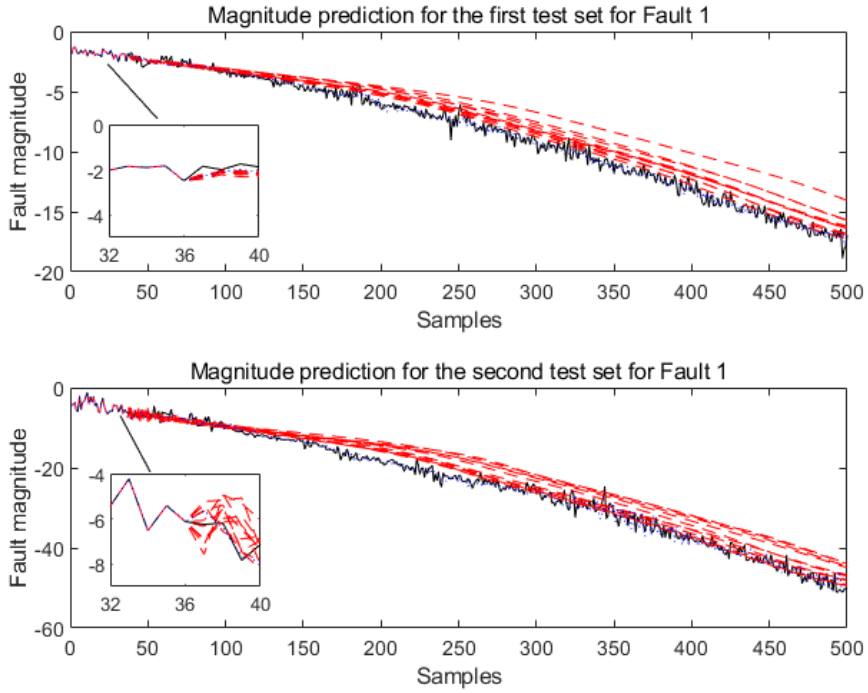

Figure 5.6 Prediction of fault magnitude for testing sets of Fault 1

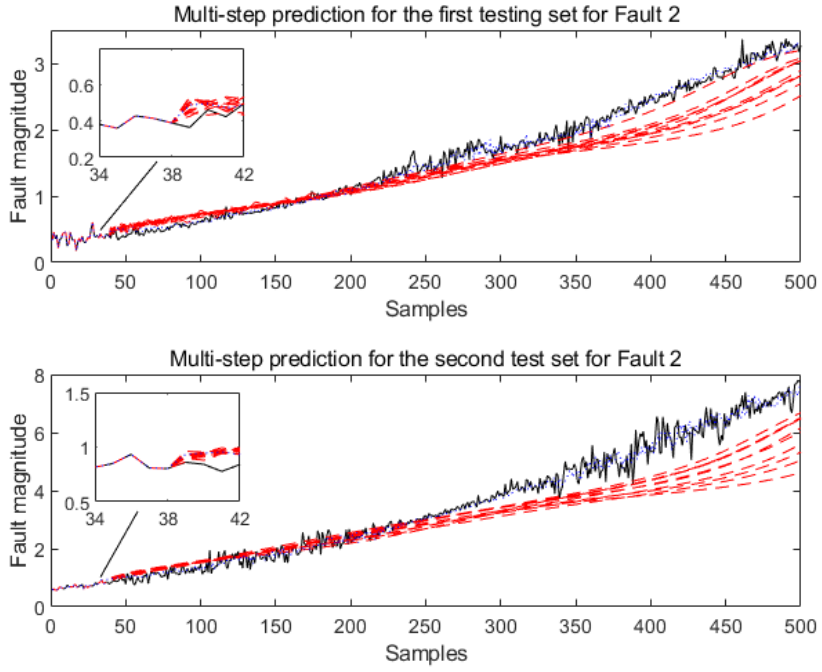

Figure 5.7 Prediction of fault magnitude for testing sets of Fault 2

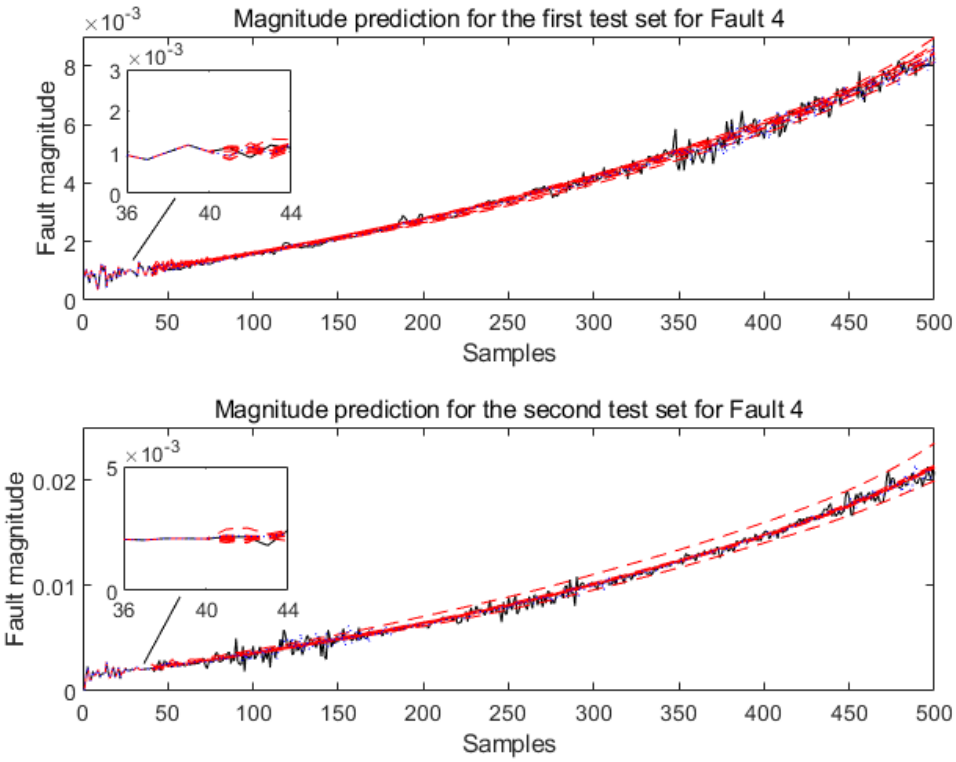

Figure 5.8 Prediction of fault magnitude for testing sets of Fault 4

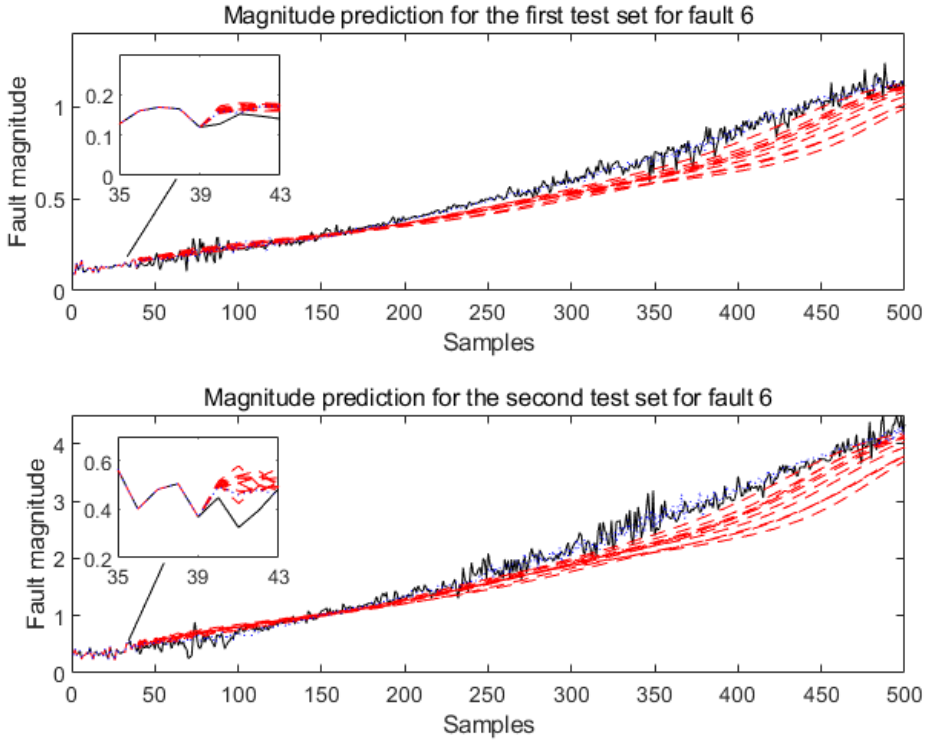

Figure 5.9 Prediction of fault magnitude for testing sets of Fault 6

As shown in Figures 5.6 to 5.9, where the solid black lines indicate the original test set data for the reconstructed fault magnitudes, the blue dotted lines represent one-step predicted fault magnitudes, and the red dashed lines are the multi-step predicted fault magnitudes.

The results demonstrate the ELM models maintain good one-step prediction performance, almost matching the actual values with minimal error, while the multi-step predictions are greatly improved compared to the AR models. In the application of ELM models for multi-step prediction, the specific time step at which these predictions commence is crucial. This initiation point is directly determined by the input sequence dimension, as indicated in Table 5.4. The input sequence dimension is pivotal as it represents the number of past observations the model uses to make its initial prediction. This ensures that the model has sufficient historical data to accurately forecast future values, which is particularly vital in the complex, non-linear data scenarios of fault magnitude prediction. Once the multi-step prediction process begins, the ELM model utilizes historical data for its early predictions. Gradually, as it extends further into the future, the model transitions to relying on its own previous predictions. This iterative process allows each subsequent prediction to build upon the last, encapsulating the data's underlying trends. Despite this, the drawbacks of the ELM are obvious. As can be seen from multi-step prediction results in Figures 5.6-5.9, a significant observation in the ELM model's multi-step predictions is the variation in trends, although the 10 models have the same network structure, their prediction results do not remain consistent due to the ELM randomly initialising the weights and biases of the hidden layer neurons. As these parameters are not adjusted during training, different models with the same network structure can yield varying outcomes. These multiple prediction trends, illustrated by various red lines in the figures, highlight the unique predictive characteristics and challenges inherent in the ELM model when applied to multi-step prediction scenarios.

Table 5.5 lists the multi-step prediction accuracy of the ELM models with the best prediction performance on the test sets for each fault.

Testing sets	Fault 1		Fault 2		Fault 4		Fault 6	
	First testing set	Second testing set	First testing set	Second testing set	First testing set	Second testing set	First testing set	Second testing set
RMSE	0.4032	1.3298	0.2646	0.3286	1.3234×10^{-4}	4.5212×10^{-4}	0.0456	0.1737
MAPE	4.3892	5.2613	7.3260	7.5296	3.1828	4.8789	7.0259	8.0573

Table 5.5 Multi-step prediction accuracy of ELM models

5.5 Conclusions

This chapter focused on the prediction of incipient process faults by developing two distinct prediction models: the AR model and the ELM model. Both models were tested on four different types of incipient process faults (faults 1, 2, 4, and 6). The analysis showed that the AR model performs well for one-step predictions, capturing the trend of the time series data. However, its multi-step ahead prediction performance was subpar due to its linear nature and the assumption of data stationarity. In contrast, the ELM model, which is designed to handle nonlinear data, showed significant improvement in multi-step predictions. Its fast-training capabilities and performance optimization based on input sequence dimensions and hidden layer nodes were notable advantages. However, the inconsistency of ELM predictions due to the random weights and biases of the hidden layers was a drawback. Overall, the ELM model outperformed the AR model for incipient process fault prognosis. It demonstrated better capabilities in capturing the trends and patterns in the fault magnitudes, offering enhanced multi-step ahead predictions. This improved performance makes the ELM model a more suitable choice for multi-step fault prognosis.

However, it is important to acknowledge the limitations and assumptions of this study. First, the analysis focused on a specific type of industrial process data, which might limit the generalizability of this chapter to other types of processes or datasets. Future research could investigate the performance of the AR and ELM models in different contexts or with other types of data. Second, this chapter did not explore potential optimizations of the ELM model, which could further enhance its

performance in long-range fault prediction. The practical implications of this chapter are valuable for both academic researchers and industrial practitioners, as they provide guidance on selecting the most suitable modelling approach for long-range fault prediction in industrial processes. By choosing the ELM model over the AR model, industry professionals can improve the effectiveness of their fault prediction strategies, potentially reducing downtime, maintenance costs, and safety hazards.

In light of this chapter and existing literature, the ELM model emerges as a valuable tool for long-range fault prediction in industrial processes. While the ELM model shows promise with its superior multi-step prediction capabilities, its performance could be further enhanced through a hybrid approach with PCA. Future research should explore integrating PCA's robust feature extraction and dimensionality reduction capabilities with ELM's efficient learning mechanism. This synergy could address ELM's prediction inconsistencies and improve overall fault prognosis accuracy. Such a hybrid model, leveraging the complementary strengths of PCA and ELM, has the potential to offer a more comprehensive solution for long-range fault prediction in industrial processes. In additional, future work can also focus on optimizing the ELM model to further enhance its stability, incorporating additional data or features, examining the potential benefits of integrating it with other modelling approaches or techniques and exploring other advanced machine learning models. In addition, give the limitations in long-range prediction performance of the ELM model, there is a need to further explore more advanced machine learning models, such as recurrent neural networks (RNNs), which are better suited to long-range sequential prediction tasks, and the next chapter examines the potential of RNNs to enhance long-range prediction of fault magnitudes in industrial processes. The aim is to enhance the understanding of how more complex and continuous data-driven models can provide more accurate and reliable predictions, thereby contributing to the wider quest for effective and efficient industrial process fault prediction.

Chapter 6 Fault Magnitude Prognosis in Chemical Processes based on Long Short-Term Memory and Gated Recurrent Unit Network

6.1 Introduction

In recent years, with the continuous development of deep learning techniques, some deep learning models have been gradually applied to the study of time series data. A deep learning model, characterized by a deep neural network with multiple non-linear mapping layers, has the inherent ability to perform both abstraction and feature extraction from the input signal. This is achieved through a layer-by-layer process, enabling the model to uncover intricate patterns hidden within the data Among the various deep learning models, recurrent neural networks introduce the concept of time series into the network structure design, making them more adaptable in time series data analysis. Among the many RNN variants, the Long Short-Term Memory (LSTM) network compensates for conventional RNN issues such as gradient vanishing, gradient explosion, and lack of long-term memory capacity. This allows LSTMs to effectively harness long-range time-series information. However, the Gated Recurrent Unit (GRU), another RNN variant, has also gained attention. GRUs, like LSTMs, capture the association between long sequences and mitigate the gradient vanishing or explosion phenomenon effectively. Furthermore, the GRU's simpler structure, in comparison to the LSTM, allows for faster training.

LSTM models have been used successfully in a number of applications in the study of time-series data in different fields, including text-to-language related language modelling, speech recognition, machine translation, multimedia-related audio and video data analysis, image caption modelling, road transport-related traffic flow prediction, etc. However, in the field of process industry fault prediction, the application of LSTM models is very limited, especially for the research problem of fault magnitude prediction.

In this chapter, a long-range fault prognosis method based on LSTM recurrent neural networks is proposed. The detailed design of the network structure, and the implementation of training and prediction are given. The proposed method is applied to the simulated CSTR system, and the results are compared with those of

the ELM model. The results demonstrate the superior performance of the proposed LSTM model in long range fault prognosis.

The remaining part of this chapter is organised as follows. Section 5.2 presents the structure of the LSTM-based fault prognosis model and describes the LSTM training and prognosis process in detail. Section 5.3 adjusts the input sequence structure and network model construction of the LSTM through different experiments, which are used to determine the optimal model to be used for fault prognosis. Section 5.4 shows the results of the LSTM model for fault magnitude prediction and the corresponding analysis. Finally, Section 5.5 presents the conclusions.

6.2 Fault Prognosis based on LSTM

The overall framework for building the LSTM prediction model is shown in Figure 6.1 and it includes five functional modules: input layer, hidden layer, output layer, network training and network prediction.

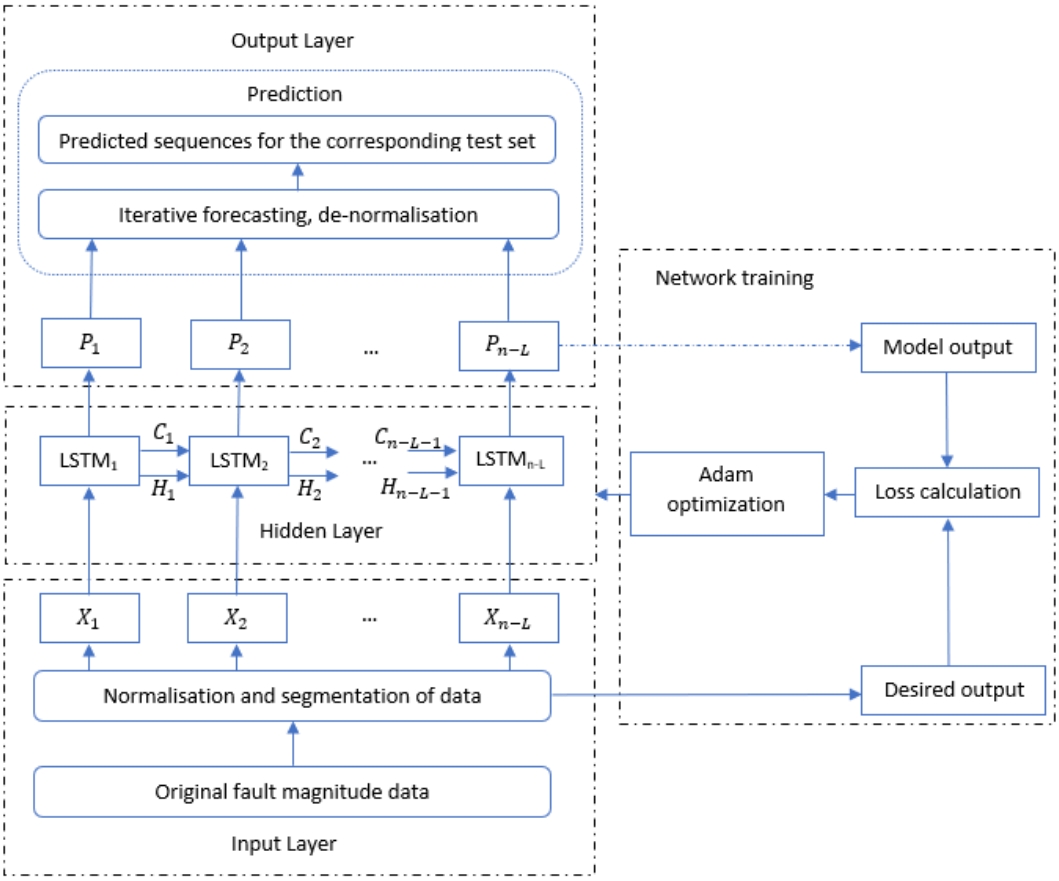

Figure 6.1 LSTM fault prognosis modelling framework

The input layer is responsible for the initial processing of the original fault magnitude time series to meet the network input requirements, the hidden layer uses the LSTM cells represented in Figure 6.1 to build a recurrent neural network, the output layer provides the prediction results, the network training uses the Adam optimisation algorithm mentioned in Chapter 3, and the network prediction uses an iterative prediction method.

6.2.1 Process of Training the LSTM Neural Network

In this thesis, the primary focus is on training the hidden layer of the network. First, in the input layer, a training set consisting of the fault magnitude time series obtained by using the reconstruction is defined as follows.

$$F = [F_1, F_2, \cdots, F_m]^T \in R^{m \times n} \tag{6.1}$$

$$F_i = [f_{1i}\ f_{2i}\ \cdots\ f_{ni}] \tag{6.2}$$

In the above equations, F represents the set of data with different fault magnitudes obtained by fault reconstruction under specific type of fault. Here, F_i refers to the time series of i^{th} ($i = 1,2,\cdots,m$) magnitude of this specific fault, m represents the total number of different magnitudes of this fault, and n is the number of samples in each time series. The training set are normalised to zero mean and unit variance. The normalized training set F' can be expressed as:

$$F' = [F_1', F_2', \cdots, F_m'] \tag{6.3}$$

$$F_i' = [f_{1i}'\ f_{2i}'\ \cdots\ f_{ni}'] \tag{6.4}$$

As fault magnitude prediction is a time series-based problem, the model cannot be trained directly on a feature vector and, instead, the model should be trained on a vector sequence. Therefore, to accommodate the hidden layer input features, the data segmentation approach is applied to each fault magnitude data F_i', and the segmentation window length is set to take the value L. The matrix obtained after segmentation of each fault magnitude used for training can be represented as XS_i.

$$XS_i = \begin{bmatrix} f_1' & f_2' & \cdots & f_L' \\ f_2' & f_3' & \cdots & f_{L+1}' \\ \vdots & \vdots & \cdots & \vdots \\ f_{n-L}' & f_{n-L+1}' & \cdots & f_{n-1}' \end{bmatrix} \in R^{(n-L)\times L} \tag{6.5}$$

By applying segmentation to each fault magnitude data, the final input training data set is transformed into an array of cells consisting of multiple matrices of segmented fault magnitude data.

$$X = \{XS_1, XS_2, \cdots, XS_m\} \tag{6.6}$$

Then the input sequence $X_s, s = 1,2, \cdots, n - L$ of each LSTM cell can be represented as each row vector in the cell array.

Similarly for the corresponding desired output Y for each fault magnitude used for training is:

$$Y = \{YS_1, YS_2, \cdots, YS_m\}$$
$$YS_i = \begin{bmatrix} f'_2 & f'_3 & \cdots & f'_{L+1} \\ f'_3 & f'_4 & \cdots & f'_{L+2} \\ \vdots & \vdots & \cdots & \vdots \\ f'_{n-L+1} & f'_{n-L+2} & \cdots & f'_n \end{bmatrix} \in R^{(n-L)\times L} \tag{6.7}$$

Next, X is fed into the hidden layer. As can be seen from Figure 6.1, the hidden layer contains $n - L$ isomorphic LSTM cells connected by time, and the output of X after passing through the hidden layer can be expressed as:

$$P = [P_1, P_2, \cdots, P_L] \tag{6.8}$$

$$P_p = LSTM_{forward}(X_p, C_{p-1}, H_{p-1}), 1 \le p \le n - L \tag{6.9}$$

where C_{p-1} and H_{p-1} are the cell state and hidden state of the previous LSTM cell, respectively, and $LSTM_{forward}$ represents the forward calculation method for LSTM. In this case, S, refers to the number of LSTM units, hence the dimensions of the cell state vector and previous LSTM cell's output C_{p-1} and output vector H_{p-1} are set to this dimensionality. It can be seen that the hidden layer output P, the model input X and the desired output Y are all two-dimensional arrays of dimension $(n - L) \times L$.

The mean square error is chosen as the error calculation formula and the training process of the loss function can be defined as:

$$loss = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6.10}$$

Given the learning rate, the number of training steps and an optimisation objective of minimising the loss function. The Adam optimization algorithm is applied to continuously update the network weights to obtain the final hidden layer network.

6.2.2 Prediction

The trained LSTM model (denoted as $LSTM^*$) is now used for prediction. The prediction process proceeds iteratively.

First, the first L data from the test set are used as the initial input to the network.

$$X_{ts} = [f_1' \; f_2' \; \cdots \; f_L'] \tag{6.11}$$

The $LSTM^*$ model take X_{ts} as input and generates an output sequence. The last value of this output sequence denoted as $\hat{p}_{L+1}$ is considered as the predicted value for the next time point. Then, the network output can be expressed as:

$$P_{ts} = LSTM^*(X_{ts}) = [p_2, p_3, \cdots, \hat{p}_{L+1}] \tag{6.12}$$

For the next iteration, the sequence X_{ts} is updated by removing the first element f_1' and appending the newly predicted value $\hat{p}_{L+1}$ to the end. This forms the new input sequence X_{ts+1}:

$$X_{ts+1} = [f_2' \; f_3' \; \cdots \; \hat{p}_{L+1}] \tag{6.13}$$

This process is iteratively performed where each new prediction is appended to the sequence while the oldest value is removed, and the updated sequence is fed back into the $LSTM^*$ model for the next prediction. The prediction process stops when the required number of future points has been generated.

This generates a sequence of future predictions:

$$\hat{P}_{ts+n} = LSTM^*(X_{ts+n-1}) = [\hat{p}_{L+1}, \hat{p}_{L+2}, \cdots, \hat{p}_n] \tag{6.14}$$

Next, the prediction sequence corresponding to the test set X_{test} is scaled back to the original scale.

6.3 Model Structure and Parameter Determination

The dataset used in this chapter is the same as the estimated process fault magnitude datasets used for ELM in Chapter 5, as shown in Table 5.1. In this chapter, the LSTM network used the following model initial parameters: one LSTM hidden layer, 200 hidden layer neuron nodes, the solver was set to 'Adam' and trained for 400 epochs, while the gradient bias was set to 1 to prevent gradient explosion. The initial learning rate was 0.005 and was reduced to 0.001 after 125 epochs.

6.3.1 Determination of Segmentation Length

The segmentation length (model order) can be defined as the quantity of previous data points, which essentially refers to the amount of historical information encompassed within a training sequence. By adjusting the segmentation length, the performance of the model for multi-step prediction can be enhanced.

Different data segmentation length sizes, ranging from 2 to 40, were compared using the initial LSTM parameters. The comparison was based on the RMSE and MAPE on the validation set for different segmentation lengths. From the result in Table 6.1 it can be observed that the optimal segmentation length varies for different faults. For instance, the best performance on the validation set for fault 2 was achieved with a segmentation length of 38, while for fault 6, a segmentation length of 30 yielded the best results. Therefore, the chosen data segmentation lengths for faults 2 and 6 are 38 and 30, respectively. This indicates that different faults may have different optimal time horizons for prediction, which could be due to the varying dynamics and complexities of different faults.

Faults	Fault 1	Fault 2	Fault 4	Fault 6
Appropriate Segmentation lengths	26	38	18	30
RMSE for validation set	0.4051	0.1819	1.4546×10^{-4}	0.0308
MAPE (%) for validation set	4.5104	4.3429	3.0598	4.8756

Table 6.1 Performance with suitable segmentation lengths for each fault

6.3.2 Determination of LSTM Structures

The structure of the neural network determines its complexity and capability to capture patterns from the data. This experiment compares the depth of various LSTMs based on the RMSE and MAPE on the validation set.

Six different LSTM models were evaluated, each with different number of nodes in the hidden (LSTM) layer and the drop rate of dropout layer. These configurations are selected based on the performance after several trials and exclusions. This experiment was conducted using the optimal segmentation length previously determined for each fault.

The six models compared were:

Model 1: Single LSTM layer with 100 nodes.

Model 2: Single LSTM layer with 200 nodes.

Model 3: Two LSTM layers with 100 and 200 nodes, respectively, separated by a dropout layer of drop rate 0.2.

Model 4: Two LSTM layers with 200 and 200 nodes, respectively, separated by a dropout layer of drop rate 0.2.

Model 5: Three LSTM layers with 100, 200 and 200 nodes, respectively, each with a dropout layer with drop rate of 0.2 between them.

Model 6: Four LSTM layers with 100, 200, 200 and 200 nodes, respectively, each with a dropout layer with drop rate of 0.2 between them.

LSTM structure	Model 1	Model 2	Model 3	Model 4	Model 5	**Model 6**
RMSE	0.4258	0.4189	0.5743	0.5340	0.5488	**0.3991**
MAPE (%)	4.6676	4.6525	5.9555	5.4202	5.6240	**4.4741**

Table 6.2 Comparison of the depth of LSTM for Fault 1

LSTM structure	Model 1	Model 2	**Model 3**	Model 4	Model 5	Model 6
RMSE	0.0833	0.0923	**0.0790**	0.0824	0.1891	0.0943
MAPE (%)	4.2649	5.0933	**4.0857**	4.4126	9.5973	5.0458

Table 6.3 Comparison of the depth of LSTM for Fault 2

LSTM structure	Model 1	Model 2	Model 3	Model 4	Model 5	**Model 6**
RMSE	2.1882×10^{-4}	2.2677×10^{-4}	2.5471×10^{-4}	4.2336×10^{-4}	2.2646×10^{-4}	$\mathbf{1.3771\times10^{-4}}$
MAPE (%)	4.9542	4.1352	5.4127	8.7884	4.7845	**3.8186**

Table 6.4 Comparison of the depth of LSTM for Fault 4

LSTM structure	Model 1	Model 2	Model 3	**Model 4**	Model 5	Model 6
RMSE	0.0396	0.0516	0.0694	**0.0341**	0.0459	0.0401
MAPE (%)	6.4089	6.8713	10.6874	**5.6500**	7.0113	6.5809

Table 6.5 Comparison of the depth of LSTM for Fault 6

It can be seen from Tables 6.2 to 6.5 that the performance of LSTM models varies significantly with different architectures for different faults. For example. Model 6, which has the most complex structure, performed the best for Faults 1 and 4, indicating that these faults may have more complex patterns that require a deeper model to capture. However, for Fault 2, Model 3 performed the best, suggesting that a simpler model may be sufficient for this fault. This indicates that the optimal structure for an LSTM model depends on the characteristics of the specific fault, and that simply increasing the complexity of the model does not always improve performance. This experiment was conducted under the condition of the optimal segmentation length for each fault.

6.3.3 Determination of Dropout Rate

In deep training process of deep learning models, it's not the case that only some of the nodes are active while others are dormant. Rather, all nodes participate in the learning process but with varying degrees of influence on the output This can be problematic as the model may become overly reliant on certain nodes, leading to overfitting. Overfitting occurs when the model overlearns the training data, which results in high prediction accuracy and a small loss function on the training data, but poor performance on the test data, resulting in a larger loss function and lower prediction accuracy. To mitigate this issue, dropout layers are introduced into the network to randomly deactivate a fixed percentage of nodes in each layer during training. The purpose of this approach is not to train every node equally, but rather to prevent the model from relying too heavily on specific sets of nodes. By deactivating a subset of nodes, dropout layer promotes a more dispersed and resilient representation of the data across the network, as the network cannot depend on the presence of any one node. This process helps to reduce overfitting by encouraging redundancy in the model, thus improving the model's ability to generalize. In effect, various subsets of nodes are trained on different mini-batches

of data, allowing each node to learn and contribute to the model's comprehension of the data over the course of the training process.

In this experiment, the dropout rate was selected in the range from 0.1 to 0.5 with increment of 0.1 and the best value was determined by the lowest RMSE of the validation data set.

Drop rate	0.1	**0.2**	0.3	0.4	0.5
RMSE	0.4856	**0.3086**	0.4615	0.6235	1.0728
MAPE (%)	5.1905	**4.5548**	4.8954	6.1237	10.2452

Table 6.6 Comparison of the drop rate of LSTM for Fault 1

Drop rate	0.1	0.2	**0.3**	0.4	0.5
RMSE	0.1050	0.0978	**0.0786**	0.2729	0.0787
MAPE (%)	5.4269	5.3793	**4.2913**	13.7883	4.6067

Table 6.7 Comparison of the drop rate of LSTM for Fault 2

Drop rate	0.1	**0.2**	0.3	0.4	0.5
RMSE	4.0830×10^{-4}	$\mathbf{1.1512\times10^{-4}}$	2.5933×10^{-4}	2.6203×10^{-4}	2.5294×10^{-4}
MAPE (%)	9.6852	**4.0129**	5.2154	5.2461	5.3208

Table 6.8 Comparison of the drop rate of LSTM for Fault 4

Drop rate	0.1	0.2	**0.3**	0.4	0.5
RMSE	0.0367	0.0439	**0.0343**	0.0369	0.0493
MAPE (%)	5.7134	7.2128	**5.5009**	5.9115	6.7082

Table 6.9 Comparison of the drop rate of LSTM for Fault 6

Tables 6.6 to 6.9 shows the prediction accuracy of the LSTM models with different dropout rates for faults 1, 2, 4, and 6 respectively. It can be seen that the optimal

dropout rates differ for different faults. For instance, Faults 1 and 4 achieved the best performance with a dropout rate of 0.2, while Faults 2 and 6 performed best with dropout rates of 0.3. These results demonstrate the importance of tuning the dropout rate to strike the right balance between learning the underlying patterns in the data and avoiding overfitting.

6.4 Results and Analysis

This section describes the prediction results and correlation analysis of the proposed model, with a particular focus on the multi-step prediction capabilities of LSTM and GRU networks. Among the many deep learning models, RNN introduces the concept of time series into the network structure design, making it highly adaptable for time series data analysis. However, as mentioned in Chapter 3, the gradient vanishing and explosion problems of RNNs make them less capable of solving long time series problems, so LSTM networks have been proposed to improve this problem. However, the relatively complex internal structure of the LSTM has led to its relatively low training efficiency (longer training time). In order to comprehensively compare the LSTM and GRU, this section follows the same experimental approach in Section 6.3 for GRU models. This approach involves determining the optimal parameters and model structure for the GRU network and comparing it directly with the LSTM. The goal is to evaluate the relative strengths and weaknesses of each model, particularly in terms of the proposed LSTM model.

In these experiments, optimal parameters were determined for the GRU model for different types of faults, which are presented in Table 6.10. While LSTM and GRU share similarities in terms of addressing the vanishing gradient problem and capturing long-term dependencies, they have some differences in their internal structure.

Fault	Fault 1	Fault 2	Fault4	Fault 6
Segmentation length	37	36	36	29
GRU structure	Model 3	Model 4	Model 5	Model 3
Dropout rate	0.1	0.2	0.2	0.1

Table 6.10 The optimal parameters and structure for GRU network

The optimal parameters and structure of the model built using the GRU layer are similar to those of the model built using the LSTM, with the difference that the model structure is simpler and requires lower dropout rate. This is because the GRU network has a simpler architecture compared to the LSTM network through combining the input and forgetting gates of the LSTM into a single update gate. This reduces the number of parameters and calculations involved and making the model more compact. Furthermore, the simple structure reduces the risk of overfitting, which also makes it possible that the GRU model may require a lower dropout rate during training.

Figures 6.2 to 6.5 illustrate, respectively, the multi-step prediction results of the test sets of faults 1, 2, 4, and 6 using LSTM and GRU optimal models. In these figures, the solid black line indicates the actual fault magnitudes, the blue dotted line indicates the multi-step prediction results of the GRU models, and the red dashed line is the multi-step prediction results of the LSTM models. A critical aspect of our multi-step prediction approach using LSTM and GRU involves maintaining a consistent length for the input sequence. For instance, for fault type 1, with a segmentation length of 37 as shown in Table 6.10, the model uses these 37 historical data points to predict the value at the 38th time step. Subsequently, to maintain a constant sequence length, the earliest time step data is discarded, and the newly predicted value is added to the input sequence. This process of updating the input sequence continues with each subsequent prediction step, gradually replacing all historical data with predicted values. This methodology is pivotal for practical applications, illustrating the models' ability to shift from historical data dependency to relying on their own predictive outputs, especially relevant for ongoing, real-time monitoring and prediction tasks in industrial settings.

Compared to the ELM model, although these two models are more time-consuming during their training process, as the weights and biases of the network are updated at each time step, they allow the models to effectively tap into the intrinsic trends of the time series and the correlations between the sequences, and ultimately give more stable and accurate multi-step prediction results.

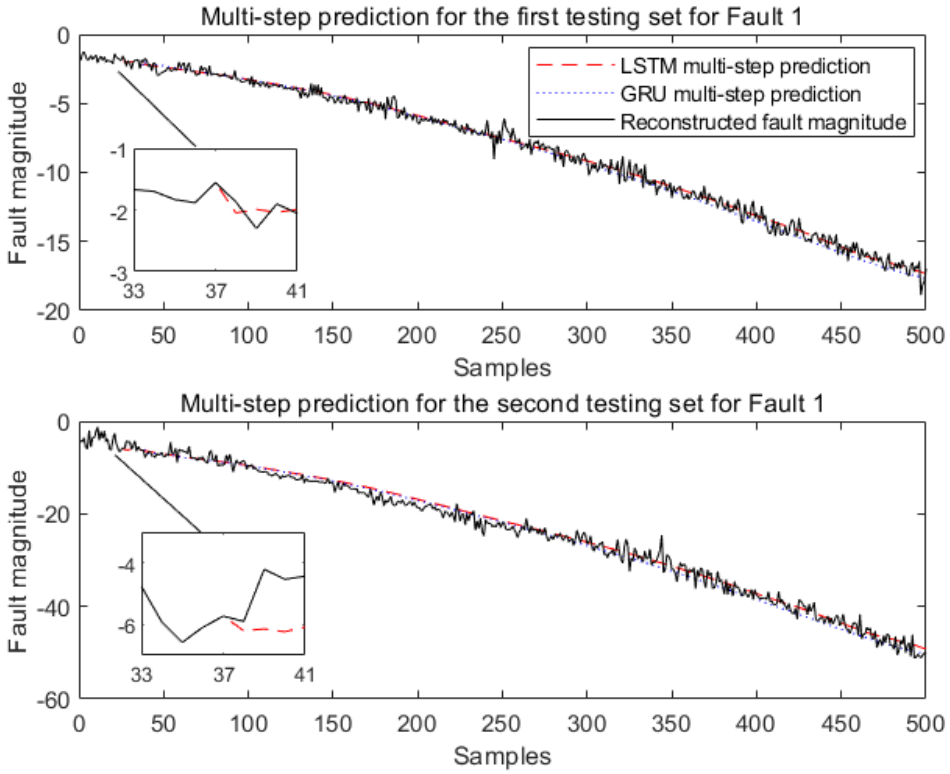

Figure 6.2 Comparison of LSTM and GRU for the testing sets of Fault 1

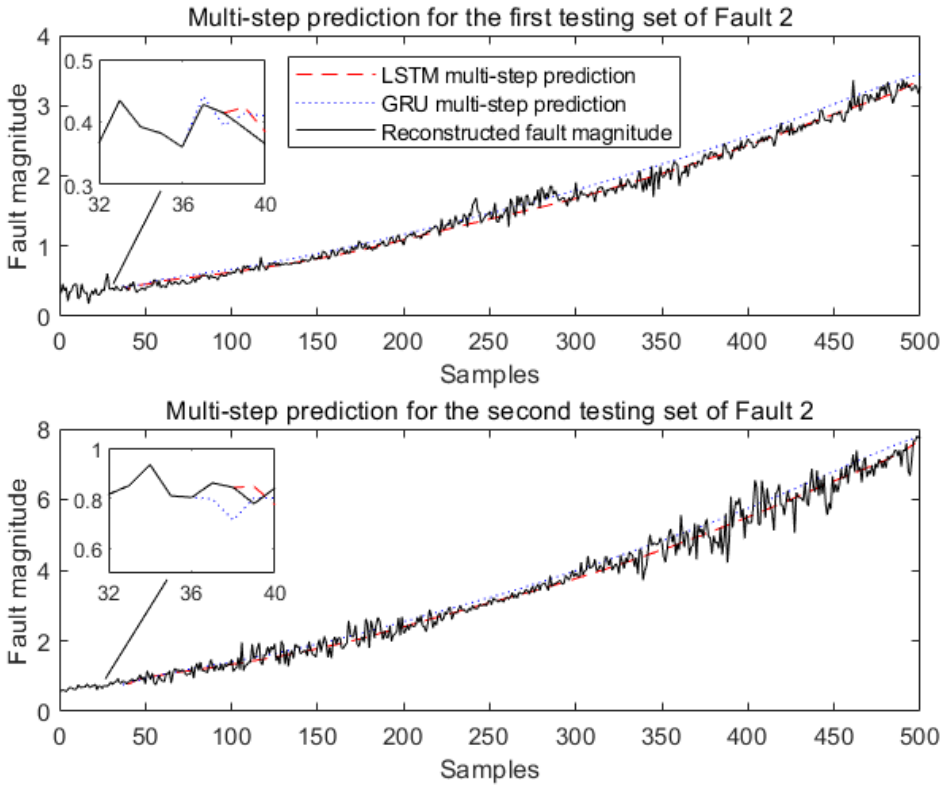

Figure 6.3 Comparison of LSTM and GRU for the testing sets of Fault 2

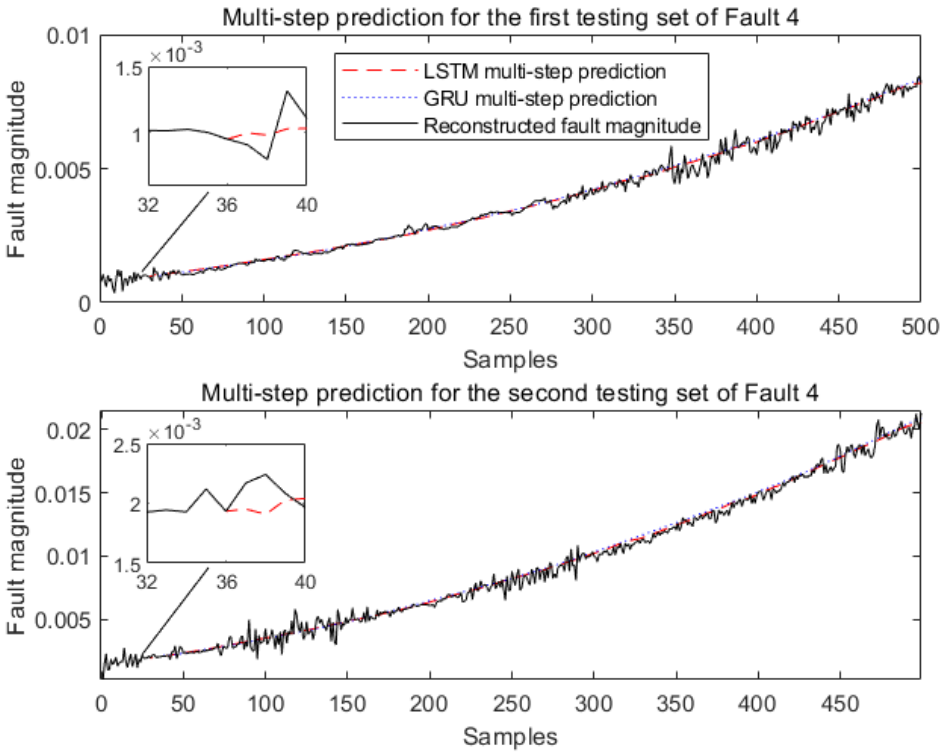

Figure 6.4 Comparison of LSTM and GRU for the testing sets of Fault 4

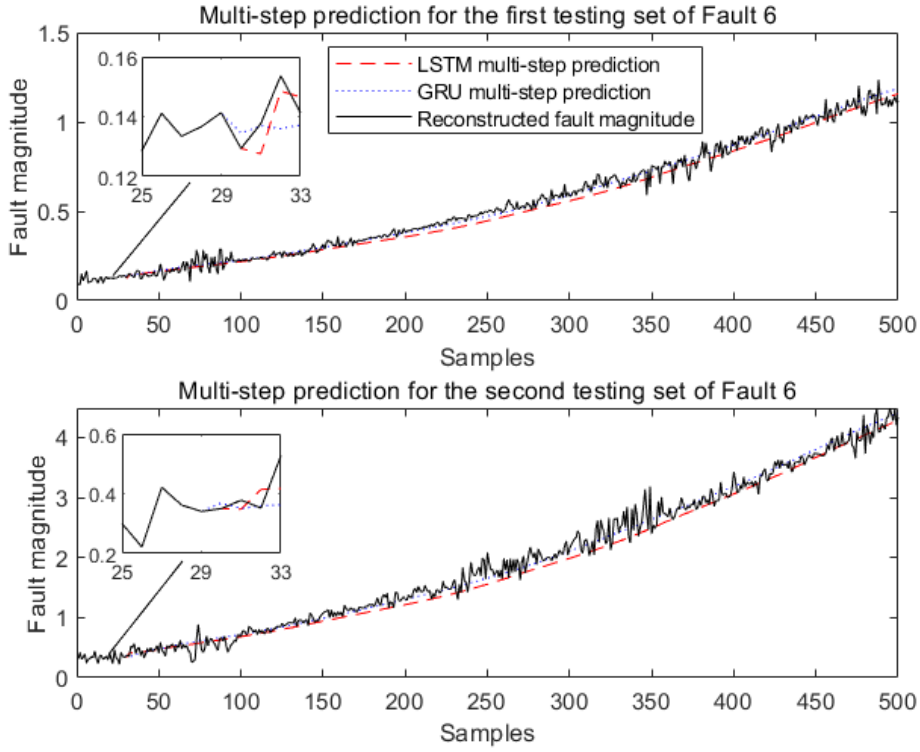

Figure 6.5 Comparison of LSTM and GRU for the testing sets of Fault 6

Table 6.11 gives the training times of the LSTM and GRU models for different faults. The training time consumption of the GRU model is relatively less compared

to the LSTM model due to its simple structure (only have update and reset gates), smaller number of parameters and no additional cell states are required.

Fault types	Fault 1	Fault 2	Fault 4	Fault 6
LSTM training time	5 mins 10 secs	10 mins 40 secs	9 mins 30 secs	6 mins
GRU training time	4 mins 23 secs	9 mins	7 mins 10 secs	5 mins 10 secs

Table 6.11 Training time consumption for LSTM and GRU models

Prediction accuracy	Frist testing set		Second testing set	
	RMSE	MAPE (%)	RMSE	MAPE (%)
LSTM	0.4001	4.4526	1.2400	4.7929
GRU	0.4131	4.3455	1.3042	5.5272

Table 6.12 Prediction performance for the testing sets of Fault 1

Prediction accuracy	Frist testing set		Second testing set	
	RMSE	MAPE (%)	RMSE	MAPE (%)
LSTM	0.0815	4.2537	0.2497	5.6804
GRU	0.1155	6.5349	0.3029	7.0400

Table 6.13 Prediction performance for the testing sets of Fault 2

Prediction accuracy	Frist testing set		Second testing set	
	RMSE	MAPE (%)	RMSE	MAPE (%)
LSTM	1.2089×10^{-4}	3.5011	1.9385×10^{-4}	3.6237
GRU	1.3222×10^{-4}	4.7417	4.0059×10^{-4}	4.4273

Table 6.14 Prediction performance for the testing sets of Fault 4

Prediction accuracy	Frist testing set		Second testing set	
	RMSE	MAPE (%)	RMSE	MAPE (%)
LSTM	0.0315	4.9269	0.1281	6.4687
GRU	0.0400	6.6686	0.1651	7.9593

Table 6.15 Prediction performance for the testing sets of Fault 6

The RMSE and MAPE were then used to evaluate the performance of the model and the results are shown in Tables 6.12 to 6.15. The RMSE, by squaring the residuals, quantifies the average magnitude of the prediction errors or residuals, thus reflecting the concentration of these errors and thus provides insight into the magnitude of large errors made by the model. On the other hand, MAPE, an absolute percentage error metric, provides an average model performance by showing the relative error between the predicted and actual values in percentage terms. It can be seen that in terms of prediction accuracy LSTM is more accurate than GRU.

6.5 Conclusions

In this chapter, an LSTM-based process fault magnitude trend prediction model is developed, which is an important step in understanding and predicting process faults. The principal strength of LSTM-based model lies in its ability to learn and remember long-term dependencies in time series data.

Various experiments and analyses demonstrate the robustness of the proposed LSTM model. Compared to the GRU method, the overall accuracy of the proposed method is higher than the GRU model as well as the ELM model in Chapter 4, although the training process is more time-consuming. Tests across multiple magnitudes of different faults revealed that distinct prediction models could be constructed for each fault, demonstrating the flexibility and adaptability of the LSTM-based approach. Furthermore, these models exhibited excellent long-term prediction results, demonstrating the model's potential for practical applications in predicting process faults over extended periods.

On the other hand, there are still certain limitations of this approach that need to be recognised. One of the main challenges is the optimal selection of LSTM model parameters. While the model performed well in our experiments, it is well-known

that LSTM models can be sensitive to the choice of hyperparameters. The process of hyperparameter tuning can be computationally intensive and time-consuming, which may limit the practicality of the LSTM-based model in some real-world scenarios. Another potential weakness lies in the model's reliance on historical data. While this data-driven approach has proven effective, it is worth noting that the accuracy of predictions depends largely on the quality and comprehensiveness of the data used for training. In scenarios where historical data is limited, inaccurate, or biased, the model's performance could be significantly compromised.

Nevertheless, these limitations also open up avenues for further research. The potential next step in this research is to explore a sophisticated hybrid model that effectively integrates PCA, ELM, and LSTM. Such a model would employ PCA for initial data preprocessing, enhancing data quality, followed by ELM for rapid pattern recognition. The process would culminate with LSTM's strength in capturing long-term dependencies. This integrative approach promises to refine fault prognosis by combining the unique strengths of each method, potentially leading to more accurate and reliable predictions. Further enhancement could also focus on developing more effective methods for LSTM parameter optimization, which could potentially lead to more accurate and reliable predictions. The data-dependency issue could be addressed by integrating domain knowledge into the model. As highlighted in this chapter, the next step is to apply key features and elements extracted from domain knowledge related to reliability. This forward study of reliability prediction methods could potentially augment the LSTM model's predictive power by providing additional context and guidance to the learning process, particularly in scenarios where historical data is scarce or flawed.

Chapter 7 Conclusions and Recommendations for Future Works

This chapter summarizes the substance of this thesis, outlining the findings of the current study, as well as limitations and areas in need of further development. It also proposes potential topics for deeper exploration in future studies.

7.1 Conclusions

Fault detection and diagnosis play an essential role in ensuring the safety and reliability of the process industry. This area has become increasingly significant in the realm of industrial process control. In recent years, with the advancement of industrial big data analytics, data-driven approaches have gained prominence and are now more frequently employed than model-based and knowledge-based methods. Research and applications in data-based fault detection and diagnostic systems have proliferated, reflecting the growing interest and importance of this field. Improvements to traditional multivariate statistical method are provided in this thesis with the goal of assuring the process industry's safe and dependable functioning. With the increasing automation and complexity of the process industry, traditional approaches to process maintenance which typically involve addressing faults after they have occurred, are becoming less effective. It is preferred to take measures to eliminate the effects of the fault before or at the beginning of the fault in order to reduce maintenance costs and improve process stability. As a result, fault prognosis technology has become a new research focus on the process monitoring field.

This thesis proposes an enhanced fault reconstruction approach for process fault detection and diagnosis, as well as a novel model for fault magnitude prediction, based on classic multivariate statistical and machine learning methods, in order to maintain the safety and reliability of industrial processes. The following are the conclusions drawn from the work reported in this thesis:

1. Previous work using fault reconstruction approaches for fault diagnosis has mainly focused on sensor fault diagnosis, where sensor measurements were not utilized for closed-loop control of processes. Process fault reconstruction is more challenging than those sensor fault reconstruction because a process fault typically affects a large number of variables to various degrees, making it difficult to generate a fault direction matrix. It is assumed that relevant process fault data is accessible for analysis. Under

this premise, the fault direction matrix for a process fault can be effectively extracted through PCA of the corresponding process data recorded during the fault condition. The loading matrix of the fault data is introduced into the fault direction matrix, allowing for a more intuitive and rapid derivation of the fault direction matrix for process faults. The enhancements made in fault reconstruction methodologies have demonstrated tangible benefits, particularly when applied to a simulated Continuous Stirred Tank Reactor (CSTR) process. These improvements include more accurate fault identification, quicker response times to anomalous conditions, and a better understanding of the fault's impact on process dynamics.

2. This thesis develops an AR-based fault magnitude prediction model as well as an ELM-based fault magnitude prediction model. The AR prediction model was used as a baseline since the reconstructed fault magnitudes acquired by the fault reconstruction approach are time series. The results show that the AR model can offer very accurate single-step forecasts. While effective in single-step predictions, the AR model's performance significantly diminishes in multi-step forecasting, particularly with non-linear industrial process data. This is primarily due to its reliance on past values and linear assumptions, which become less representative as predictions extend into the future. Consequently, the model's ability to accurately predict fault magnitudes over longer horizons is limited.

   Therefore, an ELM neural network-based prediction model was suggested, and the results demonstrate that the ELM fault prediction model could provide relatively accurate long-range fault magnitude predictions. However, while the prediction results of ELM model are far more accurate than those of the AR model, the fact that its hidden layer parameters are chosen at random and the network is not iteratively updated causes its prediction results to be inconsistent, necessitating several training sessions based on empirical judgement to find the appropriate number of hidden nodes and model order of the time series model. Furthermore, even with appropriate model parameters, stable prediction results are not always produced as demonstrated in the result analysis.

3. This thesis then proposes both LSTM and GRU based process fault magnitude trend prediction models to overcome the ELM prediction model's low stability. The resilience and performance of both LSTM and

GRU approaches are investigated and compared through several tests. Optimal model inputs, network parameters, and structure are identified for various faults. Both approaches are subjected to testing across a variety of fault magnitudes. The findings suggest that robust prediction models can be built for a wide range of faults, achieving good long-term prediction outcomes. A specific emphasis has been placed on the LSTM-based model due to its demonstrable advantages in the performed tests.

The results in this thesis contribute to the ongoing research in process monitoring, reflecting a progression from the foundational efforts of previous scholars in the field. Although it has made contributions to the field of process monitoring research in certain ways, it still needs to be improved, particularly in terms of model parameter optimization. Machine learning and deep learning models have more model complexity and more model parameters than multivariate statistical analysis models, such as the number of hidden layer nodes, network learning rate, input sequence length, and so on. However, several of the parameters rely on experience to calculate the approximate value interval, which inadvertently increases the difficulty of model optimization and diminishes the model's effectiveness. To deal with the challenge of optimising model parameters so that the model can retain a higher performance when dealing with changing data sets, several adaptive approaches or optimisation algorithms are required.

7.2 Recommendations for Future Works

Taking some of the experiences obtained from the research work conducted in this thesis, as well as the most recent trends in current advances in the field of process monitoring, the following are some possible future research directions.

1. Building upon the separate strengths of PCA-based methods, ELM, and LSTM models explored in this thesis, future research can aim at developing hybrid models. Such models would combine the best attributes of these methods to form more robust and reliable process fault prediction systems. For example, PCA and ELM could be used for early fault detection and identification, while LSTM could be employed for long-range fault prediction due to its ability to learn and remember long-term dependencies in time series data.
2. This thesis develops certain diagnostic and prognostic approaches based only on multivariate statistical and machine learning methods or ideas.

Although obtaining precise mechanistic models for large-scale process industries is challenging, it is still feasible to build sufficiently accurate mechanistic models for individual processes or units. A suitable and sufficient mixture of mechanistic models is expected to enhance large-scale process diagnosis and prognosis. Furthermore, there is tremendous value in the experience of specialists or operational engineers, which, when employed properly, has the ability to increase fault diagnosis and prognosis accuracy. Therefore, more research into diagnosis and prognosis in the process industry from the perspective of hybrid intelligence that integrates mechanical, data and empirical knowledge has the potential to address the limitations of current diagnostic and prediction methods and lead to more accurate and reliable results.

3. The models in this thesis were primarily tested on a specific type of industrial process. Hence, future research could aim to apply these models to a wider range of industrial processes or datasets. This would not only test the generalizability of the models but also potentially increase their applicability across different industries and sectors. Attention could also be given to the real-time implementation of these models in industrial settings. This would involve designing experiments to validate the performance of the models when used for online monitoring and prediction of process faults. The findings could then be used to further refine the models and make them more suited for real-world applications.
4. To address the challenges associated with the selection of model parameters, future research could investigate automated hyperparameter tuning methods like Bayesian optimization, grid search, or genetic algorithms. This could make the training process more efficient and enhance the predictive performance of the models.

Reference

Abbasi, T., Lim, K.H. and Yam, K.S. (2019) 'Predictive Maintenance of Oil and Gas Equipment using Recurrent Neural Network', *IOP Conference Series: Materials Science and Engineering*, 495, p. 012067.

Abed, W., Sharma, S., Sutton, R. and Motwani, A. (2015) 'A Robust Bearing Fault Detection and Diagnosis Technique for Brushless DC Motors Under Non-stationary Operating Conditions', *Journal of Control, Automation and Electrical Systems*, 26(3), pp. 241-254.

Abdi, H., and Williams, L. J. (2010) 'Principal component analysis', *Wiley interdisciplinary reviews: computational statistics*, 2, pp. 433-459.

Amari, S.-i. (1993) 'Backpropagation and Stochastic Gradient Descent Method', *Neurocomputing*, 5(4), pp. 185-196.

Ang, A., Piazzesi, M. and Wei, M. (2006) 'What does the Yield Curve Tell Us About GDP Growth?', *Journal of Econometrics*, 131(1-2), pp. 359-403.

Arockia Dhanraj, J., Prabhakar, M., Ramaian, C. P., Subramaniam, M., Solomon, J. M., & Vinayagam, N. (2022) 'Increasing the Wind Energy Production by Identifying the State of Wind Turbine Blade', *Technology Innovation in Mechanical Engineering: Select Proceedings of TIME 2021*, *Singapore: Springer Nature*, pp. 139-148.

Atiya, A.F., El-Shoura, S.M., Shaheen, S.I. and El-Sherif, M.S. (1999) 'A Comparison Between Neural-Network Forecasting Techniques-case Study: River Flow Forecasting', *IEEE Transactions on Neural Networks*, 10(2), pp. 402-409.

Baruah, P. and Chinnam *, R.B. (2005) 'HMMs for diagnostics and prognostics in machining processes', *International Journal of Production Research*, 43(6), pp. 1275-1293.

Bechhoefer, E., Bernhard, A. and He, D. (2008) 'Use of Paris Law for Prediction of Component Remaining Life', *2008 IEEE Aerospace Conference*. pp. 1-9.

Bie, F., Du, T., Lyu, F., Pang, M., and Guo, Y. (2021) 'An integrated approach based on improved CEEMDAN and LSTM deep learning neural network for fault diagnosis of reciprocating pump', *Ieee Access*, 9, pp. 23301-23310.

Bilbao and Alcerreca, M. (1994) 'The Guadalajara City's Accident-April 1992', *14th World Petroleum Congress*.

Bors, A. (2001) 'Introduction of the Radial Basis Function (RBF) Networks', *Online Symposium for Electronics Engineers*, pp. 1-7.

Bose, M.R., Kumar, G. and Venkateswarlu, C. (2005) 'Detection, isolation and reconstruction of faulty sensors using principal component analysis', *Indian Journal of Chemical Technology*, 12, pp. 430-435.

Cai, B., Wang, Z., Zhu, H., Liu, Y., Hao, K., Yang, Z., and Liu, Z. (2021) 'Artificial intelligence enhanced two-stage hybrid fault prognosis methodology of PMSM', *IEEE Transactions on Industrial Informatics*, 18(10), pp. 7262-7273.

Chang, C.-H. (2015) 'Deep and Shallow Architecture of Multilayer Neural Networks', *IEEE Transactions on Neural Networks and Learning Systems*, 26(10), pp. 2477-2486.

Chen, X., Xiao, H., Guo, Y. and Kang, Q. (2016) 'A multivariate grey RBF hybrid model for residual useful life prediction of industrial equipment based on state data', *Int. J. Wire. Mob. Comput.*, 10(1), pp. 90–96.

Chen, Y., Chen, B., Yao, Y., Tan, C., and Feng, J. (2019) 'A spectroscopic method based on support vector machine and artificial neural network for fiber laser welding defects detection and classification', *NDT & E International*, 108, p. 102176.

Chen, G., & McAvoy, T. J. (1997) 'Multi-block predictive monitoring of continuous processes', *IFAC Proceedings Volumes*, 30(9), pp.73-77.

Cheng, C., Ma, G., Zhang, Y., Sun, M., Teng, F., Ding, H. and Yuan, Y. (2018) 'Online Bearing Remaining Useful Life Prediction Based on a Novel Degradation Indicator and Convolutional Neural Networks', *arXiv preprint arXiv:1812.03315*.

Chiang, L., Russell, E. and Braatz, R. (2001) 'Fault Detection and Diagnosis in Industrial Systems', *Measurement Science & Technology - MEAS SCI TECHNOL*, 12.

Chinnam, R.B. and Baruah, P. (2003) 'Autonomous diagnostics and prognostics through competitive learning driven HMM-based clustering', *Proceedings of the International Joint Conference on Neural Networks, 2003.*, pp. 2466-2471.

Cho, K., Merrienboer, B.v., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014) 'Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation', *arXiv preprint arXiv:1406.1078*, pp. 1724-1734.

Coito, F.V., Palma, L.B. and Silva, R.N.d. (2005) 'Robust fault diagnosis approach using analytical and knowledge based techniques applied to a water tank system', *International journal of engineering intelligent systems for electrical engineering and communications*, 13, pp. 237-244.

Cox, D.R. (1961) 'Prediction by Exponentially Weighted Moving Averages and Related Methods', *Journal of the Royal Statistical Society. Series B (Methodological)*, 23(2), pp. 414-422.

Das, A., Maiti, J. and Banerjee, R.N. (2012) 'Process monitoring and fault detection strategies: a review', *International Journal of Quality & Reliability Management*, 29(7), pp. 720-752.

Demetgul, M. (2013) 'Fault diagnosis on production systems with support vector machine and decision trees algorithms', *The International Journal of Advanced Manufacturing Technology*, 67(9), pp. 2183-2194.

Deutsch, J. and He, D. (2018) 'Using Deep Learning-Based Approach to Predict Remaining Useful Life of Rotating Components', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(1), pp. 11-20.

Deutsch, J., He, M. and He, D. (2017) 'Remaining Useful Life Prediction of Hybrid Ceramic Bearings Using an Integrated Deep Learning and Particle Filter Approach', *Applied Sciences*, 7(7).

Ding, S.X., Zhang, P., Yin, S. and Ding, E.L. (2013) 'An Integrated Design Framework of Fault-Tolerant Wireless Networked Control Systems for Industrial Automatic Control Applications', *IEEE Transactions on Industrial Informatics*, 9(1), pp. 462-471.

Drif, M., Benouzza, N., Kraloua, B., Azeddine, B. and Dente, J. (2002) 'Squirrel cage rotor faults detection in induction motor utilizing stator power spectrum approach', *2002 International Conference on Power Electronics, Machines and Drives*, pp. 133-138.

Du, Y., and Du, D. (2018) 'Fault detection and diagnosis using empirical mode decomposition based principal component analysis', *Computers & Chemical Engineering*, 115, pp. 1-21.

Duchi, J., Hazan, E. and Singer, Y. (2011) 'Adaptive subgradient methods for online learning and stochastic optimization', *Journal of machine learning research*, 12(7).

Dunia, R. and Joe Qin, S. (1998a) 'Subspace approach to multidimensional fault identification and reconstruction', *AIChE Journal*, 44(8), pp. 1813-1831.

Dunia, R. and Joe Qin, S. (1998b) 'A unified geometric approach to process and sensor fault identification and reconstruction: the unidimensional fault case', *Computers & Chemical Engineering*, 22(7), pp. 927-943.

El-Tawil, K. and Jaoude, A.A. (2013) 'Stochastic and nonlinear-based prognostic model', *Systems Science & Control Engineering*, 1(1), pp. 66-81.

Elshenawy, L.M. and Mahmoud, T.A. (2018) 'Fault diagnosis of time-varying processes using modified reconstruction-based contributions', *Journal of Process Control*, 70, pp. 12-23.

Frank, P. M. (1990) 'Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results', *automatica*, 26(3), pp. 459-474.

Franses, P.H. and Legerstee, R. (2009) 'A Unifying View on Multi-Step Forecasting Using an Autoregression', *Wiley-Blackwell: Journal of Economic Surveys*.

Galagedarage Don, M. and Khan, F. (2019) 'Process Fault Prognosis Using Hidden Markov Model–Bayesian Networks Hybrid Model', *Industrial & Engineering Chemistry Research*, 58(27), pp. 12041-12053.Ge, W.E.I. and Fang, C.-Z. (1988) 'Detection of faulty components via robust observation', *International Journal of Control*, 47(2), pp. 581-599.

Gebraeel, N., Lawley, M., Liu, R. and Parmeshwaran, V. (2004) 'Residual life predictions from vibration-based degradation signals: a neural network approach', *IEEE Transactions on Industrial Electronics*, 51(3), pp. 694-700.

Gokilavani, N., and Bharathi, B. (2021) 'Test case prioritization to examine software for fault detection using PCA extraction and K-means clustering with ranking', *Soft Computing*, 25(7), pp. 5163-5172.

Gómez-Pau, Á., Riba, J. R., and Moreno-Eguilaz, M. (2020) 'Time series RUL estimation of medium voltage connectors to ease predictive maintenance plans', *Applied Sciences*, 10(24), p. 9041.

Haimi, H., Mulas, M., Corona, F., Marsili-Libelli, S., Lindell, P., Heinonen, M. and Vahala, R. (2016) 'Adaptive data-derived anomaly detection in the activated sludge process of a large-scale wastewater treatment plant', *Engineering Applications of Artificial Intelligence*, 52, pp. 65-80.

Hamadache, M., Dutta, S., Olaby, O., Ambur, R., Stewart, E. and Dixon, R. (2019) 'On the Fault Detection and Diagnosis of Railway Switch and Crossing Systems: An Overview', *Applied Sciences*, 9(23).

Harrath, S., Ali, J.B., Zouaghi, T. and Zerhouni, N. (2019) 'A New Adaptive Prognostic Strategy Based on Online Future Evaluation and Extended Kalman Filtering', *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 2033-2038.

Heimes, F.O. (2008) 'Recurrent neural networks for remaining useful life estimation', *2008 International Conference on Prognostics and Health*

Management, pp. 1-6.

Hochreiter, S. and Schmidhuber, J. (1997) 'Long short-term memory', *Neural Comput*, 9(8), pp. 1735-80.

Hu, N., Su, X., and Liu, B. (2018) 'A Review on Prognostics and Health Management" International Journal of Advanced Network', *Monitoring and Controls*, 3(1), pp.115-121.

Hu, Q., He, Z., Zi, Y., Zhang, Z. and Lei, Y. (2005) 'Novel hybrid intelligent forecasting model and its application to fault diagnosis', *Hsi-An Chiao Tung Ta Hsueh/Journal of Xi'an Jiaotong University*, 39(9), pp. 928-932.

Hu, Y., Baraldi, P., Di Maio, F. and Zio, E. (2016a) 'Online Performance Assessment Methodfor a Model-Based Prognostic Approach', *IEEE Transactions on Reliability*, 65(2), pp. 718-735.

Hu, Y., Chen, H., Li, G., Li, H., Xu, R. and Li, J. (2016b) 'A statistical training data cleaning strategy for the PCA-based chiller sensor fault detection, diagnosis and data reconstruction method', *Energy and Buildings*, 112, pp. 270-278.

Hu, Y., Deng, X., Cao, Y. and Tian, X. (2017) 'Dynamic process fault prediction using canonical variable trend analysis', *Chinese Automation Congress (CAC)*, pp. 2015-2020.

Huang, G.-B., Zhu, Q.-Y. and Siew, C.-K. (2004) 'Extreme learning machine: a new learning scheme of feedforward neural networks', *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*. pp. 985-990.

Huang, G., Zhu, Q.-Y. and Siew, C.K. (2006) 'Extreme learning machine: Theory and applications', *Neurocomputing*, 70, pp. 489-501.

Isermann, R. (1991) 'Fault Diagnosis of Machines via Parameter Estimation and Knowledge Processing', *IFAC Proceedings Volumes*, 24(6), pp. 43-55.

Isermann, R. (1997) 'Supervision, fault-detection and fault-diagnosis methods — An introduction', *Control Engineering Practice*, 5(5), pp. 639-652.

Ji, Hongquan, Huang, Keke, Zhou, Donghua (2019) 'Incipient sensor fault isolation based on augmented Mahalanobis distance', *Control Engineering Practice*, 86, pp. 144-154.

Jiang, H., Li, X., Shao, H. and Zhao, K. (2018) 'Intelligent fault diagnosis of rolling bearings using an improved deep recurrent neural network', *Measurement Science and Technology*, 29(6).

Jianhui, L., Namburu, M., Pattipati, K., Liu, Q., Kawamoto, M. and Chigusa, S. (2003) 'Model-based prognostic techniques [maintenance applications],'

Proceedings AUTOTESTCON 2003. IEEE Systems Readiness Technology Conference., pp. 330-340.

Joe Qin, S. (2003) 'Statistical process monitoring: basics and beyond', *Journal of Chemometrics*, 17(8-9), pp. 480-502.

Junior, R. F. R., dos Santos Areias, I. A., Campos, M. M., Teixeira, C. E., da Silva, L. E. B., and Gomes, G. F. (2022) 'Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals', *Measurement*, 190, p. 110759.

Kano, M., Tanaka, S., Hasebe, S., Hashimoto, I. and Ohno, H. (2003) 'Monitoring Independent Components for Fault Detection', *AIChE Journal*, 49.

Karthikeyan, A. and Priyakumar, U.D. (2021) 'Artificial intelligence: machine learning for chemical sciences', *Journal of Chemical Sciences*, 134(1).

Kingma, D.P. and Ba, J. (2014) 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980*.

Kinnaert, M. (1993) 'Design of Redundancy Relations for Failure Detection and Isolation by Constrained Optimization', *IFAC Proceedings Volumes*, 26(2), pp. 387-390.

Kościelny, J. M., Bartyś, M., Syfert, M., & Sztyber, A. (2022) 'A Graph Theory–Based Approach to the Description of the Process and the Diagnostic System', *International Journal of Applied Mathematics and Computer Science*, 32(2), pp. 213-227.

Kresta, J.V., Macgregor, J.F. and Marlin, T.E. (1991) 'Multivariate statistical monitoring of process operating performance', *The Canadian Journal of Chemical Engineering*, 69(1), pp. 35-47.

Ku, W., Storer, R. H., and Georgakis, C. (1995) 'Disturbance detection and isolation by dynamic principal component analysis' *Chemometrics and intelligent laboratory systems*, 30(1), pp. 179-196.

Li, Z., Wang, Y. and Wang, K. (2017) 'A data-driven method based on deep belief networks for backlash error prediction in machining centers', *Journal of Intelligent Manufacturing*, 31(7), pp. 1693-1705.

Li, F., Chen, Y., Wang, J., Zhou, X., and Tang, B. (2019) 'A reinforcement learning unit matching recurrent neural network for the state trend prediction of rolling bearings', *Measurement*, 145, pp. 191-203.

Lim, P., Goh, C.K., Tan, K.C. and Dutta, P. (2017) 'Multimodal Degradation Prognostics Based on Switching Kalman Filter Ensemble', *IEEE Trans Neural Netw*

Learn Syst, 28(1), pp. 136-148.

Liu, F., Liu, Y., Chen, F. and He, B. (2015) 'Residual life prediction for ball bearings based on joint approximate diagonalization of eigen matrices and extreme learning machine', *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 231(9), pp. 1699-1711.

Liu, H., Fan, M., Zeng, Q. and Shen, X. (2010) 'RBF network based on artificial immune algorithm and application of predicting the residual life of injecting water pipeline', *2010 Sixth International Conference on Natural Computation*. 10-12 Aug, pp. 1305-1309.

Liu, H., Zhou, J., Zheng, Y., Jiang, W. and Zhang, Y. (2018) 'Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders', *ISA Trans*, 77, pp. 167-178.

Liu, Q., Qin, S.J. and Chai, T. (2013) 'Decentralized Fault Diagnosis of Continuous Annealing Processes Based on Multilevel PCA', *IEEE Transactions on Automation Science and Engineering*, 10(3), pp. 687-698.

Liu, Y., Xiao, H., Pan, Y., Huang, D. and Wang, Q. (2016b) 'Development of multiple-step soft-sensors using a Gaussian process model with application for fault prognosis', *Chemometrics and Intelligent Laboratory Systems*, 157, pp. 85-95.

Long, B., Wu, K., Li, P. and Li, M. (2022) 'A Novel Remaining Useful Life Prediction Method for Hydrogen Fuel Cells Based on the Gated Recurrent Unit Neural Network', *Applied Sciences*, 12(1).

Lu, K.S. and Saeks, R. (1979) 'Failure Prediction for an On-Line Maintenance System in a Poisson Shock Environment', *IEEE Transactions on Systems, Man, and Cybernetics*, 9(6), pp. 356-362.

MacGregor, J.F. and Kourti, T. (1995) 'Statistical process control of multivariate processes', *Control Engineering Practice*, 3(3), pp. 403-414.

Malhotra, P., Tv, V., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P. and Shroff, G. (2016) 'Multi-Sensor Prognostics using an Unsupervised Health Index based on LSTM Encoder-Decoder', *1st SIGKDD Workshop on Machine Learning for Prognostics and Health Management*.

Marcellino, M., Stock, J.H. and Watson, M.W. (2006) 'A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series', *Journal of Econometrics*, 135(1-2), pp. 499-526.

Mazaev, T., Crevecoeur, G. and Hoecke, S.V. (2021) 'Bayesian Convolutional Neural Networks for Remaining Useful Life Prognostics of Solenoid Valves With

Uncertainty Estimations', *IEEE Transactions on Industrial Informatics*, 17(12), pp. 8418-8428.

Mena, P., Borrelli, R. A., and Kerby, L. (2022) 'Nuclear Reactor Transient Diagnostics Using Classification and AutoML', *Nuclear Technology*, 208(2), pp. 232-245.

Niu, G., Tang, S. and Zhang, B. (2018) 'Machine Condition Prediction Based on Long Short Term Memory and Particle Filtering', *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. 21-23 Oct, pp. 5942-5947.

Oppenheimer, C.H. and Loparo, K.A. (2002) 'Physically based diagnosis and prognosis of cracked rotor shafts', *Component and systems diagnostics, prognostics, and health management II*, 4733, pp. 122-132.

Orchard, M.E. and Vachtsevanos, G.J. (2009) 'A particle-filtering approach for on-line fault diagnosis and failure prognosis', *Transactions of the Institute of Measurement and Control*, 31(3-4), pp. 221-246.

Pan, J., Zi, Y., Chen, J., Zhou, Z. and Wang, B. (2018) 'LiftingNet: A Novel Deep Learning Network With Layerwise Feature Learning From Noisy Mechanical Data for Fault Classification', *IEEE Transactions on Industrial Electronics*, 65(6), pp. 4973-4982.

Pan, Y., Mei, F., Miao, H., Zheng, J., Zhu, K. and Sha, H. (2019) 'An Approach for HVCB Mechanical Fault Diagnosis Based on a Deep Belief Network and a Transfer Learning Strategy', *Journal of Electrical Engineering & Technology*, 14(1), pp. 407-419.

Paris, P.C. and Erdogan, F. (1963) 'A Critical Analysis of Crack Propagation Laws', *Journal of Basic Engineering*, 85, pp. 528-533.

Pascanu, R., Mikolov, T. and Bengio, Y. (2013) 'On the difficulty of training recurrent neural networks', *Proceedings of the 30th International Conference on Machine Learning*. 28(3), pp. 1310-1318.

Pelayo, E.E.M., Yang, J. and Yu, X. (2017) 'Research on modification of residual oil by chemical additive', *Petroleum Processing and Petrochemicals*, 48(7), pp. 22-27.

Peng, Y., Cheng, F., Qiao, W. and Qu, L. (2017) *2017 IEEE International Conference on Electro Information Technology (EIT)*. 14-17 May 2017.

Penha, R., Ipen, E. and Hines, J. (2001) 'Using Principal Component Analysis Modeling to Monitor Temperature Sensors in a Nuclear Research Reactor'.

Powell, M.J.D. (1987) 'Radial basis functions for multivariable interpolation: a

review', in *Algorithms for approximation*. Clarendon Press, pp. 143–167.

Qin, S.J. (2012) 'Survey on data-driven industrial process monitoring and diagnosis', *Annual Reviews in Control*, 36(2), pp. 220-234.

Qin, S.J., Valle, S. and Piovoso, M.J. (2001) 'On unifying multiblock analysis with application to decentralized process monitoring', *Journal of Chemometrics*, 15(9), pp. 715-742.

Rathnapriya, S. and Manikandan, V. (2020) 'Remaining Useful Life Prediction of Analog Circuit Using Improved Unscented Particle Filter', *Journal of Electronic Testing*, 36(2), pp. 169-181.

Ravikumar, K. N., Madhusudana, C. K., Kumar, H., and Gangadharan, K. V. (2022) 'Classification of gear faults in internal combustion (IC) engine gearbox using discrete wavelet transform features and K star algorithm', *Engineering Science and Technology, an International Journal*, 30, p. 101048.

Reddy, K. and Yarrakula, K. (2016) 'Analysis of accidents in chemical process industries in the period 1998-2015', *International Journal of ChemTech Research*, 9, pp. 177-191.

Reis, M. and Gins, G. (2017a) 'Industrial Process Monitoring in the Big Data/Industry 4.0 Era: from Detection, to Diagnosis, to Prognosis', *Processes*, 5(4).

Reis, M. and Gins, G. (2017b) 'Industrial Process Monitoring in the Big Data/Industry 4.0 Era: from Detection, to Diagnosis, to Prognosis', *Processes*, 5, p. 35.

Ren, L., Sun, Y., Wang, H. and Zhang, L. (2018) 'Prediction of Bearing Remaining Useful Life With Deep Convolution Neural Network', *IEEE Access*, 6, pp. 13041-13049.

Russo, L., Sarda, K., Glielmo, L., and Acernese, A. (2021) 'Fault detection and diagnosis in steel industry: a one class-support vector machine approach', *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2304-2309.

Severson, K., Chaiwatanodom, P. and Braatz, R.D. (2016) 'Perspectives on process monitoring of industrial systems', *Annual Reviews in Control*, 42, pp. 190-200.

Shao, H., Jiang, H., Wang, F. and Wang, Y. (2017) 'Rolling bearing fault diagnosis using adaptive deep belief network with dual-tree complex wavelet packet', *ISA Trans*, 69, pp. 187-201.

Shao, Y. and Nezu, K. (2000) 'Prognosis of remaining bearing life using neural networks', *Proceedings of The Institution of Mechanical Engineers Part I-journal*

of Systems and Control Engineering - PROC INST MECH ENG I-J SYST C, 214, pp. 217-230.

Sohaib, M., Kim, C.-H. and Kim, J.-M. (2017) 'A Hybrid Feature Model and Deep-Learning-Based Bearing Fault Diagnosis', *Sensors*, 17(12).

Tang, D., Cao, J. and Yu, J. (2019) 'Remaining useful life prediction for engineering systems under dynamic operational conditions: A semi-Markov decision process-based approach', *Chinese Journal of Aeronautics*, 32(3), pp. 627-638.

Sun, S., Song, X., Liu, Z., Tian, W., and Li, C. (2020) 'A combination of dynamic simulation and dynamic time warping for fault diagnosis of chemical process startups', *Chemical Engineering Transactions*, 81, pp. 301-306.

Tang, S., Shen, C., Wang, D., Li, S., Huang, W. and Zhu, Z. (2018) 'Adaptive deep feature learning network with Nesterov momentum and its application to rotating machinery fault diagnosis', *Neurocomputing*, 305, pp. 1-14.

'TCE: Asia: Sinopec publicly reprimanded over pipeline blast', (2014) *TCE The Chemical Engineer*, (872), p. 16.

Taqvi, S. A., Tufa, L. D., Zabiri, H., Maulud, A. S., and Uddin, F. (2018) 'Multiple fault diagnosis in distillation column using multikernel support vector machine', *Industrial & Engineering Chemistry Research*, 57(43), pp. 14689-14706.

Tobon-Mejia, D.A., Medjaher, K., Zerhouni, N. and Tripot, G. (2011) *2011 Prognostics and System Health Managment Confernece*. 24-25 May 2011.

Tong, C., Palazoglu, A. and Yan, X. (2013) 'An adaptive multimode process monitoring strategy based on mode clustering and mode unfolding', *Journal of Process Control*, 23(10), pp. 1497-1507.

Uren, K., Schoor, G., Rand, C. and Botha, A. (2015) 'An integrated approach to sensor FDI and signal reconstruction in HTGRs – Part I: Theoretical framework', *Annals of Nuclear Energy*, 87.

Uren, K.R., van Schoor, G., du Rand, C.P. and Botha, A. (2016) 'An integrated approach to sensor FDI and signal reconstruction in HTGRs – Part II: Case studies', *Annals of Nuclear Energy*, 87, pp. 739-749.

Venkatasubramanian, V., Rengaswamy, R. and Kavuri, S.N. (2003a) 'A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies', *Computers & Chemical Engineering*, 27(3), pp. 313-326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. and Yin, K. (2003b) 'A review of process fault detection and diagnosis: Part III: Process history based methods', *Computers & Chemical Engineering*, 27(3), pp. 327-346.

Venkatasubramanian, V., Rengaswamy, R., Yin, K. and Kavuri, S.N. (2003c) 'A review of process fault detection and diagnosis Part I Quantitative model-based methods', *Computers & Chemical Engineering*, 27(3), pp. 293-311.

Venkateswarlu, Ch., Gangiah, K. and Rao M.Bhagavantha (1992) 'Two-level methods for incipient fault diagnosis in nonlinear chemical processes', *Computers & Chemical Engineering*, 16(5), pp. 463-476.

Wang, X., Kruger, U., Irwin, G.W., McCullough, G. and McDowell, N. (2008) 'Nonlinear PCA With the Local Approach for Diesel Engine Fault Detection and Diagnosis', *IEEE Transactions on Control Systems Technology*, 16(1), pp. 122-129.

Wang, S., Xiang, J., Zhong, Y. and Zhou, Y., (2018) 'Convolutional neural network-based hidden Markov models for rolling element bearing fault identification', *Knowledge-Based Systems*, 144, pp.65-76.

Wang, L. (2018) 'Enhanced fault detection for nonlinear processes using modified kernel partial least squares and the statistical local approach'. *The Canadian Journal of Chemical Engineering*, 96(5), pp. 1116-1126.

Wen, Q., Ge, Z. and Song, Z. (2012) 'Nonlinear dynamic process monitoring based on kernel partial least squares', *Proceedings of the American Control Conference*, pp. 6650-6654.

Wise, B., Ricker, N., Veltkamp, D. and Kowalski, B. (1990) 'A Theoretical Basis for the Use of Principal Component Models for Monitoring Multivariate Processes', *Process Control and Quality*, 1.

Wold, Herman (1973) 'Nonlinear iterative partial least squares (NIPALS) modelling: some current developments', *Multivariate analysis–III. Academic Press*, pp. 383-407.

Yan, J., Guo, C. and Wang, X. (2011) 'A dynamic multi-scale Markov model based methodology for remaining life prediction', *Mechanical Systems and Signal Processing*, 25(4), pp. 1364-1376.

Yan, J., Koç, M. and Lee, J. (2004) 'A prognostic algorithm for machine performance assessment and its application', *Production Planning & Control*, 15(8), pp. 796-801.

Yan, Z., Yao, Y., Huang, T.-B. and Wong, Y.-S. (2018) 'Reconstruction-Based Multivariate Process Fault Isolation Using Bayesian Lasso', *Industrial & Engineering Chemistry Research*, 57(30), pp. 9779-9787.

Yang, S.K. and Liu, T.S. (1999) 'State estimation for predictive maintenance using Kalman filter', *Reliability Engineering & System Safety*, 66(1), pp. 29-39.

Yang, Z., Baraldi, P. and Zio, E. (2016b) *2016 Prognostics and System Health Management Conference (PHM-Chengdu)*. 19-21 Oct. 2016.

Yao, Y. and Gao, F. (2009) 'A survey on multistage/multiphase statistical modeling methods for batch processes', *Annual Reviews in Control*, 33(2), pp. 172-183.

Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G. and Fei-Fei, L. (2018) 'Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos', *International Journal of Computer Vision*, 126(2), pp. 375-389.

Yi, J., Huang, D., He, H., Zhou, W., Han, Q. and Li, T. (2017) 'A Novel Framework for Fault Diagnosis Using Kernel Partial Least Squares Based on an Optimal Preference Matrix', *IEEE Transactions on Industrial Electronics*, 64(5), pp. 4315-4324.

Yin, S., Ding, S.X., Xie, X. and Luo, H. (2014a) 'A Review on Basic Data-Driven Approaches for Industrial Process Monitoring', *IEEE Transactions on Industrial Electronics*, 61(11), pp. 6418-6428.

Yin, S., Luo, H. and Ding, S.X. (2014b) 'Real-Time Implementation of Fault-Tolerant Control Systems With Performance Optimization', *IEEE Transactions on Industrial Electronics*, 61(5), pp. 2402-2411.

Yoon, S. and MacGregor, J.F. (2001) 'Fault diagnosis with multivariate statistical models part I: using steady state fault signatures', *Journal of Process Control*, 11(4), pp. 387-400.

Young, T., Hazarika, D., Poria, S. and Cambria, E. (2018) 'Recent Trends in Deep Learning Based Natural Language Processing [Review Article]', *IEEE Computational Intelligence Magazine*, 13(3), pp. 55-75.

Yu, H., Tian, Z., Li, H., Xu, B., and An, G. (2020) 'A novel deep belief network model constructed by improved conditional RBMs and its application in RUL prediction for hydraulic pumps'. *Int. J. Acoust. Vib*, 25(3), pp. 373-383.

Yunus, M.Y.M., Zhang, J. (2010) 'Multivariate Process Monitoring Using Classical Multidimensional Scaling and Procrustes Analysis', *IFAC Proceeding*, 43(5), pp. 165-170.

Yue, H.H. and Qin, S.J. (2001) 'Reconstruction-Based Fault Identification Using a Combined Index', *Industrial & Engineering Chemistry Research*, 40(20), pp. 4403-4414.

Zarei, J. and Shokri, E. (2014) 'Robust sensor fault detection based on nonlinear unknown input observer', *Measurement*, 48, pp. 355-367.

Zhang, C., Lim, P., Qin, A.K. and Tan, K.C. (2017) 'Multiobjective Deep Belief

Networks Ensemble for Remaining Useful Life Estimation in Prognostics', *IEEE Trans Neural Netw Learn Syst*, 28(10), pp. 2306-2318.

Zhang, D., Bi, G., Sun, Z. and Guo, Y. (2015) 'Online monitoring of precision optics grinding using acoustic emission based on support vector machine', *The International Journal of Advanced Manufacturing Technology*, 80(5), pp. 761-774.

Zhang, P., Wang, G. and Zhou, D.D. (2000) 'Fault diagnosis methods for dynamic systems', *Kongzhi Lilun Yu Yingyong/Control Theory and Applications*, 17, pp. 155-158.

Zhang, X., Polycarpou, M.M. and Parisini, T. (2002) 'A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems', *IEEE Transactions on Automatic Control*, 47(4), pp. 576-593.

Zhang, Y. and Qin, J. (2007) 'Fault Detection of Nonlinear Processes Using Multiway Kernel Independent Component Analysis', *Industrial & Engineering Chemistry Research - IND ENG CHEM RES*, 46.

Zhang, Y., Xiong, R., He, H. and Pecht, M.G. (2018) 'Long Short-Term Memory Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-Ion Batteries', *IEEE Transactions on Vehicular Technology*, 67(7), pp. 5695-5705.

Zhang, Y., Yang, N. and Li, S. (2014) 'Fault Isolation of Nonlinear Processes Based on Fault Directions and Features', *Control Systems Technology, IEEE Transactions on*, 22, pp. 1567-1572.

Zhang, Z. and Zhao, J. (2017) 'A deep belief network based fault diagnosis model for complex chemical processes', *Computers & Chemical Engineering*, 107, pp. 395-407.

Zhao, C. and Sun, Y. (2014) 'Multispace Total Projection to Latent Structures and its Application to Online Process Monitoring', *IEEE Transactions on Control Systems Technology*, 22(3), pp. 868-883.

Zhao, F., Tian, Z. and Zeng, Y. (2013) 'Uncertainty Quantification in Gear Remaining Useful Life Prediction Through an Integrated Prognostics Method', *IEEE Transactions on Reliability*, 62(1), pp. 146-159.

Zhong, K., Han, M. and Han, B. (2020) 'Data-driven based fault prognosis for industrial systems: a concise overview', *IEEE/CAA Journal of Automatica Sinica*, 7(2), pp. 330-345.

Zhou, B., Ye, H., Zhang, H., and Li, M. (2017) 'A new qualitative trend analysis algorithm based on global polynomial fit', *AIChE Journal*, 63(8), pp. 3374-3383.

Zinn, J., Vogel-Heuser, B. and Ocker, F. (2020) 'A concept for fault diagnosis

combining case-based reasoning with topological system models', *IFAC-PapersOnLine*. pp. 8217-8224.

Chatifield, C. and Xing, H. (2019) 'The analysis of time series: an introduction with R', CRC press.

Berbache, S., Harkat, M. F., and Kratz, F. (2019) 'Sensor fault detection and isolation techniques based on PCA', *2019 international conference on advanced electrical engineering (ICAEE)*, pp. 1-7.

Dong, Y., and Qin, S. J. (2018). 'A novel dynamic PCA algorithm for dynamic data modeling and process monitoring', *Journal of Process Control*, 67, pp. 1-11.

Bakdi, A., Kouadri, A., and Bensmail, A. (2017). 'Fault detection and diagnosis in a cement rotary kiln using PCA with EWMA-based adaptive threshold monitoring scheme', *Control Engineering Practice*, 66, pp. 64-75.