MACHINE LEARNING METHODS FOR FAULT DETECTION AND DIAGNOSIS OF DIGITALIZED PROCESSING SYSTEM

By

©Rajeevan Arunthavanathan

A thesis submitted to the

School of Graduate Studies

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Faculty of Engineering and Applied Science

Memorial University of Newfoundland

May 2022

St.John's Newfoundland

Dedication

This thesis is dedicated to the memory of my mother, Suganthy Arunthavanthan, who always believed in my ability to be successful in the academic arena. You left me a few months back, but your faith made this journey possible, Amma.

Abstract

This thesis presents novel development and applications of machine learning techniques for process fault detection, diagnosis, and prognosis from safety and predictive maintenance perspectives. The main contributions of this thesis include the development of (i) new algorithms to diagnose the unlabelled faults; (ii) a self-learning tool for fault detection and diagnosis of untrained faults; (iii) a forecast model for fault conditions; (iv) a framework for root cause analysis in an automated environment; and (v) a methodology for estimating the remaining useful life.

In the context of Industry 4.0, process plants' operations have become increasingly autonomous and run in an intelligent mode. An intelligent process operation takes advantage of online data, uses advanced modelling approaches and utilizes automation to achieve a flexible, smart, and reconfigurable operation. In such an autonomous environment, process fault detection, diagnosis, and prognosis play critical roles in ensuring its safety and integrity. In this study intelligent fault detection and diagnosis methods are developed based on state-of-the-art machine learning techniques. Further, this study is extended to calculate the remaining useful life online, using the fault to failure transmission time.

The research study results in five signification contributions. First, a comprehensive review of the existing fault detection and diagnosis approaches was conducted to identify the knowledge gaps and to develop fault detection and diagnosis approaches that are best suited for Industry 4.0. Second, a cognitive fault detection and diagnosis technique using unlabelled process data and an anomaly detection technique using machine learning were developed. Third, a self-learning neural network and permutation algorithm were developed for prediction of the root cause of a detected fault. Fourth, a methodology was developed to early predict faults, based on monitoring the fault

symptoms using a deep learning algorithm. Fifth, a model was developed to estimate the remaining useful life using the system's failure threshold and a degradation model.

In this research work, all the proposed models were developed using self-learning methodologies. Therefore, the work constitutes an essential step towards developing an autonomous fault detection, diagnosis, and remaining useful life estimation tool. The proposed frameworks are validated using experimental data and simulated process system data.

The findings from this study highlight that by integrating unsupervised and supervised learning, without prior knowledge of the fault condition, the proposed machine learning model was able to detect and diagnose the fault conditions. Unsupervised learning was used to detect the unknown fault conditions, and a neural network permutation algorithm was used to identify the root cause for the detected unknown faults. This work also used supervised learning to classify the known fault conditions. Furthermore, by investigating the failure condition of the identified root cause variable or feature, remaining useful life was estimated by developing a regression model. Likewise, this thesis finds the solution for early fault detection in real-time by integrating the deep learning tools with unsupervised learning.

Acknowledgments

I would like to sincerely express my appreciation and gratitude to my supervisor Dr. Faisal Khan and co-supervisors, Dr. Salim Ahamed and Dr. Syed Imtiaz, for their continuous support and constant encouragement to accomplish all the objectives to fulfill my Ph.D. work. Their enormous research experience, knowledge, and diverse interests helped me to complete the work by exploring different engineering disciplines. I will always appreciate the way they guided me throughout the entire four years.

I would also like to thank all my fellow colleagues at the Centre for Risk Integrity and Safety Engineering (C-RISE) at the Memorial University of Newfoundland for their valuable help and suggestions for improving this work. Sincere thanks to Mihiran Pathmika, Md Alauddin, Md Tanjin, Mohammad Zaid Kamil, Mohammad Asif and Rioshar Yarviesy for their continued support and valuable advice on graduate work topics, as well as their accompaniment in all the hardest and happiest moments during these four years. Thanks to the Natural Science and Engineering Council of Canada (NSERC) of Canada and, the Canada Research Chair (CRC) Tier I Program for the collaborative research and their financial support.

I must express my thank and gratitude to my loving and supportive wife Amuthajini, and my loving daughter Ruthwisaa, who has grown up with all my thesis work, and the sacrifices and commitments both of them made during my studies. Finally, I would like to extend my deepest and most sincere thanks to my parents, whose love and guidance are with me in whatever I pursue. I heartily thank my loving mother, who gave continuous guidance, support, strength, advice and love until her last minute.

Table of Contents

Table of Figures

List of Tables

List of Abbreviation

AI	Artificial Intelligent
ASM	Abnormal Situation Management
BN	Bayesian Network
BR	Bayesian Regression
CM	Condition Monitoring
CNN	Convolutional Neural Network
DL	Deep Learning
FDD	Fault Detection and Diagnosis
HMM	Hidden Markov Model
LSTM	Long Short-Term Memory
ML	Machine Learning
NN	Neural Network
OC-SVM	One class SVM
ONCNN	One class Neural Network
PCA	Principal Component Analysis
PM	Predictive Maintenance
RUL	Remaining useful life
RNN	Recurrent Neural Network
RA	Risk Assessment
SVM	Support Vector Machine
SIS	Safety Instrumented System
TE	Tennessee Eastman Process

Authorship Statement

This thesis written in the manuscript format. Most chapters of this thesis are either published or submitted in a peer-reviewed Journal. The contribution of the authorship for each manuscript is presented below.

Chapter 2

R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, "An analysis of process fault diagnosis methods from safety perspectives," Computers and Chemical Engineering, vol. 145. Elsevier Ltd, p. 107197, Feb. 01, 2021, doi: 10.1016/j.compchemeng.2020.107197.

I am the primary author of the paper. The work is conducted under the supervision of the coauthors, Dr. Faisal Khan, Dr. Salim Ahmed, and Dr. Syed Imtiaz. I carried out the major part, including conceptualization, developing the review question, reviewing the relevant literature, and preparing the first draft of the paper, and subsequently revising the manuscript based on the coauthors' and reviewers' comments in the peer-reviewing process. The coauthor Faisal Khan helped with conceptualization, developing the research questions, analyzing the results, and discussion. Also, co-authors Salim Ahmed and Syed Imtiaz contributed through support in the discussion. While I wrote the first draft of the manuscript, coauthors assisted in reviewing and revising the manuscript and managed the peer-review process.

Chapter 3

R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, "Fault detection and diagnosis in process system using the artificial intelligence-based cognitive technique," Computers and Chemical Engineering, vol. 134, p. 106697, Mar. 2020, doi: 10.1016/j.compchemeng.2019.106697.

I am the principal author of the paper. The work is conducted under the supervision of the coauthors, Dr. Faisal Khan, Dr. Salim Ahmed, and Dr. Syed Imtiaz. With the help of Dr. Faisal Khan, I have conceptualized the problem and developed a solution strategy. I have reviewed the relevant literature, and developed the methodology to solve the problem, also prepared the first draft of the paper, and subsequently revised the manuscript based on the coauthors' and reviewers' comments in the peer-reviewing process. The coauthor, Dr. Faisal Khan, helped analyze the results and discussion. Co-authors Drs. Salim Ahmed, Syed Imtiaz, and Riza Rusil contributed to the methodology development and discussion. While I wrote the first draft of the manuscript, coauthors assisted in reviewing and revising the manuscript and managed the peer-review process.

Chapter 4

R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, “Autonomous Fault Diagnosis and Root Cause Analysis for the Processing System Using One-Class SVM and NN Permutation Algorithm,” Industrial & Engineering Chemistry Research, 2022, doi: 10.1021/acs.iecr.1c02731.

I am the primary author of the paper. The work is conducted with the supervision of the coauthors, Dr. Faisal Khan, Dr. Salim Ahmed, and Dr. Syed Imtiaz. With the help and support of co-authors Dr. Faisal Kahn, I carried out the conceptualization, developed the methodology and the algorithm. I reviewed the relevant literature, prepared the first draft of the paper, and subsequently revised the manuscript based on the coauthor's and reviewers' comments in the peer-review process. The coauthor Dr. Khan helped to analyze the results and discussion. Also, co-authors Salim Ahmed and Syed Imtiaz contributed to the methodology development, results, and discussion. While I wrote the first draft of the manuscript, the coauthors assisted in reviewing and revising the manuscript and managed the peer-review process.

Chapter 5

R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, "A deep learning model for process fault prognosis," Process Safety and Environmental Protection, Aug. 2021, doi: 10.1016/J.PSEP.2021.08.022.

I am the primary author of the paper. The work is conducted under the supervision of the coauthors, Dr. Faisal Khan, Dr. Salim Ahmed, and Dr. Syed Imtiaz. With the help and support of co-author Dr. Faisal Khan, I conceptualized the problem and developed the methodology. I reviewed the relevant literature, developed the computer-based algorithm, prepared the first draft of the paper, and subsequently revised the manuscript based on the coauthor's and reviewers' comments in the peer-review process. The coauthor Dr. Khan helped to analyze the results. Also, co-authors Salim Ahmed and Syed Imtiaz contributed to the algorithm development, results analysis, and discussion. While I wrote the first draft of the manuscript, the coauthors assisted in reviewing and revising the manuscript and managed the peer-review process.

Chapter 6

R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, "Online remaining useful life estimation using fault to failure transformation in process systems," IEEE systems journal. (Under Review)

I am the principal author of the paper. The work is conducted with the supervision of the coauthors, Dr. Faisal Khan, Dr. Salim Ahmed, and Dr. Syed Imtiaz. With the help and support of co-author

Dr. Faisal Khan, I carried out the conceptualization and developed a computer-based algorithm. I reviewed the relevant literature, prepared the first draft of the paper, and subsequently revised the manuscript based on the coauthor's and reviewers' comments in the peer-review process. The coauthor Dr. Khan helped to analyze results and discussion. Also, co-authors Salim Ahmed and Syed Imtiaz contributed to the methodology development. While I wrote the first draft of the manuscript, the coauthors assisted in reviewing and revising the manuscript and managed the peer-review process.

Chapter 1 Introduction

1.1 The Rationale of the Research

Fault detection, diagnosis, and prognosis remain as key components in ensuring process safety; however, they pose significant challenges for the process industries. Digital systems are widely used to assist in process safety and abnormal situation management (ASM) throughout the life cycle of process plants. In recent years, there has been a plethora of promotion of digitalization, machine learning, and digital twins due to the transformative capability that exists within these technologies to improve the operational performance of automated fault detection and diagnosis (FDD) and process safety [1].

Basic process control became highly automated with the third industrial revolution [2]. However, in the past decade, automation in FDD has not been realized significantly, and most of the process industries still rely on human operators for the decisions regarding abnormal situation management and the shutdown of systems' operations. With the forthcoming industrial revolution and smart process plants, providing appropriate, reliable, and automatic decision support to the operators about ASM will be an important factor. Therefore, accurate and faster automated FDD and root cause analysis for the detected fault condition are becoming important research areas in Industry 4.0 process safety.

In the past decade, physical models, data-driven models, and knowledge-based models were widely used for process systems' monitoring and fault diagnosis [3]–[5]. However, model-based approaches may lead to difficulties due to the industrial process's complexity and extreme dimensionality, especially when the process system's or sub-system's mathematical model is unavailable [2]. Moreover, the knowledge-based model requires human experts in the industry to develop and initially train the model to define the normal and abnormal conditions of the process [6]. The significant involvement of humans in developing these models makes their evolution incompatible with Industry 4.0 concepts.

In the complex process system, data-driven approaches do not rely on a physical system but require good quality historical data. For Industry 4.0 and digitalized process systems, with the recent use of the IoT and the big data concept, collecting a larger amount of high-quality data is not a complex task. However, in the past decades, most of the data-driven approaches have mainly focused on supervised learning methods, for which the historical data, containing enough information about the normal and abnormal operations, are required.

To develop an automated FDD and ASM framework, detecting abnormal conditions from unsupervised data will be an initial challenge in the data-driven approaches [7]. Moreover, further classifying the different fault conditions and determining the root cause of the detected abnormal event will be ongoing research.

For automated FDD, it is important to update the model in an online fashion; the model should learn the new fault scenarios without any human interaction. Chen et al. and Arunthavanathan et al. attempted to develop cognitive models to learn fault signatures from data [8], [9][1]. From their view, traditional statistical approaches are insufficient to develop cognitive models. Moreover, statistical models are inefficient for self-updating. Therefore, machine learning approaches will play a vital role in the upcoming process of FDD.

[1] Reference [9] is based on the result of this thesis.

Furthermore, it is also imperative to detect a fault condition as early as possible by using the fault symptoms along with autonomous detection. In this way, the FDD method is capable of diagnosing and taking mitigating actions before a fault escalates to failure and causes harmful accidents.

Advancing autonomous fault detection and diagnosis for Industry 4.0 has been reviewed recently [7], [10]–[13]. From these reviews, it can be concluded that employing machine learning approaches instead of the statistical FDD methods may help to develop smart FDD methods more suitable to Industry 4.0.

Therefore, this research is mainly focused on identifying the knowledge gaps, concentrating on automated FDD from safety and integrity perspectives. The research theme is developed to answer the following key questions:

i. What are the important factors influencing FDD methodologies due to the digitalization of process systems and the advent of Industry 4.0?

ii. How can machine learning techniques be used to predict new faults from unlabelled data?

iii. How can accurate self-learning models be developed to investigate and diagnose the fault condition?

iv. How can the root cause for the detected fault condition be analyzed using machine learning algorithms?

v. How effective is it to develop a fault prognosis model by investigating fault symptoms in the process model?

vi. Can machine learning techniques be used for predicting the remaining useful life by analyzing fault to failure transmission?

The overall goal of the work is to develop knowledge to enable self-learning FDD models which can serve Industry 4.0 transformation. Future research works can focus on interfacing these models with online process data, with the development of appropriate internet of things sensor interfaces and big data storage concepts. This thesis will further the development of Industry 4.0 concepts.

1.2 Objectives

The aim of the research is to identify the gap between current FDD approaches and future needs for a digitalized process system in the process industry and to develop automated FDD approaches to detect, diagnose, and find the cause of a fault condition. Further, this work investigates the possibility of fault prognosis and preventive maintenance by estimating the failure condition early and evaluating the remaining useful life of the system. To reach the goal, the following objectives have been set for the research.

Objective 1: Investigate the past, and recent FDD approaches in process safety and identify the required changes in the current FDD models to be applied in digitalized process systems and the Industry 4.0 smartness concept.

Objective 2: Propose a self-updating machine learning approach trained using unlabelled data to detect and diagnose new process faults.

Objective 3: Develop a cognitive fault detection and root cause analysis framework in an automated environment with less computational time.

Objective 4: Early detection of the faults by capturing the early fault symptoms using deep learning techniques.

Objective 5: Develop a remaining useful life (RUL) prediction algorithm to estimate the RUL by evaluating the fault to failure time.

The objectives of the project and the defined research questions of the research are summarized in Figure 1.1.

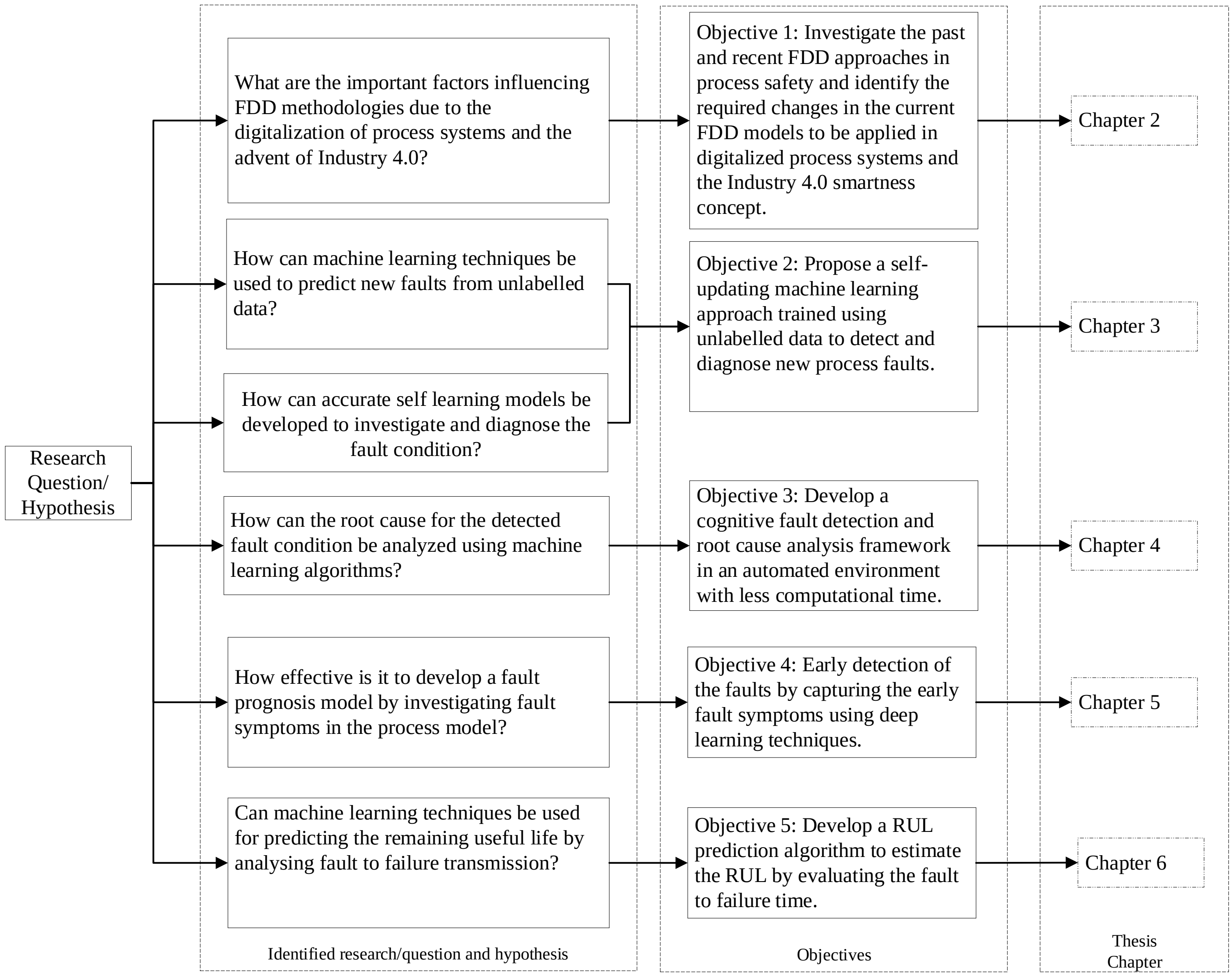


Figure 1.1 Identified research question and objectives of the research.

1.3 Scope and Limitation

The proposed FDD model development and investigation are based on multivariate simulated process data due to the unavailability and inaccessibility of real-time data from industrial plants. However, experimental laboratory system data are used to test the developed methods. Table 1.1 summarize the models developed for each objective and data used to test the objectives.

Table 1.1 : Research Summary.

Objective	Proposed Model	Case study
1. Identify the knowledge gab between recent FDD models and requirement in Industry 4.0.	N/A	N/A
2. Develop a self-learning model to detect and classify the fault condition using unlabeled data.	Shallow NN, One-class NN	RT 580 experimental setup + TE process data.
3. Develop a cognitive fault detection and root cause approach using unlabeled data.	One-class SVM, Shallow NN, and Permutation algorithm	Continuous stirred tank heater (CSTH) + TE process.
4. Develop an early detection of the fault condition by capturing the early fault symptoms.	CNN, LSTM, and one class SVM	TE process data.
5. Develop a RUL prediction algorithm online.	One Class SVM, NN, and Bayesian Linear Regression.	TE process data.

The proposed FDD methods have been developed and tested in the python environment. Therefore, the computational process highly relies on computer specification. In this work, an Intel Core i7, 3GHz processor with 8GB RAM, has been used in the development and testing. The models are expected to perform better with high-performance processors specially designed for artificial intelligence, such as tensor processors.

1.4 Organization of the Thesis

This thesis is written in the manuscript format and combines five peer-reviewed journal papers. The rest of the thesis is organized as follows:

Chapter 2 investigates the current approaches in the process of FDD and their integration into ASM. Further, the knowledge gap between the current FDD methods and that required for digitalized process safety in Industry 4.0 is investigated. This chapter was published in the Computers and Chemical Engineering Journal.

Chapter 3 proposes a cognitive FDD methodology using a machine learning algorithm to detect the unknown fault conditions and classify the known fault conditions using unsupervised data. A one-class NN method is proposed to detect a fault using unlabelled data and increment output layered neural network proposed to classify the fault condition. This chapter also highlights the self-learning model, which will continually monitor the system performance and update itself when a new fault occurs. RT 580 Experimental data and Tennessee Eastman process data are used to validate the proposed algorithm. This chapter was published in the Computers and Chemical Engineering Journal.

Chapter 4 presents a methodology to analyze the root cause of a detected fault condition. In this work, the neural network permutation algorithm is integrated with a shallow neural network for fault classification and root cause analysis. Also, one-class SVM is proposed to detect the unknown fault condition. Data from a Continuous Stirred-Tank Heater (CSTH) and the Tennessee Eastman process are used to validate the proposed method. This chapter was published in the Industrial and Engineering Chemistry Research Journal.

Chapter 5 proposes a deep learning approach to early detect incipient faults by examining the fault symptoms in a multivariate process. The hybrid convolutional neural network (CNN) and the long short-term memory (LSTM) model are used to forecast the data; CNN is used to extract the input features from time-dependent data, and LSTM is used to forecast the fault condition. Also, the one-class- support vector machines (OC-SVM) is proposed to predict the fault condition. Tennessee Eastman process data are used to validate the proposed model. This chapter was published in Process Safety and Environmental Protection Journal.

Chapter 6 initiates the RUL estimation using fault to failure deviation. FDD and root cause analysis methods are similar to those discussed in chapters 3 and 4. Further, the linear Bayesian regression approach is proposed to estimate the fault to failure deviation time by analyzing faulty data patterns. Tennessee Eastman process data are used to validate the proposed model. This chapter has been submitted to IEEE access for peer review.

References

[1] S. R. Saufi, Z. A. Bin Ahmad, M. S. Leong, and M. H. Lim, "Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review," *IEEE Access*, vol. 7, pp. 122644–122662, 2019, doi: 10.1109/ACCESS.2019.2938227.

[2] V. Venkatasubramanian, "Process Fault Detection and Diagnosis: Past, Present and Future," *IFAC Proceedings Volumes*, vol. 34, no. 27, pp. 1–13, Jun. 2001, doi: 10.1016/S1474-6670(17)33563-2.

[3] M. Alauddin, F. Khan, S. Imtiaz, and S. Ahmed, "A Bibliometric Review and Analysis of Data-Driven Fault Detection and Diagnosis Methods for Process Systems," *Industrial and Engineering Chemistry Research*, vol. 57, no. 32, pp. 10719–10735, 2018, doi: 10.1021/acs.iecr.8b00936.

[4] A. Alzghoul, B. Backe, M. Löfstrand, A. Byström, and B. Liljedahl, "Comparing a knowledge-based and a data-driven method in querying data streams for system fault detection: A hydraulic drive system application," *Computers in Industry*, vol. 65, no. 8, pp. 1126–1135, Oct. 2014, doi: 10.1016/J.COMPIND.2014.06.003.

[5] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control

systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, Dec. 2008, doi: 10.1016/J.ARCONTROL.2008.03.008.

[6] C. M. Fan and Y. P. Lu, "A bayesian framework to integrate knowledge-based and data-driven inference tools for reliable yield diagnoses," *Proceedings - Winter Simulation Conference*, pp. 2323–2329, 2008, doi: 10.1109/WSC.2008.4736337.

[7] R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, "An analysis of process fault diagnosis methods from safety perspectives," *Computers and Chemical Engineering*, vol. 145. Elsevier Ltd, p. 107197, Feb. 01, 2021, doi: 10.1016/j.compchemeng.2020.107197.

[8] H. Chen, P. Tiňo, and X. Yao, "Cognitive fault diagnosis in Tennessee Eastman Process using learning in the model space," *Computers and Chemical Engineering*, vol. 67, pp. 33–42, 2014, doi: 10.1016/j.compchemeng.2014.03.015.

[9] R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, "Fault detection and diagnosis in process system using artificial intelligence-based cognitive technique," *Computers and Chemical Engineering*, vol. 134, p. 106697, Mar. 2020, doi: 10.1016/j.compchemeng.2019.106697.

[10] J. Lee, I. Cameron, and M. Hassall, "Improving process safety: What roles for digitalization and industry 4.0?," *Process Safety and Environmental Protection*, vol. 132, pp. 325–339, 2019, doi: 10.1016/j.psep.2019.10.021.

[11] M. Prist *et al.*, "Online Fault Detection: A Smart Approach for Industry 4.0," *2020 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2020 - Proceedings*, pp. 167–171, 2020, doi: 10.1109/MetroInd4.0IoT48571.2020.9138295.

[12] A. Angelopoulos *et al.*, "Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects," *Sensors (Switzerland)*, vol. 20, no. 1, pp. 1–34, 2020, doi: 10.3390/s20010109.

[13] Y. Cohen and G. Singer, "A smart process controller framework for Industry 4.0 settings," *Journal of Intelligent Manufacturing*, no. 0123456789, 2021, doi: 10.1007/s10845-021-01748-5.

Chapter 2 Analysis of Fault Detection and Diagnosis Methodologies in Process Systems.

Preface

In this chapter, detailed preliminary research has been done to analyze the past, present, and future trends in the FDD in the process industry. The review focused on the detailed interconnection with FDD, Abnormal management, and risk assessment perspective to the process system safety. The content of this chapter has been peer-reviewed and published as a manuscript in the computers and chemical engineering journal, 2021. This chapter contributes to the identified objective 1 of the thesis as mentioned in chapter 1.

Abstract

Industry 4.0 provides substantial opportunities to ensure a safer environment through online monitoring, early detection of faults, and preventing the faults to failures transitions. Decision making is an important step in abnormal situation management. Assigning risk based on the consequences may provide additional information for abnormal situation management decisions to prevent the accident before it occurs. This section analyzes the interconnections between the three essential aspects of process safety: fault detection and diagnosis (FDD), risk assessment (RA), and abnormal situation management (ASM) in the context of the current and next generation of process systems. The authors present their thoughts on research directions in process safety in Industry 4.0. This article aims to serve as a road map for the next generation of process safety research to enable safer and sustainable process operations and development.

2.1 Introduction

Modern process plants are becoming more complicated due to process units' interconnectivity resulting from plantwide control and optimization. In such a plant, control systems connect with many sensors and actuators to control plant operations. The sensors aid in monitoring process conditions while the actuators control the process by physically adjusting the system's variables. Despite these monitoring and control measures, processes can drift beyond their safe operating range due to actuator, sensor, or system faults. These faults may lead to a system failure and ultimately cause a plant accident.

In process systems, potential accidents are prevented using layers of protection. Failures in such layers increase accident probability and lead to its consequences. Hazard identification, probability assessment, and the consequences of hazardous incidents, considering the layers of protection in a system, provide an understanding of the system safety status.

Hazard identification, FDD, RA, and mitigation action play vital roles in maintaining plant safety. Many researchers have reviewed hazard identification approaches [Dunjo et al., Cameron et al., and Willey][1]–[3]. Also, there are several review articles on FDD methods, by [Gao et al., Zhong et al., Puncochar and Skach , Md Nor et al., and Hoang and Kang][4]–[8], and on RA in process systems, reviewed by [Khan and Abbasi, Khan et al., Swuste et al., and Amin et al.][9]–[11]. However, these articles focus on different process safety elements without defining the interrelation and the overall ASM process. For process safety management, the methods and models used must be analyzed in combination to obtain a holistic view of the safety management framework. Therefore, this article attempts to review and analyze process safety elements' methods and models, focusing on their interrelations. Specifically, the article focuses on addressing the following questions:

1. How can risk be used for fault diagnosis and abnormal situation management (process safety perspective)?
    a. How can fault detection and diagnosis be used from the process safety perspective?
    b. How is abnormal situation management practiced from the process safety perspective?
    c. How are FDD and ASM integrated with the safety system?
    d. What are the key knowledge and technological gaps in the preceding areas?
2. How could operational risk be a tool for process safety management for Industry 4.0?
    a. What are the available approaches for operational risk assessment?
    b. What are the potential uses of machine learning techniques in assessing operational risk?
    c. What are the key knowledge and technological gaps in implementing novel machine learning tools in process safety management?
3. What is the way forward with Industry 4.0 to make a smart process plant a safe environment?

2.1.1 Interconnection Between FDD, ASM, and RA

Based on the SAFEPROCESS committee's definition, ASM is a centralized, continuous, and comprehensive process to prevent and control the potential hazards in process systems [12]. Moreover, ASM should identify the deviation from normal operation to faulty and failure conditions and bring the system back to normal operation.

In the process industry, determining the risk margin, using appropriate modeling such as failure models, accident models, and risk models, helps to provide information to prevent the fault from becoming a failure condition.

A failure model evaluates the accident probability by determining system failure based on a data-driven or physical model approach. Similarly, the accident model relates to the causes and effects to address the consequences. However, to develop the failure model and the accident model, hazard identification will be an initial step. When fault leads to failure, the failure model and accident model can evaluate the possible hazards and consequences.

As shown in Figure 2.1, in the process industry, FDD, RA, and ASM may apply in a closed loop. Approaches used in FDD to determine the fault condition and are initiated to identify the possible hazard. The failure model and accident model evaluate the probability and consequences of the system hazard when the process systems fail to identify and control the system's fault condition by ASM. Assessing a risk margin using risk models gives feedback to ASM regarding the hazardous event. With the feedback information, ASM changes the decision to control the operation.

From the Industry 4.0 perspective, interconnecting FDD, ASM, and RA help to develop an intelligent safety system by learning the risk and taking necessary action autonomously to prevent the hazard.

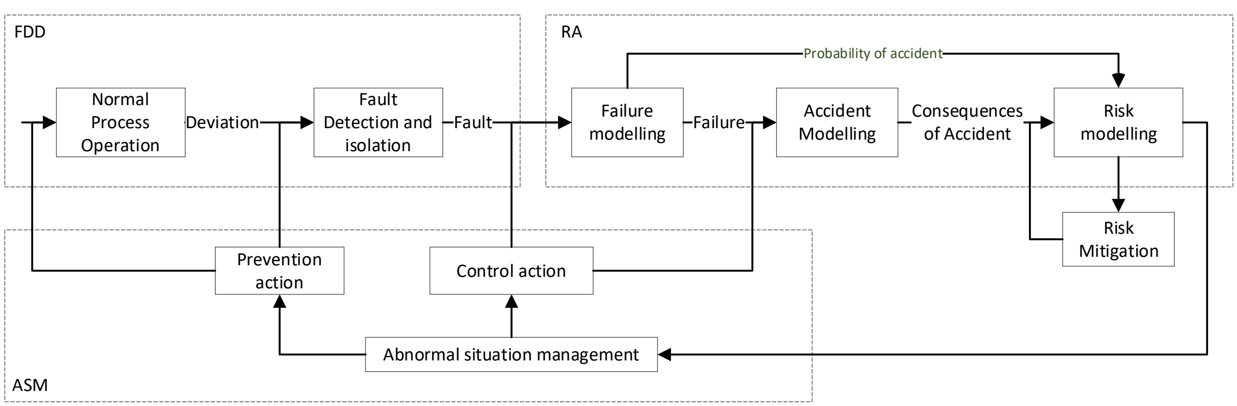


Figure 2.1 The review framework is based on the relationships among FDD, ASM, RA, and process safety.

2.1.2 Review Framework

Several review articles on the recent methods and models for FDD and RA have been published and are available in the open literature. However, these reviews have been limited in scope, mainly focusing on FDD methods. This review comprehensively studies ASM from a holistic perspective, discusses the past and present methods and techniques, and directs the future trend of process safety for Industry 4.0. The review's scope includes the topics directly related to system faults, failure analysis, RA, mitigation action, and process safety, published in journal papers. However, to discuss the FDD and ASM standards, some industrial standards and conference papers are used.

The rest of the sections are organized as follows: Sections 2.2 and 2.3 focus on past and present methods and models for FDD and ASM's role to prevent hazards in process systems. Section 2.4 summarizes the past and present risk assessment models, failure models, and hazard identification to protect the plant from accident consequences. Finally, the last section highlights the next-generation research needed for Industry 4.0 and its challenges.

2.2 FDD models and ASM Methods From a Safety Perspective.

Investigation of past accidents reveals that more than 70% of process accidents have been caused by technical and design failures, including piping system failure, deterioration of construction materials, corrosion and erosion, mass and heat transfer, and failure of the control system [13]–[15]. Moreover, the AIChE center for chemical process safety investigation reports that almost all accidents are the ultimate result of deviation from expected operations. Abnormal situation management is involved and activated when primary process control fails to protect an operation from hazardous incidents [16]–[18].

ASM does not only mean exposing the process deviation by fault detection and early warning of abnormal situations but also appropriately diagnosing the causes and making decisions to bring the process back to the normal operation. Therefore, by looking at the overall plant safety process, ASM lies between FDD and RA in the management of deviating operations. According to Dai et al. (2016), quantitative risk assessment ought to be the first essential step of ASM, to get an initial clear outline of the risk scenario to manage. FDD identifies the process deviation and diagnoses the root causes.

Basic process control became highly automated with the third industrial revolution [19]. However, in the past decade, automation in ASM has not been realized significantly, and most of the process industries still rely on human operators. With the forthcoming industrial revolution and smart process plants, providing appropriate, reliable, and automatic decision support to the operators about abnormal situations will be an important factor. This section comprehensively reviews the past and present models for fault detection and diagnosis.

2.2.1 FDD Model and Methods Review Framework and Article Selection

Most of the fault detection and diagnosis models and their applications are available in public journals, conferences' symposiums, and magazine articles. In this section, the authors have made an effort to review, categorize, and summarize the technical articles published in scientific journals. Since this review's scope is limited to process fault detection and diagnosis, six key journals with similar aims and scope are selected.

The literature survey is performed based on the keywords: fault detection and diagnosis, abnormal situation management, fault causality analysis, fault tolerance, fault prevention, routine monitoring, and preventive monitoring.

The technical articles' direct relations to the review's scopes are filtered. A simple statistical analysis is done based on Scopus, the web of science, and IEEE Xplore. Figure 2.2 summarizes the number of articles published from 1985 to the present.

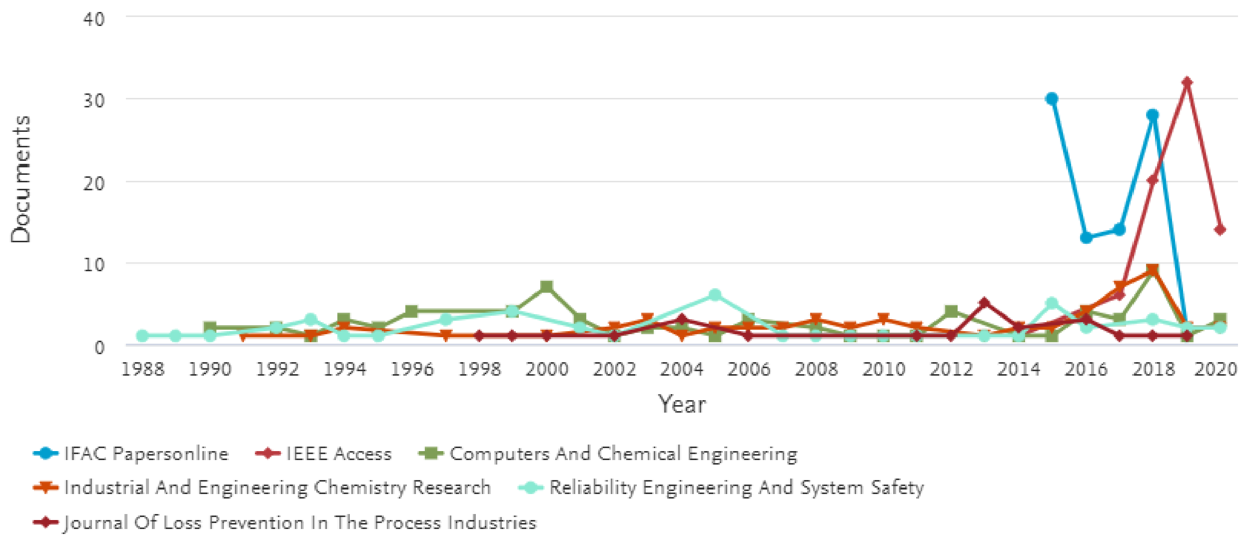


Figure 2.2 Distribution of the number of articles related to fault detection and diagnosis in the process industry over the past twenty years.

[As of 15th May 2020]

Most of the recent articles related to the focused area in process systems are published in the International Federation of Automated Control (IFAC) papers online and IEEE access. Industrial Engineering Chemistry Research (I&EC research), Computers Chemical Engineering (CCE), Reliability Engineering and System Safety (RESS), and the Journal of Loss Prevention in the Process Industries (JLP) widely focus on FDD and related areas of processing system safety. I&EC research mainly focuses on physical or chemical-based experimental, theoretical mathematical, or informative work. CCE covers the topics related mostly to process dynamics, control and monitoring, abnormal event management, and process safety. Also, CCE has published significant amount of articles related to fault detection and diagnosis and abnormal situations. JLP mainly emphasizes chemical and process plant safety. RESS is devoted to developing and applying complex technological systems' safety and reliability, mainly focusing on process industries.

2.2.2 FDD Models and Methods

Methods and models developed in the past years to detect and diagnose fault conditions use mathematical, analytical, data-driven, statistical, computational, and hybrid approaches. Based on the FDD methodology classification in the work of Chiang et al., Venkatasubramanian et al., Zhang and Jiang, Mouzakitis, and Alzghoul et al. [20]–[24], a refined classification of existing FDD methods is shown in Figure 2.3.

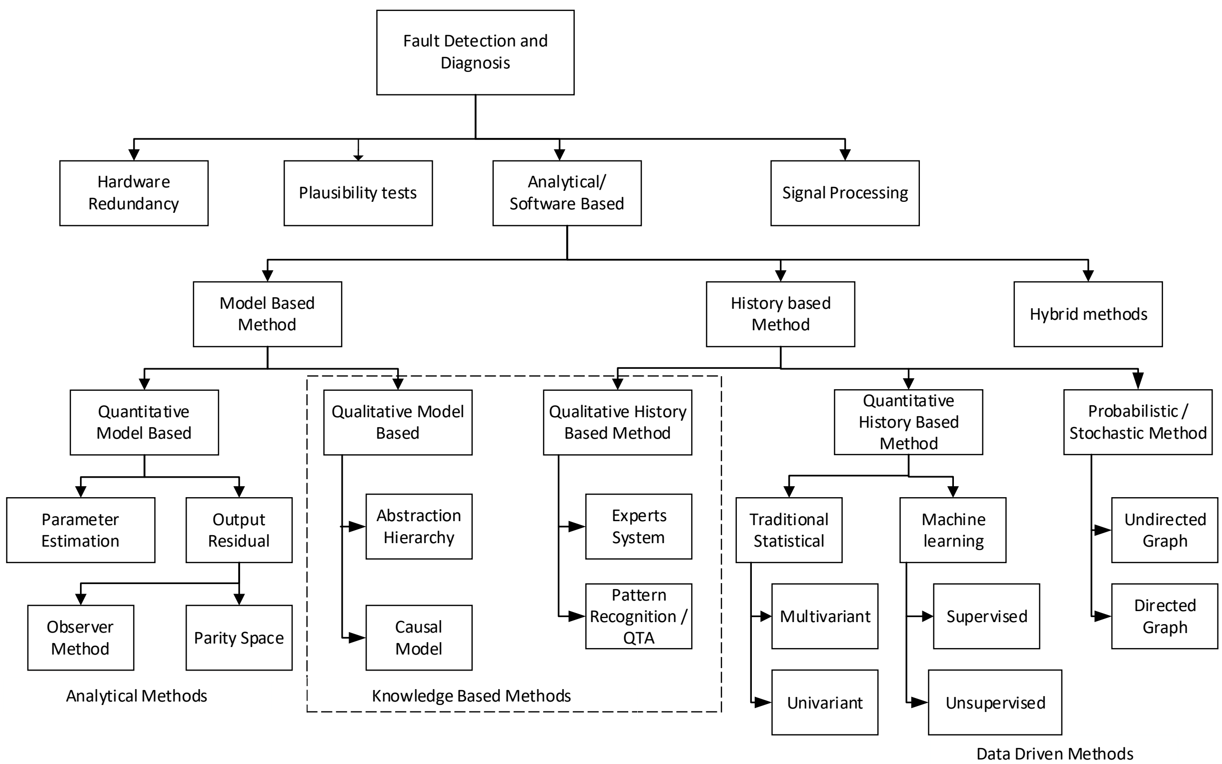


Figure 2.3 Fault detection and diagnosis methodology classification (Adapted from [24])

Since the early 1970s, fault detection and diagnosis in the process industry has been classified into four primary methodologies: hardware redundancy, plausibility tests, analytical models, and signal processing methods [20]. The hardware redundancy scheme initially developed the FDD method using identical hardware components redundant to the working system. The major drawback of this approach is that if an identical component system fails to generate the appropriate output, it may fail to detect the fault condition. The plausibility test highly relies on the investigation of physics laws in the system process component. When process systems become more complex, plausibility tests fail to detect fault conditions due to the physics laws' assumptions and real-time system accuracy. After Industry 3.0, to tackle the drawbacks mentioned above, hardware redundancy and plausibility tests have been replaced by computer-based analytical or data-based models. Apart from these models, signal processing methods use steady-state condition evaluation of the process signals to evaluate the fault condition. However, for complex process systems with many process parameters, this method is insufficient.

In this review, we mainly focus on reviewing the current state of the FDD methodologies. Computerized analytical redundancies and software-based redundancy models are primarily focused and further categorized as 1. analytical model based 2. knowledge-based, and 3. data-driven methods.

2.2.2.1 Analytical Model-Based Approaches

Analytical model-based approaches use first principles to develop mathematical models of the system. As shown in Figure 2.4, these model outputs are compared with an actual plant's measured data to obtain the fault knowledge. The process systems measured data are incorporated with system noise, disturbance, and other uncertainties. Also, the model output includes errors due to the model accuracy and design assumptions. Therefore, when processing residual using measured and model data, the appropriate threshold is applied to evaluate the analytical models' fault condition.

Parameter estimation, state observer, and parity relations are the classified approaches in the analytical-based FDD [20], [21], [25]. Gertler, Ding, and Isermann [26]–[28] further recommended for the analytical model-based approach primary contexts.

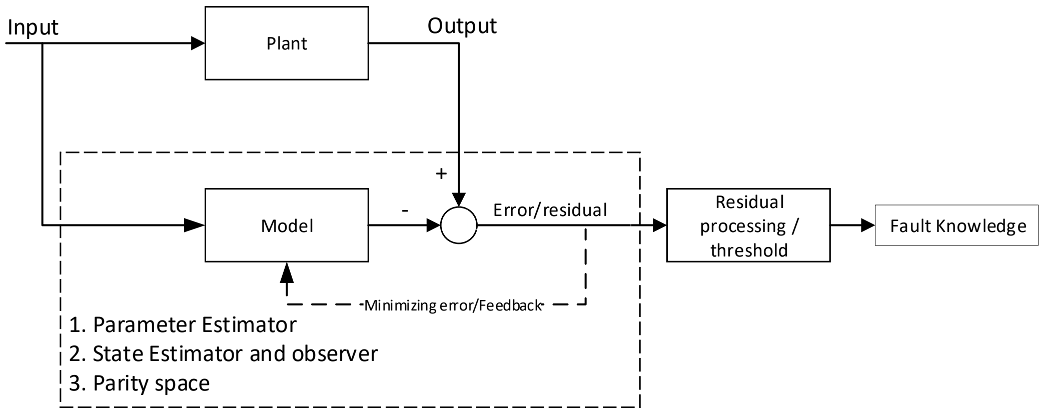


Figure 2.4 General Schematic description of the analytical model-based method (Adapted from Ding et al. (1999) [29])

However, from recent research, applying merely analytical model-based FDD will be ineffective due to the process system's complexity. Table 2.1 summarizes some of the recently developed analytical-based models.

Table 2-1 Analytical model-based approach in FDD

Model Classification	**Method**	**Approach**
As based on Parameter Estimation	Least squares (LS)	Least-squares [30] Parameter estimation with recursive least squares [31] Recursive Least Squares method with exponential weighting [32]
	Regression Analysis	Recursive ridge regression parameter estimation [33]
	Bounding Parameters	Bounded error parameter estimation [34] Parameter uncertainty [35]

State Estimation and Observer	Observer-Method	Observer-based approach [36] Observer approach with unknown input [37] Observer gain matrix [38]
	Kalman Filter	Bank of Kalman filter [39] Linear Kalman filter [40] Kalman Filter [41] Extended Kalman filter [42] Extended Kalman filter [43] Interacting Multiple Model Kalman filter [44] Two-stage Kalman filter [45] Decentralized Kalman filter [46] Fast converting Kalman filter [47] Particle Filter [48] Intelligent particle filter [49] Kalman filter with least square residual [50]
Parity space	State space-based method	Parity relation based residual generation [29] Parity space and CUSUM [51] Introduce stationary wavelet transform in the traditional parity relation based residual generator [52]
	Input output-based methods	Parity equations based on the input-output model [53]

2.2.2.2 Knowledge-Based Approaches

The knowledge-based methods are appropriate when a detailed mathematical model is not available, and the number of inputs, outputs, and states is relatively small [54]. However, with the development of computational applications and software packages, the knowledge-based method becomes more applicable to complex systems. There are not many recent survey articles published related to knowledge-based approaches based on FDD. However, Frank, Venkatasubramanian, et al., Gandhi et al., and Dai and Gao [55]–[57] have performed comprehensive surveys of the qualitative approach for FDD.

The knowledge-based methods are a qualitative approach of model-based and history-based methods [24], [55], [58]. Causal analysis, expert systems, pattern recognition, and qualitative trend analysis (QTA) are commonly applied knowledge-based methods of FDD in process safety. As illustrated in Figure 2.5, the knowledge-based FDD method is developed based on a large amount of data. Model-based qualitative methods are generated based on understanding the physical system and learning from the model's data, while historical-based qualitative methods learn from the measured data.

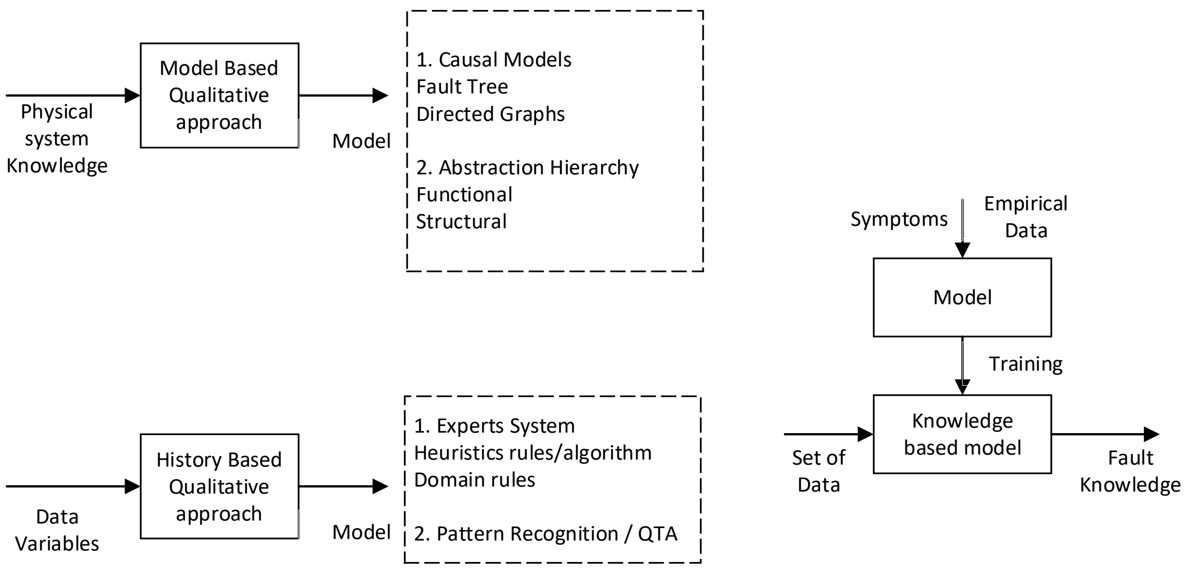


Figure 2.5 Knowledge-based model

Readers are recommended to read works by Chiang et al. and the Vesely et al. [20], [59] for the model-based qualitative approaches' primary principles. Table 2.2 summarizes most of the commonly used knowledge-based approaches.

Table 2-2 Knowledge-based approach in FDD

Knowledge-based model classification		**Methods/model approach**	**Approach**
Model-based Qualitative	Causal method	Fault tree Directed graph	Fault tree [60]–[62] Stochastic approach [63]–[67]
	Abstraction Hierarchy	Functional Structural	Functionally abstract information in FDD. [68], [69] Structural Approach. [70]
History-based Qualitative method	Expert system	Heuristic Algorithm/rules	Fuzzy classifier [71] Fuzzy logic system [72] Fuzzy-genetic algorithm [73] Fuzzy rules integrate with genetic algorithms [74]
	QTA		(QTA) - Principle Component Analysis (PCA) [75] QTA-based diagnostic system [76]

2.2.2.3 Data-Driven Approaches.

Data-driven approaches are the quantitative approach of history-based models. Data-driven methods can capture information and be translated to knowledge without much information about the physical system [24], [54], [77], [78]. Therefore, these models do not rely on the system's first principles or in-depth system information, which means that data-driven approaches are most suitable for modern complex and large-scale process systems.

As shown in Figure 2.6, data-driven models have been primarily developed offline using process history data with scopes for updating in real-time. However, data preprocessing and sampling/variable selection are essential steps to develop or update the model in both situations. The preprocessing step improves the model input data's quality by supplying missing data, removing outliers, and normalizing [79]. The sample/variable selection procedure determines the operating conditions of the system. A model can be developed based on the preprocessed input data by applying a statistical or machine learning approach.

The constructed models are verified and updated based on testing and real-time data. In the monitoring stage, the control limit or data pattern indicates abnormal events. Especially in machine learning approaches, a developed model can improve repeatedly based on real-time performance.

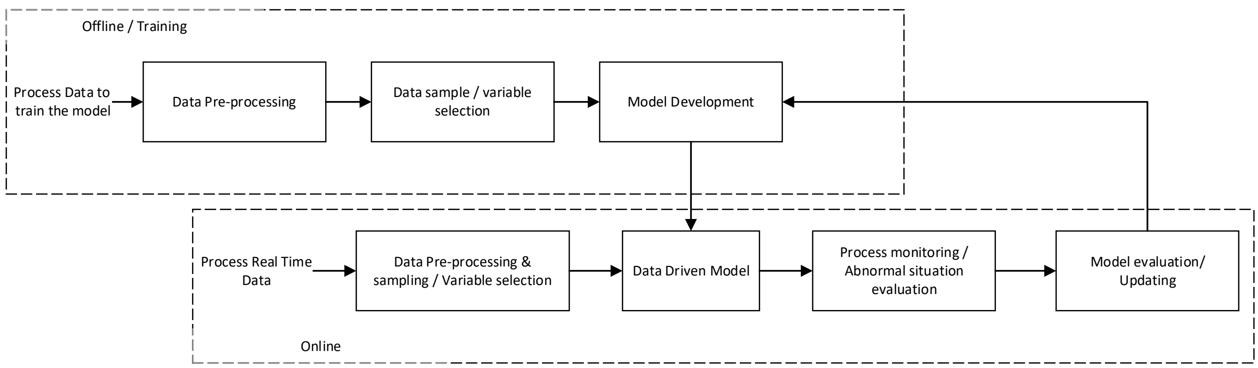


Figure 2.6 The generic framework of Data-Driven Approach

Yin et al. and Arunthavanathan et al. classify the data-driven approaches as traditional statistical and novel machine learning approaches [80], [81]. Traditional statistical models are designed to infer the relationships among the variables to estimate the model, using a sample population and hypothesis. In contrast, machine learning models are designed for more accurate predictions based on supervised or unsupervised learning.

A Traditional Statistical Approach

The data-based traditional statistical approaches are classified as univariate and multivariate [82]. Almost all real-time process systems contain multiple variables. Consequently, multivariate approaches are extensively used in recent data-driven approaches. However, in the univariate methods, each variable is monitored individually to obtain fault detection using each variable threshold limit. Figure 2.7 classifies the traditional statistical approach applied to fault detection and diagnosis.

Univariate Approach

According to Kano et al. and MacGregor and Kourti, statistical process controls (SPC) such as the Shewhart control chart, cumulative control charts (CUSUM), and exponentially weighted moving average control charts (EWMA) are broadly applied to investigate the univariate features in process systems [83]–[85]. These methods attempt to distinguish between normal and abnormal operations by setting upper and lower control limits using estimations. The univariate methods are commonly used for uncorrelated process data. Alternatively, when the process data variables are highly correlated, multivariate statistical analysis is applied to develop the model [86].

Multivariate Approaches

Due to the process complexity and highly correlated nature of process data, multivariate techniques have been extensively used in current decades. A principal component analysis (PCA) and partial least squares (PLS) methods have widely used Gaussian approaches, and the significant advantage of these methods is their ability to handle highly correlated data without preprocessing [87]. These two methods are used for feature dimension reduction and are combined with other statistical hypothesis testing algorithms such as T^2, ANOVA, and the MANOVA test to make them ideal methods for complex system FDD [85]. Readers are recommended to read works by Zhiqiang and Song and Ding for multivariate approaches' primary principles [88], [89]. Table 2.3 includes the most recent cited journal articles related to univariate and multivariate approaches.

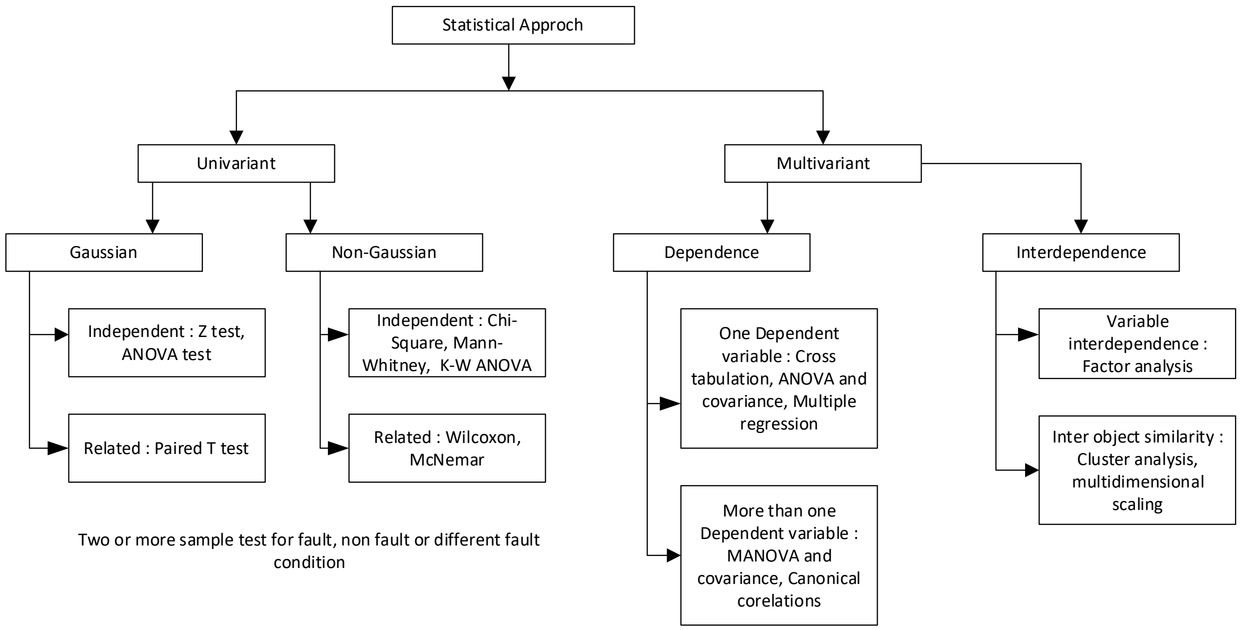


Figure 2.7 Statistical methods used to evaluate the fault condition.

B Machine Learning Approach

In this review, the authors distinguish statistical, knowledge-based, and machine learning approaches. Machine learning models learn from data based on patterns or inferences without depending on rules. In contrast, knowledge-based models learn from data following rules, and statistical models formulate the relationships to develop knowledge from input data. Liu et al. and Lei et al. have recently reviewed the machine learning approaches for fault detection and diagnosis [90], [91]. From their review, based on the learning process, machine learning techniques are classified as supervised and unsupervised learning. In the supervised learning process, a user provides information to train the computation model how to learn, and in unsupervised learning, the process model learns by itself.

Supervised Machine Learning

Supervised learning depends on a historical data set that contains a large amount of faulty and normal data [57], [92]. Therefore, it is required to collect all the fault condition data in the system-level approach to develop the model. However, it is challenging for a newly developed system to follow this approach due to the lack of faulty data. The study has also identified the Neural Network (NN) and Support Vector Machines (SVM) are commonly used supervised machine learning techniques to categorize the fault conditions.

The use of NN in process systems for fault detection began in the late ’80s. Hoskins and Himmelblau, Venkatasubramanian and Chan, and Lit et al. investigated and applied NN-based approaches to detect and diagnose the fault condition in process systems [93]–[95]. The NN-based FDD algorithms are capable of handling nonlinear dynamic system data. Therefore, sufficient research interest was shown in the early 90s; however, due to the requirements for computation capability and data collection for normal and abnormal conditions, developing models based on NN has become complex.

The SVM-based approach for FDD for nonlinear systems has been developed from statistical learning and pattern recognition, proposed by Vapnik [96], [97]. He introduces the kernel trick to create nonlinear binary classifiers and succeeds with the application. The SVM aims for classification by avoiding the general machine learning problems such as model selection, overfitting, nonlinearity, dimensionality, and a local minimum [98]–[100]. After the invention of the kernel trick, SVM became one of the major research topics in FDD. However, SVM was initially developed for binary classification and has been effectively extended for multiclass classification with recent research [101].

SVM and NN approaches are capable of classifying normal conditions and fault deviation. However, to isolate and analyze the fault condition, these two approaches have required modification in the algorithm or are interfaced with other approaches, such as probabilistic or physical system models. Some of the recent approaches based on NN and SVM are delineated in Table 2.3. Moreover, supervised learning approaches are better suited for classification but cannot be generalized.

Unsupervised Machine Learning

One of the major drawbacks of supervised learning is that labeled data are required to train the model. However, in the process industry, generating data with such information will be a challenge. Consequently, the unsupervised learning approach has become popular in machine learning FDD in recent years.

Unsupervised machine learning models are trained using unlabelled historical data. However, by finding a hidden pattern from the data, these models select classification tags using the algorithm. The clustering and anomaly detection are widely used unsupervised model behaviors in FDD [102], [103]. Clustering is a process of aligning the set of data as a smaller group and is used for multiclass classification. Anomaly detection is a binary classification to determine the outlier. Recent unsupervised approaches for FDD are included in Table 2.3.

C Probabilistic Methods

The Hidden Markov Model (HMM) and the Bayesian Network (BN) based approaches commonly used FDD methods built on the stochastic probabilistic theories. HMM is a graphical model with random variables that use the Markov property described by an undirected graph. The Bayesian network is a directed acyclic probabilistic graphical model applied in various applications, including FDD. Table 2.3 illustrates recent FDD approaches based on BN and HMM.

Table 2-3 Data-Driven Approach in FDD

Data-Driven Model	**Method**	**Approach**
Traditional Approach	Univariate / Multivariate	CUSUM and PCA based approach [104] PCA based T^2 and Q statistics method [105] Statistical Process Control (SPC) based on PLS and PCA [85] Independent component analysis and SPC [83] PCA based Hoteling's T2 -statistic, Q-statistic and Q-residual contribution [106]
Machine Learning	Supervised	Neural Network [107], [108] Support vector machine [109] One-class support vector machines [110] Support vector machine [111], [112]
	Unsupervised	One-class support vector machine [113] PCA T^2 statistic; hierarchical clustering, K-Means, Fuzzy C-Means clustering, and model-based clustering [114] Support vector clustering (SVC)-based probabilistic approach [115] Generative adversarial network (GAN) [116] Incremental one class NN [117]
Probabilistic – stochastic	Bayesian Network	Bayesian network inference [118] Hybrid dynamic Bayesian networks (DBN) [119] Multi-time slice dynamic Bayesian network with a mixture of the Gaussian output [120] Bayesian network, and Conditional Gaussian Network (CGN) [121] Dynamic Bayesian network with fuzzy sets theory [122]
	HMM	Hidden Markov chain model [123] Hidden Markov model-based independent component analysis [124] Hidden Markov Model (HMM) for abnormalities detection, and Bayesian Network (BN) diagnoses the ,28s of faults [125] Machine learning (ML)-based Hidden Markov model (HMM) and the principal component analysis (PCA) model [126]

2.2.2.4 Current Research Trends in Data-Driven Approaches and Challenges for Industry 4.0

Process data are becoming more sophisticated and highly correlated due to process systems' growth, based on technological progress and Industry 4.0. Therefore, shallow machine learning approaches are becoming insufficient to learn large deep process data. In recent years, deep learning (DL) approaches have been developed to learn the mapping from collected featured data to detect faults. After the year 2006, DL became most popular in FDD, due to the continuous increase of computational ability in the past decades and the development of NN optimization algorithms [127].

Wu and Zhao, Jia et al. and, Khan and Yairi have contributed surveys of recent deep learning approaches in FDD [128]–[130]. They reported that the Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Autoencoder (AE), and Restricted Boltzmann Machine (RBM) enable widely applied recent DL methods for FDD. Autoencoder approaches are further classified as stacked autoencoders (SAE) and variational autoencoders (VAE). As for RBM, methods are further classified as the Deep Belief Network (DBN) and Deep Boltzmann Machine (DBM). Shrestha and Mahmood, and Saufi et al. review and describe each deep learning approach's detailed architecture and algorithm in their review [131], [132]. Table 2.4 presents a comparison and use of the deep learning model in FDD.

Table 2-4 The comparison of DL methods

Deep Learning Model		**Training Type**	**Training Algorithm**	**Use of ASM**
CNN		Supervised	Gradient descent-based Backpropagation	Automated feature extraction, FDD, and Estimation of RUL
RNN		Supervised	Gradient descent and Backpropagation through time	RUL estimation, Failure prognosis, and FDD
AE	SAE	Unsupervised	Backpropagation	Anomaly detection, Failure Prognosis RUL estimation, and FDD
	VAE			Anomaly detection, RUL Estimation
RBM	DBM	Unsupervised	Backpropagation	Fault diagnosis.
	DBN			Feature extraction, Failure Prognosis, and FDD

Beyond fault detection and diagnosis, DL models are used to evaluate the remaining useful life (RUL), failure prognosis, and system health monitoring. Therefore, current researchers mainly focus on DL approaches to develop future process safety tools for the smart plant. Table 2.5 summarizes the recent research contribution based on DL approaches. However, due to forthcoming approaches, a few contributions from other areas, and some of the conference articles with useful technical contributions, are also included in Table 2.5.

Table 2-5 The recent DL approaches used in ASM

DL Model		**Approach and suggested ASM applications**
CNN		Automated feature extraction [133], [134] Fault identification, detection, diagnosis and, classification [135]–[137] Estimate RUL [138]–[142] Prognostic and health management [143]
RNN		Estimate RUL [144]–[147] Failure prognosis [148]–[153] FDD in chemical process [154], [155]
Autoencoder	SAE	Anomaly detection [156] RUL Estimation [157] Prognosis and health monitoring [158]
	VAE	Anomaly detection [159], [160] RUL estimation [161], [162]
Restricted Boltzmann Machine	DBN	Anomaly detection [163] RUL and health monitoring [164] FDD in process systems [165], [166]
	DBM	FDD in process systems [167]
Hybrid	CNN-LSTM	Estimate RUL [168], [169] Fault prognosis [170]
	AE - LSTM	FDD [171]

2.2.3 The Comparison of FDD Approaches

Based on the article reviews in sections 2.2.2.1, 2.2.2.2, and 2.2.2.3, FDD Analytical/Software-based methods are compared based on the performance and summarize in Table 2.6. This comparison relates to the fundamental classification of FDD.

Table 2-6: Qualitative comparison for FDD methods discussed in section 2.2

	Model-Based Approaches	**Knowledge-Based Approaches**	**Data-Driven Approaches**	
			Statistical	**Machine Learning**
Diagnosis Ability	Good	Excellent	Satisfactory	Good
Detection Speed	Quick	Quick	It depends on data size	Depending on data size and computational speed
Isolability	Satisfactory	Good	Excellent	Good
Identifiability	Good	Satisfactory	Excellent	Good
Model Development Complexity	Hard	Medium	Easy	Easy
Handling nonlinearity	Poor	Satisfactory	Good	Excellent
Generalization Capability	Good	Satisfactory	Poor	Poor (Unworkable)
Robustness	Poor	Satisfactory	Good	Excellent

2.3 ASM Approaches to Protect the Hazard Using FDD

Process systems can be protected from hazards by providing appropriate prevention and control by ASM. This section discusses the approaches and safety standards used in the process industry in ASM.

2.3.1 The Recent Trend in ASM Approaches

ASM approaches aim to provide early detection and timely corrective action in the process industry. The traditional ASM approaches such as Alarm Management (AM) and the Safety Instrumented System (SIS) alert the operators and take necessary actions for the system to proceed in a safe state [172], [173]. Whenever a system deviates from its normal operations, guidance on safety integrity levels, based on IEC 61508 and 61511 standards, determines SIS's required performance to reduce the system risk by implementing a Safety Instrumented Function (SIF). In many instances, this approach is ineffective due to the detection time and unplanned equipment shutdown. Further, most of the FDD approaches used in ASM fail to diagnose the source of faults. Therefore, in the past decade, ASM prevention and control highly rely on emergency shutdown and human interactions to repair the system. However, most plants attempt to avoid an emergency shutdown if they can do so without any risk [172], [174]. If a shutdown occurs, plant operation will be significantly restricted, and this may lead to mass loss of production, loss of profit in the economy, equipment damage, and a considerable degree of an internal investigation.

2.3.1.1 Alarm Management

Process alarm design, installation, and management are mainly based on industrial standards, e.g., EEMUA 191 British standards, and ANSI/ISA 18.2 American standards.

There are not many reviews of alarm management in the process systems published in past years. Alford et al., Wang et al., and Goel et al. review the alarm management guidelines, regulations, standards, and challenges in process systems [175]–[177]. From their review, alarm flooding, nuisance alarms, and operator workload are major alarm management issues. Based on the investigations of major process industry accidents such as Mile Island (1979), Bhopal (1984), and disasters in three Milford Haven accidents, a common alarm management issue was identified as alarm flooding.

Many researchers have aimed to provide appropriate methods to eliminate alarm flooding in recent years. Some of the contributions are summarized in Table 2.7.

Table 2-7 Alarm management issues and recent contributions

Alarm Management Issue	Contribution
Alarm Flooding	Smith-Waterman algorithm for pattern matching of alarm flood sequences used to eliminate the alarm flooding [178] Determine the causes of alarms of an alarm flood [179] Detection of frequent alarm patterns using the Itemset mining method [180] PrefixSpan sequential pattern recognition algorithm is used for alarm pattern detection [181] The exponentially attenuated component analysis is used for an early alarm flood classification [182] Online alarm flood classification [183] Semi-supervised machine learning and cased based reasoning used for alarm flood issues [184]
Operator workload	Quantify the effectiveness of alarm management based on the human factor [185] Assist operator during critical events using the probability of the event and risk priority Data-driven method is used to construct an operator workflow model in response to the alarm [186]
Effective alarm management	Intelligent alarm management framework [187] Alarm and event management based on alarm historical log data set [188] False alarm management [189] Sequential pattern mining is used for alarm management [190]

A Recent Trend in Alarm Design and Installation

In general, when a fault is detected, a sound or flashing alarm is activated to alert the operator. However, this alarm does not provide more information regarding faults or failures. Rather than using a loud or flashing alarm, a graphical visualized interface would be a better solution for conveying the fault or failure information to operators. In current decades, industrial alarms have been implemented using a Distributed Control System (DCS) or Programmable Logic Controller (PLC) interconnected with the Human-Machine Interface (HMI), helping to build the alarm system with visualized and graphical methods [187]. All DCSs are installed with an inbuilt alarm display [191]. Also, the DCS is the most scalable device, with a large number of input-output ports with appropriate data transmission facilities. Therefore, in recent decades, the DCS has been commonly used in alarm management.

2.3.1.2 Safety Instrumented System (SIS)

The main drawback of alarm systems is the necessity of manual intervention of the operator. SIS was initially invented by the International Electromechanical Commission (IEC) in electrical, electronic, and programmable electronic safety-related systems for IEC 61508 publication standards (1998). American standards ANSI/ISA 84.00.01 (2004), adopted with IEC 61511 standards (2003), support reliable SIS design in the process industry. Recently, the IEC 61511 standards (2016) have been updated with a focus on security levels and functional safety.

Based on the standards, the general SIS system comprises sensors, logic controllers, and actuator devices [192]. Sensors sense the data from the operation (such as temperature, pressure, or flow data). Logic controllers are the heart of the system and process the operating data (PLC, DCS, or any microprocessor-based system), and an actuator controls the valves.

SIS's have been implemented to prevent the process systems or plants from becoming a hazardous environment. Therefore, the failure rate and reliability of the SIS play major roles in industrial implementation. Reliability calculation goals are frequently allocated to each safety instrumented function (SIF) performed by an SIS, and detailed reliability analysis is carried out to prove compliance to these calculated goals. Lundteigen and Rausand (2010) define the function that is performed by an SIS as SIF [193]. SIF targets for reliability in the process industry are set by IEC 61508 and IEC 61511 standards. These standards are used to measure reliability and distinguish among four safety integrity levels (SIL). The reliability calculation goal determines the average probability of failure on demand (PFD). Current approaches for SIS reliability analysis are summarized in Table 2.8.

Table 2-8 SIS reliability analysis methods

Reliability quantification methods	**References**
Reliability block diagram (RBD)	PFD calculation using a reliability block diagram [194]
Fault tree analysis (FTA)	FTA using priority AND gates [195] Safety loop for PFD calculations [196] FTA uses to evaluate the PFD of SIS [197]
Markov Models (MM)	Markov model with constant failure rates [198] Multiphase Markovian approach [199] Based on different demand modes [200]
Petri-Net (PN) approach	PFD calculation is based on FTA, MM, and PN [201] PN modeling [202]
Monte Carlo (MC) Simulation	MC simulation [203] MC and RBD [204]

2.3.2 ASM Challenges in Industry 4.0

A smart process system for Industry 4.0 requires automated SIS and repair to bring the abnormal condition to normal operation without human intervention. Therefore, fault detection time and repairing process time will play major roles in next-generation ASM research.

Moreover, a fault to failure transmission is a critical issue in ASM. Past decades of research trends demonstrate that process systems' early warning and timely diagnosis help prevent loss by appropriately modeling the failure condition. Adhitya et al. performed an experimental study to quantify an early warning system's benefit [185]. They found that early warning was effective in reducing diagnosis delay, and this was subjectively perceived to be beneficial by the experiment's contributors. However, it did not improve the diagnosis accuracy.

2.4 Review of Risk Assessment Approaches

A process accident generally follows the sequence of initiation (starting event for the accident), propagation (maintaining or expanding an event to prolong an accident scenario), and termination (an event that stops the accident). Hazardous events continuously change in each stage of accident scenarios. Therefore, appropriately investigating the accident scenario at each stage is important in process safety management to identify the hazardous environment. System failures cause process systems accidents. Therefore, investigating system failure will be the initial starting point to determine the hazardous event and examine the accident's initiation.

By properly evaluating the risk assessment and applying appropriate ASM to the process, the system will help control a system hazard before it leads to an accident. As shown in Figure 2.1, when the fault to failure transition leads to an accident, the failure model determines the system failure probability, consequences of failure are quantified by the accident model, and risk models assess the risk based on the obtained failure probability and consequences.

Many researchers have reviewed methods and tools for process safety, including hazard identification, risk assessment, and safety management based on accident models and failure models. Khan and Abbasi, have investigated the available risk assessment techniques and methods [9], [205]. Khan et al. also have developed a risk-based approach to measure process safety using a set of safety performance indicators [206]. Swuste et al. have investigated leading and lagging safety indicators for process safety [10]. Amin et al. recently performed a bibliometric analysis of process safety under the key areas, performance tools, and leading research contributions [207]. This section reviews the past and present failure models, accident models, and risk models that aid decision making during abnormal situation management.

2.4.1 Review Framework and the Selection of the Articles

Most of the risk assessment approaches are available in public journals, conferences' symposiums, and magazine articles. However, this review article summarizes the technical articles published in scientific journals. Since this review's scope is limited to process systems, seven key journals with similar aims and scope are selected; the number of publications in the journals is summarized in Figure 2.8.

The literature survey is performed based on the keywords: Quantitative risk assessment, Risk assessment, Failure models, Accident models, Consequence Model, Risk model, Risk Analysis, and Failure analysis. The technical articles' direct relations to the scopes are filtered for the study. A simple statistical analysis is done based on Scopus, the web of science, and IEEE Xplore.

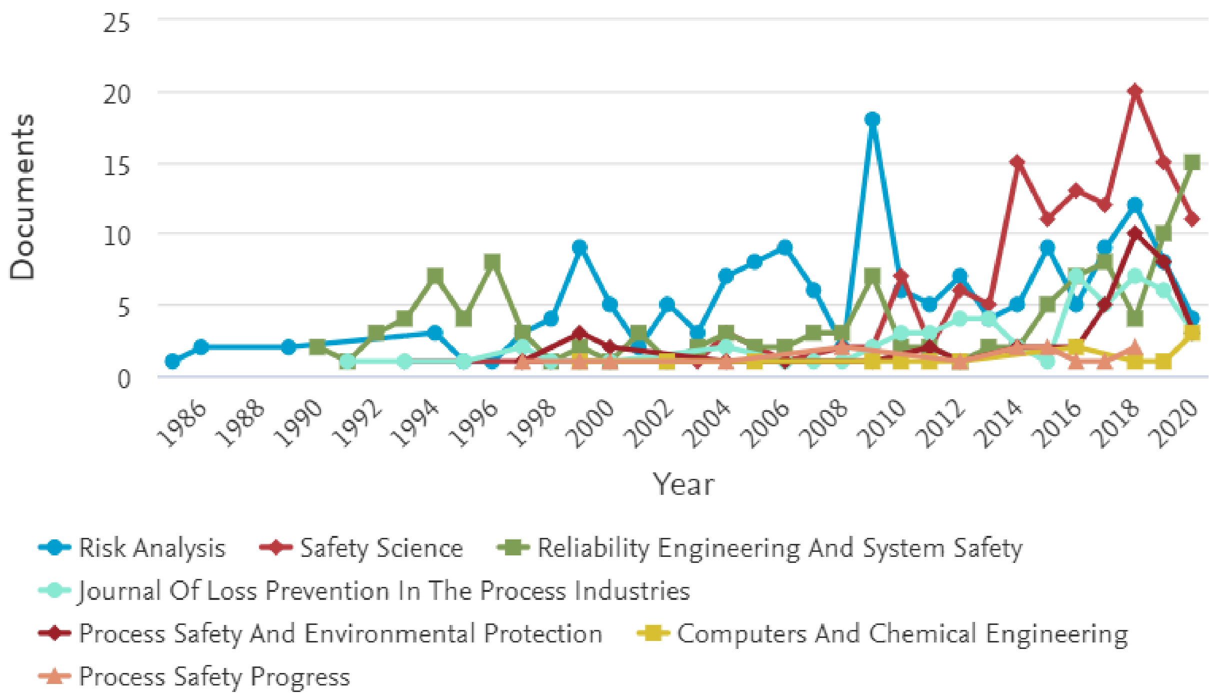


Figure 2.8 Risk assessment review article published in the relevant journals [As of 27th July 2020]

The largest number of articles related to the subject areas are published in RESS, JLP, and CCE. The foci of RESS JLP and CCE are discussed in section 2.2.1. The Risk Analysis journal provides a crucial point for new developments in the field of risk analysis. Safety Science has mainly focused on articles based on accidents and disasters of special significance. Process Safety Progress mainly addresses hazardous chemical management and leak prevention, risk assessment, process hazard evaluation, and preventive maintenance related to process safety. Papers related to process system safety, including modeling, accident investigation, risk assessment, and safety-related topics, are the main concern areas for Process Safety and Environmental Protection.

2.4.2 System Failure, Risk Assessment, and Risk Mitigation

The process system design or plant implementation are the main factors in hazard identification and RA [208]. If a system or a plant applies proper abnormal situation management, alarm management, and SIS, the probability of a hazardous event might be reduced. However, when a fault occurs and leads to failure and an accident, investigating the accident scenario and finding the accident's cause will be the RA's starting process.

Using the RA in the safety perspective of system failure analysis, it is important to investigate the systems' fault, failure condition, and the accident that occurred due to the failure. Therefore, investigating and modeling the fault to failure probability by failure modeling, and studying the accident scenario with initiating events and consequences with the accident model, will be the first tasks for the RA perspective of system failure.

The relationships among RA, FDD, and ASM are discussed in Figure 2.9. Using the appropriate fault analysis, process systems can implement appropriate safety barriers to reduce accident probability. Therefore, developing accurate accident models based on process system failure may suggest the required fault detection and diagnosis models and alarm or SIS management applicable to the system.

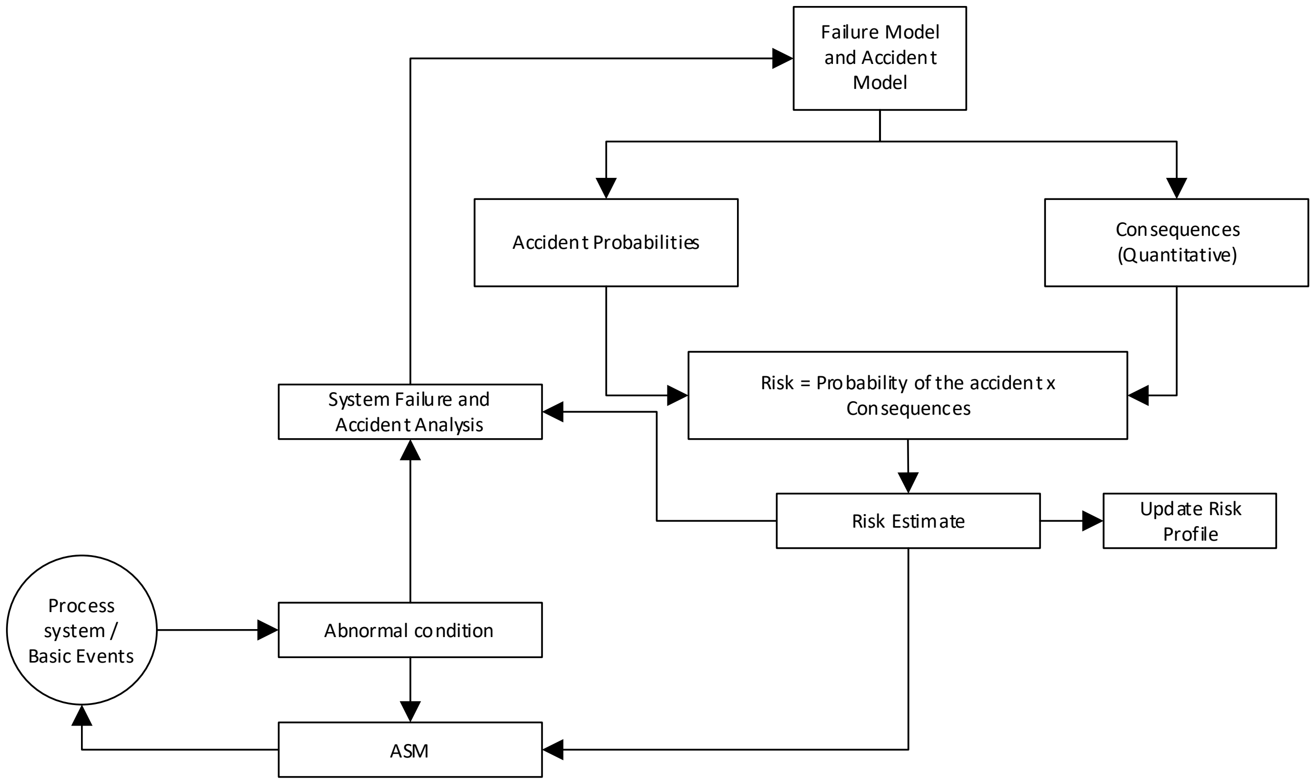


Figure 2.9 Process system failure to accident relationship based on risk assessment.

2.4.3 Review of Accident and Failure Models

Accident modeling is a methodology used to relate the cause and consequences of the incident that leads to the accident from system failure. It is necessary to analyze hazard identification and risk assessment based on the available process information to evaluate the system's risk from a process failure perspective with an accident model. A relative ranking, hazard checklists, hazard surveys, safety reviews, what-if analysis, failure mode and effect analysis (FMEA), and hazard and operability (HAZOP) studies are commonly used hazard identification methods in the process industries [209]. A HAZOP is the most widely used hazard identification approach in the process industry, and its study approach has been modified over the years based on evolving technologies [210].

2.4.3.1 Accident Models Classifications

An accident models' development starts with the domino theory introduced by Heinrich in 1941 [211]. Different reviewers have classified accident models. Qureshi classified the models into traditional and modern accident models [211]. Al-Shanini et al. have further classified traditional models into sequential and epidemiological models, and modern accident models are classified into systematic and formal models [212]. In their recent review, Fu et al. classified accident models as linear and nonlinear, based on the accident's logical sequence [213]. On the basis of accident models currently used in the process industry and from the work of Hollnagel, accident models in process systems are also classified as sequential models, epidemiological models, and systemic models in this review [214].

Sequential models represent the accident as the outcome of a series of individual steps, based on the order of occurrence of the accident. Epidemiological accident models describe an accident as analogous to a disease resulting from a combination of latent and active system failures.

Systemic accident models are based on system theory; rather than treating accidents as a sequence of cause-effect events, accident models describe losses as the system's unexpected behavior based on the system component's uncontrolled operation. A systemic accident model must be developed due to process systems' complexity [215]. However, these models are mainly used by academic researchers to analyze accidents.

Dynamic sequential accident models classified by Al-Shanini et al., and dynamic risk analysis based on sequential models, represent the accident scenario and combine with other systemic approaches to accommodate nonlinear and complex interactions as dynamically updating features in a single model [212]. Accident prevention models, dynamic risk assessment, and commonly used process accident models are classified in Figure 2.10.

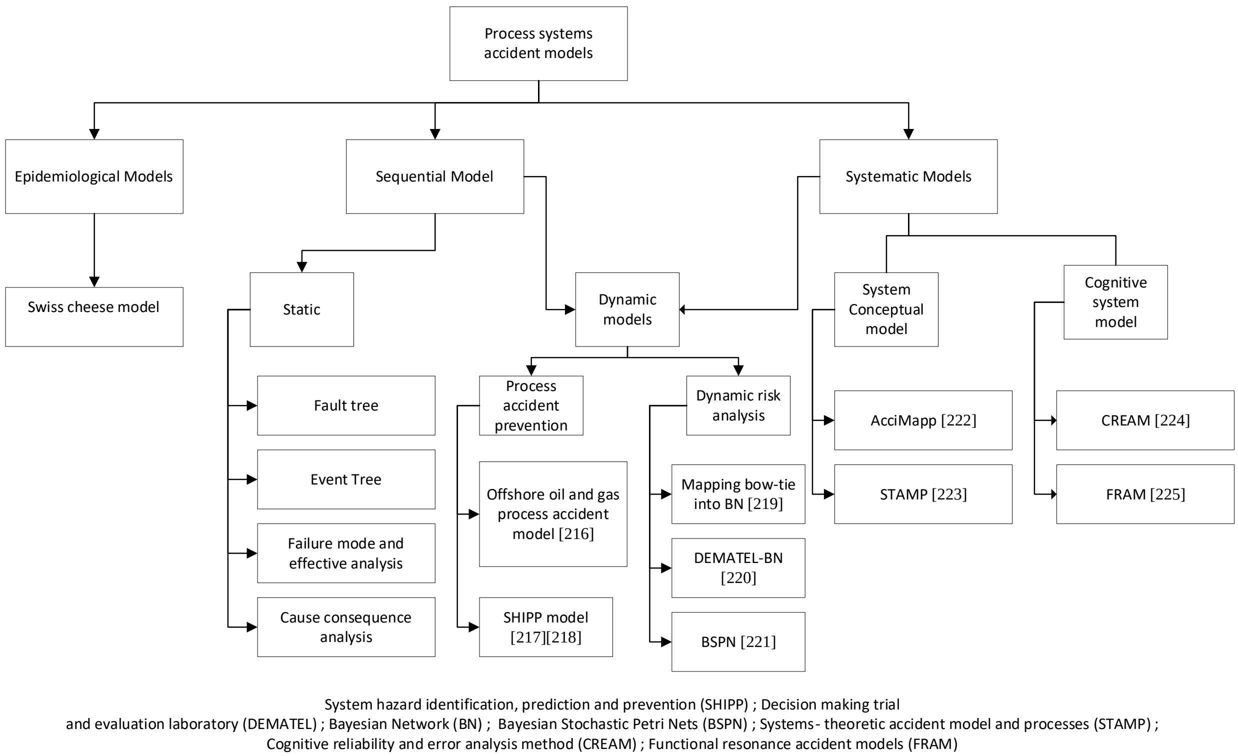


Figure 2.10 Process system accident and failure models and reference citation (adopted from [212])

2.4.3.2 The Recent Trend in Accident Models and Failure Models

Dynamic risk assessment, dynamic safety analysis, and dynamic accident models are forthcoming keywords and research areas for process accident models. The dynamic Bayesian network and Petri-nets are commonly used approaches to study the dynamic behavior of the system in recent years [216]–[218]. Combining these approaches with conventional accident models will result in the development of dynamic accident models in the future.

2.4.4 Review of Risk Modeling

After the Industry 3.0 era, risk assessment has emerged as an essential and systematic tool that plays a critical role in overall safety management. Many reviewers have reviewed and classified process systems' safety-related risk assessments in the past. Khan and Abbasi presented a comprehensive analysis of the quantitative and qualitative risk assessment models available up to 1998 [9]. Tixier et al. listed and reviewed 62 risk analysis approaches in both qualitative and quantitative terms for general plant industries [219]. Marhavilas et al. reviewed and presented the qualitative, quantitative, and hybrid risk assessment approaches from 2000 to 2009 [220]. Researchers have focused on quantitative and hybrid approaches in recent decades due to their safety risk mitigation and decision-making abilities. Necci et al. reviewed quantitative risk assessment for process industries, specifically regarding the domino accident theory [221]. Villa et al. reviewed risk assessment methods to enable dynamic risk assessment for next-generation implementation [222]. Recently, Kabir and Papadopoulos reviewed and comprehensively described Bayesian network approaches and Petri net approaches used in the risk assessment process [223].

RA methods are classified as quantitative, qualitative, and comprehensive methods [224], [225]. The qualitative techniques are based on analytical estimation and human ability [222], [225]. The quantitative approaches quantify the risks and further estimate and express them using mathematical relations based on real-time accident data. The comprehensive methods effectively incorporate the benefits of qualitative and quantitative methods. Therefore, these approaches are currently widely used in risk assessment techniques. Figure 2.11 summarizes the present risk assessment model classifications.

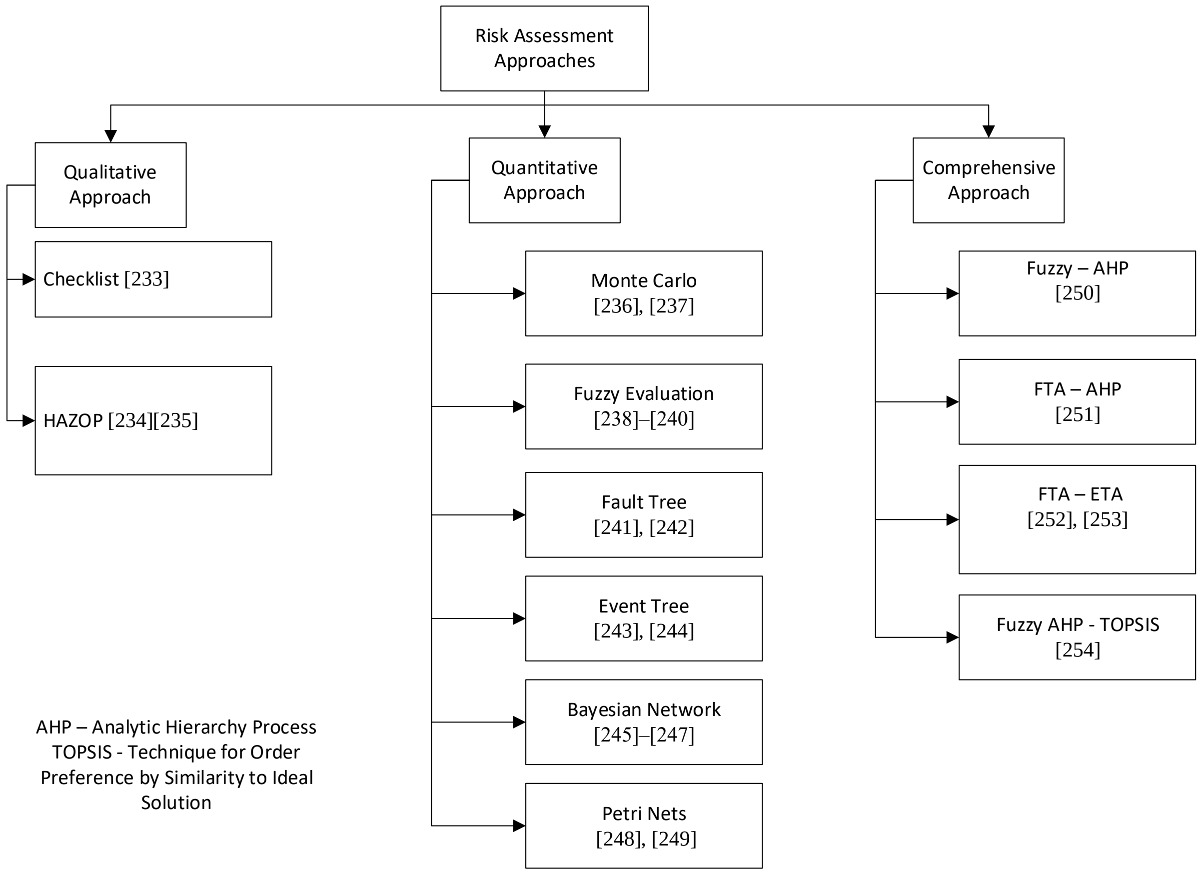


Figure 2.11 Risk Assessment Approaches

2.4.5 Machine Learning Approaches for RA and Industry 4.0 Challenges

Machine learning algorithms have aided risk assessment in recent years. However, there are very few articles found which relate them to process systems. Due to Industry 4.0 and an increase in intelligence ability in the dynamically changing risk, process system researchers intend to apply machine learning algorithms for dynamic risk assessment in the near future.

Paltrinieri et al. and Hegde and Rokseth recently reviewed machine learning approaches for risk assessment [226], [227]. They report that the automotive and construction industries are leading the adoption of ML for risk assessment. Furthermore, they have found that ANN, SVM, Decision Tree, K-means, and Naïve Bayes are the most commonly used machine learning approaches in

RA. The most recent ML approaches from other process-related applications for RA are summarized in Table 2.9.

Table 2-9 RA ML model recent approach summary

Method/Model	Approach
ANN	Natural Language Processing [228] Mamdani Fuzzy Neural Network [229] ANN to train both simulator and plant parameters. [230] Multilayer perception ANN [231]
SVM	Hazard identification and prediction using SVM [232]
Decision Tree	Decision Tree is used as a black box [233]
K-means	FMEA model using double hierarchy hesitant fuzzy linguistic term sets and k-means clustering is developed to evaluate and cluster the risk [234] K-means based risk assessment using pipeline data [235]
Naïve Bayes	Naive Bayes classifier [236]
Random Forest	Random forest classifier [237]

2.5 Next-Generation Process Safety and Risk Management Based on Process System Failure

With extended technology, process system plants become more complex and advanced. Hence, process safety will be a challenging topic in the upcoming years. Industry 4.0 and smart industrial development process systems confer several effects on the design and development of the largest plant industry [238]–[240]. This will profoundly affect FDD methods and models, risk assessment approaches, and ASM strategies.

Based on Industry 4.0, next-generation plant development might improve with smart technologies such as smart sensors, IoT, and advanced communication. As a result of smart plants and physical system complexity, data-based approaches with a large amount of data can be more operable to implement FDD and RA models and methods. Therefore, handling large process system data, big data analysis, and cloud computing may involve model implementation.

With the changes in the next generation of process plant based on the Industry 4.0 era, some of the impacts on process safety elements models and methods are as follows:

1. Due to a large amount of sensor data to detect the abnormal behavior of the system, deep learning approaches may be more employable models in future FDD. When implementing learning from a large amount of data, the processing time must be considered from a safety perspective. Therefore, data processing, computation, and speed of data communication will significantly affect the models.

2. Process plant prognosis and health management based on failure prognosis and RUL are a current trend in fault detection and diagnosis. Failure prognosis can predict the failure, and RUL can predict the gap in time between fault and failure. A hybrid approach based on data-driven models, such as supervised and unsupervised machine learning, and system model descriptions such as graphical, stochastic approaches, will be forthcoming models for developing failure prognosis and RUL.

3. To implement the SISs with automated fault correction before system failure, success of future ASM will depend on skillful utilization of the following Industry 4.0 technologies:

    Industrial Internet of Things (IIoT), Cybersecurity with a wireless communication layer, Real-time constraints including data digitization and extensive data processing, big data and cloud computing, system modularization to replace or expand individual modules, intelligent advanced controllers, digital twins (a combination of IoT and ANN).

4. The ability to implement repair, replacement, or maintenance for basic controllers based on the fault prognosis and RUL will be another contributor in process plant economics.

5. With Industry 4.0 era and smart plants, system safety highly depends on the sensors and other physical components, such as controller devices, communication devices, protocols, and actuators. Therefore, the reliability of the components will be a primary concern in process safety.

6. With smart technologies and combining plant units with digital communication may increase failure probability and risk. Therefore, implementing risk assessment models with different scenarios will challenge the next generation. Developing accident models based on the dynamic risk assessment will be another addition to process safety.

Overall, the next generation in process safety highly depends on electronics, communication, advanced controller devices, and computational algorithms such as machine learning and state-space stochastic models. However, dealing with the smart plant, the Industry 4.0 era, and with frequently modifying safety standards in process safety will be an incredible challenge for forthcoming safety generation.

2.6 Conclusions

This review's main objective is to illustrate the safety framework for the process industry by integrating fault detection and diagnosis, abnormal situation management, and risk assessment. The review's main scope was restricted to published journal articles on topics directly related to these three areas. Limited conference papers and industrial standards were also reviewed to discuss the safety levels and standard industrial approaches related to process safety. Overall, this review article mainly focused on researchers interested in process systems and process safety.

It is noted that FDD researchers tend to focus more on data-driven quantitative approaches than model-based system descriptions and knowledge-based methods. Moreover, rather than traditional fault detection and diagnosis, implementing failure prognosis and remaining useful life approaches are becoming a major topic in current and future research, to facilitate predict the failure scenario as early as possible. Less complex smaller applications may involve analytical modeling approaches. However, large scale complex applications will continue to use data-driven approaches to evaluate the fault condition.

This section also summarizes the abnormal situation management industry standards and techniques, including alarm management and safety instrumented system approaches, to maintain safe operation in process systems.

Failure models, accident models, and risk models are further reviewed in this article. The transition from static risk assessment considering a single accident scenario to a dynamic quantitative accident scenario has been a recent dominant trend and a prospective upcoming research area.

Finally, with improving technology and the upcoming smart industry integrated safety framework, research interest is discussed. To keep up with Industry 4.0, fault detection and diagnosis must focus on early failure prognosis, and the process needs to operate in a more autonomous fashion, including automation of the abnormal situation management tasks. The risk assessment should also focus on the evolving nature of risk to consider the potential design and operational failure scenarios. However, appropriate data collection and failure probability prediction, dynamic risk forecasting, autonomous safety instrumented system development, and obtaining risk margins for abnormal situation management will be a technical challenge for future research.

Reference

[1] J. Dunjó, V. Fthenakis, J. A. Vílchez, and J. Arnaldos, "Hazard and operability (HAZOP) analysis. A literature review," *Journal of Hazardous Materials*, vol. 173, no. 1–3, pp. 19–32, Jan. 2010, doi: 10.1016/J.JHAZMAT.2009.08.076.

[2] I. Cameron *et al.*, "Process hazard analysis, hazard identification and scenario definition: Are the conventional tools sufficient, or should and can we do much better?," *Process Safety and Environmental Protection*, vol. 110, pp. 53–70, Aug. 2017, doi: 10.1016/J.PSEP.2017.01.025.

[3] R. J. Willey, "Layer of protection analysis," in *Procedia Engineering*, 2014, vol. 84, pp. 12–22, doi: 10.1016/j.proeng.2014.10.405.

[4] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques-part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015, doi: 10.1109/TIE.2015.2417501.

[5] M. Zhong, T. Xue, and S. X. Ding, "A survey on model-based fault diagnosis for linear discrete time-varying systems," *Neurocomputing*, vol. 306, pp. 51–60, Sep. 2018, doi: 10.1016/j.neucom.2018.04.037.

[6] I. Punčochář and J. Škach, "A Survey of Active Fault Diagnosis Methods," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 1091–1098, Jan. 2018, doi: 10.1016/J.IFACOL.2018.09.726.

[7] N. Md Nor, C. R. Che Hassan, and M. A. Hussain, "A review of data-driven fault detection and diagnosis methods: Applications in chemical process systems," *Reviews in Chemical Engineering*, vol. 36, no. 4, pp. 513–553, May 2020, doi: 10.1515/REVCE-2017-0069/MACHINEREADABLECITATION/RIS.

[8] D.-T. Hoang and H.-J. Kang, "A survey on Deep Learning based bearing fault diagnosis," *Neurocomputing*, vol. 335, pp. 327–335, Mar. 2019, doi: 10.1016/J.NEUCOM.2018.06.078.

[9] F. I. Khan and S. . Abbasi, "Techniques and methodologies for risk analysis in chemical process industries," *Journal of Loss Prevention in the Process Industries*, vol. 11, no. 4, pp. 261–277, Jul. 1998, doi: 10.1016/S0950-4230(97)00051-X.

[10] P. Swuste, J. Theunissen, P. Schmitz, G. Reniers, and P. Blokland, "Process safety

indicators, a review of literature," *Journal of Loss Prevention in the Process Industries*, vol. 40, pp. 162–173, 2016, doi: 10.1016/j.jlp.2015.12.020.

[11] M. T. Amin, F. Khan, and P. Amyotte, "A bibliometric review of process safety and risk analysis," *Process Safety and Environmental Protection*, vol. 126, pp. 366–381, 2019, doi: 10.1016/j.psep.2019.04.015.

[12] R. Isermann and P. Ballé, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control Engineering Practice*, vol. 5, no. 5, pp. 709–719, May 1997, doi: 10.1016/S0967-0661(97)00053-1.

[13] F. I. Khan and S. . Abbasi, "Major accidents in process industries and an analysis of causes and consequences," *Journal of Loss Prevention in the Process Industries*, vol. 12, no. 5, pp. 361–378, Sep. 1999, doi: 10.1016/S0950-4230(98)00062-X.

[14] I. Duguid, "Take this Safety Database to Heart," *Chemical Engineering*, vol. 108, no. 7, pp. 80–80, Jul. 2001, Accessed: Dec. 04, 2021. [Online]. Available: https://go.gale.com/ps/i.do?p=AONE&sw=w&issn=00092460&v=2.1&it=r&id=GALE%7CA76770524&sid=googleScholar&linkaccess=fulltext.

[15] K. Kidam and M. Hurme, "Analysis of equipment failures as contributors to chemical process accidents," *Process Safety and Environmental Protection*, vol. 91, no. 1–2, pp. 61–78, Jan. 2013, doi: 10.1016/J.PSEP.2012.02.001.

[16] P. Bullemer and J. C. Laberge, "Common operations failure modes in the process industries," *Journal of Loss Prevention in the Process Industries*, vol. 23, no. 6, pp. 928–935, Nov. 2010, doi: 10.1016/J.JLP.2010.05.008.

[17] P. T. Bullemer, L. Kiff, and A. Tharanathan, "Common procedural execution failure modes during abnormal situations," *Journal of Loss Prevention in the Process Industries*, vol. 24, no. 6, pp. 814–818, Nov. 2011, doi: 10.1016/J.JLP.2011.06.007.

[18] F. Eljack and M.-K. Kazi, "Process safety and abnormal situation management," *Current Opinion in Chemical Engineering*, vol. 14, pp. 35–41, Nov. 2016, doi: 10.1016/J.COCHE.2016.07.004.

[19] V. Venkatasubramanian, "Process Fault Detection and Diagnosis: Past, Present and Future," *IFAC Proceedings Volumes*, vol. 34, no. 27, pp. 1–13, Jun. 2001, doi: 10.1016/S1474-6670(17)33563-2.

[20] L. H. Chiang, E. L. Russell, and R. D. Braatz, "Fault Detection and Diagnosis in Industrial Systems," 2001, doi: 10.1007/978-1-4471-0347-9.

[21] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 293–311, Mar. 2003, doi: 10.1016/S0098-

1354(02)00160-6.

[22] Y. Zhang and J. Jiang, “Bibliographical review on reconfigurable fault-tolerant control systems,” *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, Dec. 2008, doi: 10.1016/J.ARCONTROL.2008.03.008.

[23] A. Mouzakitis, “Classification of Fault Diagnosis Methods for Control Systems:,” *https://doi.org/10.1177/0020294013510471*, vol. 46, no. 10, pp. 303–308, Dec. 2013, doi: 10.1177/0020294013510471.

[24] A. Alzghoul, B. Backe, M. Löfstrand, A. Byström, and B. Liljedahl, “Comparing a knowledge-based and a data-driven method in querying data streams for system fault detection: A hydraulic drive system application,” *Computers in Industry*, vol. 65, no. 8, pp. 1126–1135, Oct. 2014, doi: 10.1016/J.COMPIND.2014.06.003.

[25] Y. Zhang and J. Jiang, “Bibliographical review on reconfigurable fault-tolerant control systems,” *Annual Reviews in Control*, vol. 32, no. 2, pp. 229–252, Dec. 2008, doi: 10.1016/J.ARCONTROL.2008.03.008.

[26] J. J. Gertler, “Fault detection and diagnosis in engineering systems,” *Fault Detection and Diagnosis in Engineering Systems*, pp. 1–488, Jan. 2017, doi: 10.1201/9780203756126/FAULT-DETECTION-DIAGNOSIS-ENGINEERING-SYSTEMS-JANOS-GERTLER.

[27] S. X. Ding, “Model-Based Fault Diagnosis Techniques,” 2013, doi: 10.1007/978-1-4471-4799-2.

[28] R. Isermann, “Fault-Diagnosis Applications,” *Fault-Diagnosis Applications*, 2011, doi: 10.1007/978-3-642-12767-0.

[29] X. Ding, L. Guo, and T. Jeinsch, “A Characterization of Parity Space and Its Application to Robust Fault Detection,” *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 337–343, 1999, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4838119.

[30] R. Isermann, “Fault diagnosis of machines via parameter estimation and knowledge processing—Tutorial paper,” *Automatica*, vol. 29, no. 4, pp. 815–835, Jul. 1993, doi: 10.1016/0005-1098(93)90088-B.

[31] E. M. Cimpoesu, B. D. Ciubotaru, and D. Stefanoiu, “Fault detection and diagnosis using parameter estimation with recursive least squares,” *Proceedings - 19th International Conference on Control Systems and Computer Science, CSCS 2013*, pp. 18–23, 2013, doi: 10.1109/CSCS.2013.35.

[32] E. M. Cimpoeşu, B. D. Ciubotaru, and D. Ştefănoiu, “Fault detection and identification using parameter estimation techniques,” *UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science*, vol. 76, no. 2, pp. 3–14, 2014.

[33] X. Wu and Y. Liu, “Leakage detection for hydraulic IGV system in gas turbine compressor with recursive ridge regression estimation,” *Journal of Mechanical Science and Technology*, vol. 31, no. 10, pp. 4551–4556, 2017, doi: 10.1007/s12206-017-0901-y.

[34] N. Verdiere and C. Jauberthie, “Fault detection and identification via bounded-error parameter estimation using distribution theory,” *2019 6th International Conference on Control, Decision and Information Technologies, CoDIT 2019*, pp. 2058–2060, 2019, doi: 10.1109/CoDIT.2019.8820406.

[35] M. Zhou, Z. Cao, M. Zhou, J. Wang, and Z. Wang, “Zonotoptic Fault Estimation for Discrete-Time LPV Systems with Bounded Parametric Uncertainty,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 690–700, 2020, doi: 10.1109/TITS.2019.2898853.

[36] R. J. Patton and J. Chen, “Observer-based fault detection and isolation: Robustness and applications,” 1997, doi: 10.1016/S0967-0661(97)00049-X.

[37] X. Yan, B. Tian, and H. Wang, “An adaptive observer-based fault detection and diagnosis for nonlinear systems with sensor and actuator faults,” *International Conference on Advanced Mechatronic Systems, ICAMechS*, vol. 2015-Octob, pp. 491–496, 2015, doi: 10.1109/ICAMechS.2015.7287160.

[38] H. Jeong, B. Park, S. Park, H. Min, and S. Lee, “Fault detection and identification method using observer-based residuals,” *Reliability Engineering & System Safety*, vol. 184, pp. 27–40, Apr. 2019, doi: 10.1016/J.RESS.2018.02.007.

[39] T. Kobayashi and D. L. Simon, “Evaluation of an Enhanced Bank of Kalman Filters for In-Flight Aircraft Engine Sensor Fault Diagnostics,” *Journal of Engineering for Gas Turbines and Power*, vol. 127, no. 3, pp. 497–504, Jul. 2005, doi: 10.1115/1.1850505.

[40] S. Huang, K. K. Tan, and T. H. Lee, “Fault diagnosis and fault-tolerant control in linear drives using the Kalman filter,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4285–4292, 2012, doi: 10.1109/TIE.2012.2185011.

[41] G. Heredia and A. Ollero, “Sensor fault detection in small autonomous helicopters using observer/Kalman filter identification,” *IEEE 2009 International Conference on Mechatronics, ICM 2009*, vol. 00, no. April, pp. 1–6, 2009, doi: 10.1109/ICMECH.2009.4957236.

[42] R. Li and J. H. Olson, “Fault Detection and Diagnosis in a Closed-Loop Nonlinear Distillation Process: Application of Extended Kalman Filters,” *Industrial and Engineering Chemistry Research*, vol. 30, no. 5, pp. 898–908, 1991, doi: 10.1021/ie00053a012.

[43] R. Oehler, A. Schoenhoff, and M. Schreiber, “Online Model-Based Fault Detection and Diagnosis for a Smart Aircraft Actuator,” *IFAC Proceedings Volumes*, vol. 30, no. 18, pp. 575–580, Aug. 1997, doi: 10.1016/S1474-6670(17)42462-1.

[44] C. Rago, R. Prasanth, R. K. Mehra, and R. Fortenbaugh, "Failure Detection and Identification and Fault Tolerant Control using the IMM-KF with applications to the Eagle-Eye UAV," *Proceedings of the IEEE Conference on Decision and Control*, vol. 4, no. December, pp. 4208–4213, 1998, doi: 10.1109/cdc.1998.761963.

[45] M. H. Amoozgar, A. Chamseddine, and Y. Zhang, "Experimental test of a two-stage kalman filter for actuator fault detection and diagnosis of an unmanned quadrotor helicopter," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 70, no. 1–4, pp. 107–117, 2013, doi: 10.1007/s10846-012-9757-7.

[46] M. Mosallaei, K. Salahshoor, and M. R. Bayat, "Process fault detection and diagnosis by synchronous and asynchronous decentralized kalman filtering using state-vector fusion technique," *Proceedings of the 2007 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP*, pp. 209–214, 2007, doi: 10.1109/ISSNIP.2007.4496845.

[47] S. Jayaram, "A new fast converging Kalman filter for sensor fault detection and isolation," *Sensor Review*, vol. 30, no. 3, pp. 219–224, 2010, doi: 10.1108/02602281011051407.

[48] J. Yu, "A particle filter driven dynamic Gaussian mixture model approach for complex process monitoring and fault diagnosis," *Journal of Process Control*, vol. 22, no. 4, pp. 778–788, Apr. 2012, doi: 10.1016/J.JPROCONT.2012.02.012.

[49] S. Yin and X. Zhu, "Intelligent particle filter and its application to fault detection of nonlinear system," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3852–3861, 2015, doi: 10.1109/TIE.2015.2399396.

[50] X. He, Z. Wang, Y. Liu, and D. H. Zhou, "Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1670–1679, 2013, doi: 10.1109/TII.2013.2251891.

[51] H. M. Odendaal and T. Jones, "Actuator fault detection and isolation: An optimised parity space approach," *Control Engineering Practice*, vol. 26, pp. 222–232, May 2014, doi: 10.1016/J.CONENGPRAC.2014.01.013.

[52] H. Ye, G. Wang, and S. X. Ding, "A New Parity Space Approach for Fault Detection Based on Stationary Wavelet Transform," *IEEE Transactions on Automatic Control*, vol. 49, no. 2, pp. 281–287, Feb. 2004, doi: 10.1109/TAC.2003.822856.

[53] J. Gertler, "Diagnosing parametric," in *Proceedings of 1995 American Control Conference - ACC'95*, 1995, pp. 1615–1620, doi: 10.1109/ACC.1995.529780.

[54] C. M. Fan and Y. P. Lu, "A bayesian framework to integrate knowledge-based and data-driven inference tools for reliable yield diagnoses," *Proceedings - Winter Simulation Conference*, pp. 2323–2329, 2008, doi: 10.1109/WSC.2008.4736337.

[55] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 313–326, Mar. 2003, doi: 10.1016/S0098-1354(02)00161-8.

[56] A. Gandhi, T. Corrigan, and L. Parsa, "Recent advances in modeling and online detection of stator interturn faults in electrical motors," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1564–1575, May 2011, doi: 10.1109/TIE.2010.2089937.

[57] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013, doi: 10.1109/TII.2013.2243743.

[58] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, May 1990, doi: 10.1016/0005-1098(90)90018-D.

[59] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl, "Fault Tree Handbook (NUREG-0492)," p. 209, 1981, doi: NUREG–0492.

[60] G. J. Powers and N. H. Ulerich, "On-Line Hazard Aversion and Fault Diagnosis in Chemical Processes: The Digraph + Fault-Tree Method," *IEEE Transactions on Reliability*, vol. 37, no. 2, pp. 171–177, 1988, doi: 10.1109/24.3738.

[61] J. Antonio, B. Geymayr, N. Francisco, and F. Ebecken, "Fault-Tree Analysis : A Knowledge-Engineering Approach," vol. 44, no. 1, 1995.

[62] S. A. Lapp and G. J. Powers, "Computer-aided Synthesis of Fault-trees," no. April, pp. 2–13, 1977.

[63] J. Shiozaki, B. Shibata, H. Matsuyama, and E. O'Shima, "Fault Diagnosis of Chemical Processes Utilizing Signed Directed Graphs-Improvement by Using Temporal Information," *IEEE Transactions on Industrial Electronics*, vol. 36, no. 4, pp. 469–474, 1989, doi: 10.1109/41.43004.

[64] C. C. Chang and C. C. Yu, "On-Line Fault Diagnosis Using the Signed Directed Graph," *Industrial and Engineering Chemistry Research*, vol. 29, no. 7, pp. 1290–1299, 1990, doi: 10.1021/ie00103a031.

[65] D. Gao, B. Zhang, X. Ma, and C. Wu, "Application of signed directed graph based fault diagnosis of atmospheric distillation unit," *Proceedings - 2010 2nd International Workshop on Intelligent Systems and Applications, ISA 2010*, pp. 1–5, 2010, doi: 10.1109/IWISA.2010.5473271.

[66] M. R. Maurya, R. Rengaswamy, and V. Venkatasubramanian, "A signed directed graph and qualitative trend analysis-based framework for incipient fault diagnosis," *Chemical*

Engineering Research and Design, vol. 85, no. 10 A, pp. 1407–1422, Jan. 2007, doi: 10.1016/S0263-8762(07)73181-7.

[67] D. Mylaraswamy and V. Venkatasubramanian, "A hybrid framework for large scale process fault diagnosis," *Computers & Chemical Engineering*, vol. 21, pp. S935–S940, May 1997, doi: 10.1016/S0098-1354(97)87622-3.

[68] D.-H. Ham and W. C. Yoon, "The effects of presenting functionally abstracted information in fault diagnosis tasks," *Reliability Engineering & System Safety*, vol. 73, no. 2, pp. 103–119, Aug. 2001, doi: 10.1016/S0951-8320(01)00053-9.

[69] A. M. Bisantz and K. J. Vicente, "Making the abstraction hierarchy concrete," *International Journal of Human - Computer Studies*, 1994, doi: 10.1006/ijhc.1994.1005.

[70] M. Lind, "Making Sense of the Abstraction Hierarchy," *Cognition, Technology & Work*, vol. 5, no. 2, pp. 67–81, 1999.

[71] A. Lemos, W. Caminhas, and F. Gomide, "Adaptive fault detection and diagnosis using an evolving fuzzy classifier," *Information Sciences*, vol. 220, pp. 64–85, Jan. 2013, doi: 10.1016/J.INS.2011.08.030.

[72] C. Nan, F. Khan, and M. T. Iqbal, "Real-time fault diagnosis using knowledge-based expert system," *Process Safety and Environmental Protection*, 2008, doi: 10.1016/j.psep.2007.10.014.

[73] C. H. Lo, P. T. Chan, Y. K. Wong, A. B. Rad, and K. L. Cheung, "Fuzzy-genetic algorithm for automatic fault detection in HVAC systems," *Applied Soft Computing*, vol. 7, no. 2, pp. 554–560, Mar. 2007, doi: 10.1016/J.ASOC.2006.06.003.

[74] K. S. Yap, S. Y. Wong, and S. K. Tiong, "Compressing and improving fuzzy rules using genetic algorithm and its application to fault detection," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, pp. 1–4, 2013, doi: 10.1109/ETFA.2013.6648106.

[75] M. Ram Maurya, R. Rengaswamy, and V. Venkatasubramanian, "Fault diagnosis by qualitative trend analysis of the principal components: Prospects and some new results," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 36, no. 5, pp. 783–788, 2003, doi: 10.1016/S1474-6670(17)36588-6.

[76] M. R. Maurya, P. K. Paritosh, R. Rengaswamy, and V. Venkatasubramanian, "A framework for on-line trend extraction and fault diagnosis," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 6, pp. 950–960, Sep. 2010, doi: 10.1016/J.ENGAPPAI.2010.01.027.

[77] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee

Eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, Oct. 2012, doi: 10.1016/J.JPROCONT.2012.06.009.

[78] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part III: Process history based methods," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 327–346, Mar. 2003, doi: 10.1016/S0098-1354(02)00162-X.

[79] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 171, pp. 16–25, Dec. 2017, doi: 10.1016/J.CHEMOLAB.2017.09.021.

[80] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418–6428, 2014, doi: 10.1109/TIE.2014.2301773.

[81] R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, "Fault detection and diagnosis in process system using artificial intelligence-based cognitive technique," *Computers and Chemical Engineering*, vol. 134, p. 106697, Mar. 2020, doi: 10.1016/j.compchemeng.2019.106697.

[82] T. J. Harris, C. T. Seppala, and L. D. Desborough, "A review of performance monitoring and assessment techniques for univariate and multivariate control systems," *Journal of Process Control*, vol. 9, no. 1, pp. 1–17, Feb. 1999, doi: 10.1016/S0959-1524(98)00031-6.

[83] M. Kano, S. Tanaka, S. Hasebe, I. Hashimoto, and H. Ohno, "Monitoring independent components for fault detection," *AIChE Journal*, vol. 49, no. 4, pp. 969–976, 2003, doi: 10.1002/aic.690490414.

[84] J. F. MacGregor and T. Kourti, "Statistical process control of multivariate processes," *Control Engineering Practice*, vol. 3, no. 3, pp. 403–414, Mar. 1995, doi: 10.1016/0967-0661(95)00014-L.

[85] T. Kourti and J. F. MacGregor, "Process analysis, monitoring and diagnosis, using multivariate projection methods," *Chemometrics and Intelligent Laboratory Systems*. 1995, doi: 10.1016/0169-7439(95)80036-9.

[86] M. Alauddin, F. Khan, S. Imtiaz, and S. Ahmed, "A Bibliometric Review and Analysis of Data-Driven Fault Detection and Diagnosis Methods for Process Systems," *Industrial and Engineering Chemistry Research*, vol. 57, no. 32, pp. 10719–10735, 2018, doi: 10.1021/acs.iecr.8b00936.

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

[88] Z. Ge and Z. Song, “Multivariate Statistical Process Control,” 2013, doi: 10.1007/978-1-4471-4513-4.

[89] S. X. Ding, “Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems,” 2014, doi: 10.1007/978-1-4471-6410-4.

[90] R. Liu, B. Yang, E. Zio, and X. Chen, “Artificial intelligence for fault diagnosis of rotating machinery: A review,” *Mechanical Systems and Signal Processing*, vol. 108, pp. 33–47, Aug. 2018, doi: 10.1016/J.YMSSP.2018.02.016.

[91] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, “Applications of machine learning to machine fault diagnosis: A review and roadmap,” *Mechanical Systems and Signal Processing*, vol. 138, p. 106587, Apr. 2020, doi: 10.1016/J.YMSSP.2019.106587.

[92] N. G. Lo, J. M. Flaus, and O. Adrot, “Review of Machine Learning Approaches in Fault Diagnosis applied to IoT Systems,” *2019 International Conference on Control, Automation and Diagnosis, ICCAD 2019 - Proceedings*, 2019, doi: 10.1109/ICCAD46983.2019.9037949.

[93] J. C. Hoskins and D. M. Himmelblau, “Artificial neural network models of knowledge representation in chemical engineering,” *Computers and Chemical Engineering*, vol. 12, no. 9–10, pp. 881–890, 1988, doi: 10.1016/0098-1354(88)87015-7.

[94] V. Venkatasubramanian and K. Chan, “A neural network methodology for process fault diagnosis,” *AIChE Journal*, vol. 35, no. 12, pp. 1993–2002, 1989, doi: 10.1002/aic.690351210.

[95] R. Lit, J. H. Olsont, and D. L. Chester, “Dynamic Fault Detection and Diagnosis Using Neural Networks,” *IEEE*, pp. 1169–1174, 1990.

[96] V. N. Vapnik, “The Nature of Statistical Learning Theory,” *The Nature of Statistical Learning Theory*, 1995, doi: 10.1007/978-1-4757-2440-0.

[97] V. N. Vapnik, “Statistical learning theory,” p. 736, 1998, Accessed: Dec. 05, 2021. [Online]. Available: https://www.wiley.com/en-ie/Statistical+Learning+Theory-p-9780471030034.

[98] C. Corties and V. Vapnik, “Support-vector network,” *Machine Learning*, vol. 20, pp. 1–25, 1995, Accessed: Oct. 07, 2020. [Online]. Available: http://ci.nii.ac.jp/naid/10029626531/en/.

[99] V. Vapnik, S. E. Golowich, and A. Smola, “Support vector method for function approximation, regression estimation, and signal processing,” *Advances in Neural Information Processing Systems*, pp. 281–287, 1997.

[100] O. Chapelle, P. Haffner, and V. N. Vapnik, “Support vector machines for histogram-based image classification,” *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1055–1064, 1999, doi: 10.1109/72.788646.

[101] H. Li, F. Qi, and S. Wang, "A comparison of model selection methods for multi-class support vector machines," *Lecture Notes in Computer Science*, vol. 3483, no. IV, pp. 1140–1148, 2005, doi: 10.1007/11424925_119.

[102] H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, "Unsupervised Learning Based On Artificial Neural Network: A Review," *2018 IEEE International Conference on Cyborg and Bionic Systems, CBS 2018*, pp. 322–327, 2019, doi: 10.1109/CBS.2018.8612259.

[103] A. Purarjomandlangrudi, G. Nourbakhsh, M. Esmalifalak, and A. Tan, "Fault detection in wind Turbine: A systematic literature review," *Wind Engineering*, vol. 37, no. 5, pp. 535–548, 2013, doi: 10.1260/0309-524X.37.5.535.

[104] M. A. Bin Shams, H. M. Budman, and T. A. Duever, "Fault detection, identification and diagnosis using CUSUM based PCA," *Chemical Engineering Science*, vol. 66, no. 20, pp. 4488–4498, Oct. 2011, doi: 10.1016/J.CES.2011.05.028.

[105] M. Ahmed, M. Baqqar, F. Gu, and A. D. Ball, "Fault detection and diagnosis using Principal Component Analysis of vibration data from a reciprocating compressor," *Proceedings of the 2012 UKACC International Conference on Control, CONTROL 2012*, no. September, pp. 461–466, 2012, doi: 10.1109/CONTROL.2012.6334674.

[106] H. Bendjama, M. S. Boucherit, S. Bouhouche, and J. Bast, "Fault detection and diagnosis using principal component analysis. Application to low pressure lost foam casting process," *International Journal of Modelling, Identification and Control*, vol. 14, no. 1–2, pp. 102–111, 2011, doi: 10.1504/IJMIC.2011.042344.

[107] S. Heo and J. H. Lee, "Fault detection and classification using artificial neural networks," *IFAC-PapersOnLine*, 2018, doi: 10.1016/j.ifacol.2018.09.380.

[108] T. Sorsa and H. N. Koivo, "Application of artificial neural networks in process fault diagnosis," *Automatica*, vol. 29, no. 4, pp. 843–849, Jul. 1993, doi: 10.1016/0005-1098(93)90090-G.

[109] A. Widodo and B. S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*. 2007, doi: 10.1016/j.ymssp.2006.12.007.

[110] S. Mahadevan and S. L. Shah, "Fault detection and diagnosis in process data using one-class support vector machines," *Journal of Process Control*, vol. 19, no. 10, pp. 1627–1639, Dec. 2009, doi: 10.1016/J.JPROCONT.2009.07.011.

[111] J. S. L. Senanayaka, S. T. Kandukuri, H. Van Khang, and K. G. Robbersmyr, "Early detection and classification of bearing faults using support vector machine algorithm," *Proceedings - 2017 IEEE Workshop on Electrical Machines Design, Control and Diagnosis, WEMDCD 2017*, pp. 250–255, 2017, doi: 10.1109/WEMDCD.2017.7947755.

[112] S. Yin, X. Zhu, and C. Jing, “Fault detection based on a robust one class support vector machine,” *Neurocomputing*, vol. 145, pp. 263–268, Dec. 2014, doi: 10.1016/j.neucom.2014.05.035.

[113] F. Harrou, A. Dairi, B. Taghezouit, and Y. Sun, “An unsupervised monitoring procedure for detecting anomalies in photovoltaic systems using a one-class Support Vector Machine,” *Solar Energy*, vol. 179, pp. 48–58, Feb. 2019, doi: 10.1016/J.SOLENER.2018.12.045.

[114] N. Amruthnath and T. Gupta, “A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance,” *2018 5th International Conference on Industrial Engineering and Applications, ICIEA 2018*, no. August 1993, pp. 355–361, 2018, doi: 10.1109/IEA.2018.8387124.

[115] Jie Yu, “A Support Vector Clustering-Based Probabilistic Method for Unsupervised Fault Detection and Classification of Complex Chemical Processes Using Unlabeled Data,” *AIChE Journal*, vol. 59, no. 4, pp. 407–419, 2012, doi: 10.1002/aic.

[116] K. Yan, J. Huang, W. Shen, and Z. Ji, “Unsupervised learning for fault detection and diagnosis of air handling units,” *Energy and Buildings*, vol. 210, p. 109689, Mar. 2020, doi: 10.1016/J.ENBUILD.2019.109689.

[117] R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, “Fault detection and diagnosis in process system using artificial intelligence-based cognitive technique,” *Computers & Chemical Engineering*, vol. 134, p. 106697, Mar. 2020, doi: 10.1016/J.COMPCHEMENG.2019.106697.

[118] Y. Zhang, L. You, and C. Jia, “Fault detection and diagnosis using Bayesian-network inference,” *Proceedings IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, vol. 2017-Janua, no. 3, pp. 5049–5053, 2017, doi: 10.1109/IECON.2017.8216872.

[119] U. Lerner, R. Parr, D. Koller, and G. Biswas, “Bayesian Fault Detection and Diagnosis in Dynamic Systems,” *Seventeenth National Conference on Artificial Intelligence (AAAI-00)*, no. August, pp. 531–537, 2000.

[120] Z. Zhang and F. Dong, “Fault detection and diagnosis for missing data systems with a three time-slice dynamic Bayesian network approach,” *Chemometrics and Intelligent Laboratory Systems*, vol. 138, pp. 30–40, Nov. 2014, doi: 10.1016/J.CHEMOLAB.2014.07.009.

[121] M. A. Atoui, S. Verron, and A. Kobi, “Fault Detection and Diagnosis in a Bayesian Network classifier incorporating probabilistic boundary1,” *IFAC-PapersOnLine*, vol. 48, no. 21, pp. 670–675, Jan. 2015, doi: 10.1016/J.IFACOL.2015.09.604.

[122] Y. Zhao, J. Tong, L. Zhang, and G. Wu, “Diagnosis of operational failures and on-demand failures in nuclear power plants: An approach based on dynamic Bayesian networks,” *Annals of Nuclear Energy*, vol. 138, p. 107181, Apr. 2020, doi:

10.1016/J.ANUCENE.2019.107181.

[123] Y. Zhangt, R. Lit, and K. Zhous, "A Fault Detection and Diagnosis Approach Based on Hidden Markov Chain Model," in *Proceedings of the American Control Conferenc*, 1998, no. June, pp. 2012–2016.

[124] M. M. Rashid and J. Yu, "Hidden markov model based adaptive independent component analysis approach for complex chemical process monitoring and fault detection," *Industrial and Engineering Chemistry Research*, vol. 51, no. 15, pp. 5506–5514, 2012, doi: 10.1021/ie300203u.

[125] M. Galagedarage Don and F. Khan, "Process fault prognosis using hidden markov model-bayesian networks hybrid model," *Industrial and Engineering Chemistry Research*, vol. 58, no. 27, pp. 12041–12053, 2019, doi: 10.1021/acs.iecr.9b00524.

[126] A. Kouadri *et al.*, "Hidden Markov model based principal component analysis for intelligent fault diagnosis of wind energy converter systems," *Renewable Energy*, vol. 150, pp. 598–606, May 2020, doi: 10.1016/J.RENENE.2020.01.010.

[127] D.-T. Hoang and H.-J. Kang, "A survey on Deep Learning based bearing fault diagnosis," *Neurocomputing*, vol. 335, pp. 327–335, Mar. 2019, doi: 10.1016/J.NEUCOM.2018.06.078.

[128] H. Wu and J. Zhao, "Fault detection and diagnosis based on transfer learning for multimode chemical processes," *Computers & Chemical Engineering*, vol. 135, p. 106731, Apr. 2020, doi: 10.1016/J.COMPCHEMENG.2020.106731.

[129] F. Jia, Y. Lei, L. Guo, J. Lin, and S. Xing, "A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines," *Neurocomputing*, vol. 272, pp. 619–628, Jan. 2018, doi: 10.1016/J.NEUCOM.2017.07.032.

[130] S. Khan and T. Yairi, "A review on the application of deep learning in system health management," *Mechanical Systems and Signal Processing*, vol. 107, pp. 241–265, Jul. 2018, doi: 10.1016/J.YMSSP.2017.11.024.

[131] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.

[132] S. R. Saufi, Z. A. Bin Ahmad, M. S. Leong, and M. H. Lim, "Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review," *IEEE Access*, vol. 7, pp. 122644–122662, 2019, doi: 10.1109/ACCESS.2019.2938227.

[133] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417–420, Jan. 2016, doi: 10.1016/J.CIRP.2016.04.072.

[134] F. Shaheen, B. Verma, and M. Asafuddoula, "Impact of Automatic Feature Extraction in

Deep Learning Architecture," *2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016*, pp. 1–8, 2016, doi: 10.1109/DICTA.2016.7797053.

[135] W. Yu and C. Zhao, "Broad Convolutional Neural Network Based Industrial Process Fault Diagnosis with Incremental Learning Capability," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 5081–5091, 2020, doi: 10.1109/TIE.2019.2931255.

[136] L. Wen, X. Li, L. Gao, and Y. Zhang, "A New Convolutional Neural Network-Based Data-Driven Fault Diagnosis Method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018, doi: 10.1109/TIE.2017.2774777.

[137] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016, doi: 10.1109/TIE.2016.2582729.

[138] L. Ren, Y. Sun, H. Wang, and L. Zhang, "Prediction of bearing remaining useful life with deep convolution neural network," *IEEE Access*, vol. 6, pp. 13041–13049, 2018, doi: 10.1109/ACCESS.2018.2804930.

[139] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, Apr. 2018, doi: 10.1016/J.RESS.2017.11.021.

[140] B. Yang, R. Liu, and E. Zio, "Remaining useful life prediction based on a double-convolutional neural network architecture," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9521–9530, 2019, doi: 10.1109/TIE.2019.2924605.

[141] Z. Zhang and J. Chen, "Fault detection and diagnosis based on particle filters combined with interactive multiple-model estimation in dynamic process systems," *ISA Transactions*, vol. 85, pp. 247–261, 2019, doi: 10.1016/j.isatra.2018.10.015.

[142] X. Xu, Q. Wu, X. Li, and B. Huang, "Dilated Convolution Neural Network for Remaining Useful Life Prediction," *Journal of Computing and Information Science in Engineering*, vol. 20, no. 2, 2020, doi: 10.1115/1.4045293.

[143] Y. Qu, X. Ming, S. Qiu, M. Zheng, and Z. Hou, "An integrative framework for online prognostic and health management using internet of things and convolutional neural network," *Sensors (Switzerland)*, vol. 19, no. 10, 2019, doi: 10.3390/s19102338.

[144] L. Guo, N. Li, F. Jia, Y. Lei, and J. Lin, "A recurrent neural network based health indicator for remaining useful life prediction of bearings," *Neurocomputing*, vol. 240, pp. 98–109, May 2017, doi: 10.1016/J.NEUCOM.2017.02.045.

[145] Y. Wu, M. Yuan, S. Dong, L. Lin, and Y. Liu, "Remaining useful life estimation of engineered systems using vanilla LSTM neural networks," *Neurocomputing*, vol. 275, pp.

167–179, 2018, doi: 10.1016/j.neucom.2017.05.063.

[146] M. Ma and Z. Mao, "Deep recurrent convolutional neural network for remaining useful life prediction," *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019*, pp. 1–4, 2019, doi: 10.1109/ICPHM.2019.8819440.

[147] Y. Chen, G. Peng, Z. Zhu, and S. Li, "A novel deep learning method based on attention mechanism for bearing remaining useful life prediction," *Applied Soft Computing*, vol. 86, p. 105919, Jan. 2020, doi: 10.1016/J.ASOC.2019.105919.

[148] Q. Cui, Z. Li, J. Yang, and B. Liang, "Rolling bearing fault prognosis using recurrent neural network," in *29th Chinese Control and Decision Conference, CCDC 2017*, 2017, pp. 1196–1201, doi: 10.1109/CCDC.2017.7978700.

[149] H. Cheng, J. Wu, Y. Liu, and D. Huang, "A novel fault identification and root-causality analysis of incipient faults with applications to wastewater treatment processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 188, pp. 24–36, May 2019, doi: 10.1016/j.chemolab.2019.03.004.

[150] L. Liu, Q. Guo, D. Liu, and Y. Peng, "Data-Driven Remaining Useful Life Prediction Considering Sensor Anomaly Detection and Data Recovery," *IEEE Access*, vol. 7, pp. 58336–58345, 2019, doi: 10.1109/ACCESS.2019.2914236.

[151] L. Tang, S. Zhang, X. Yang, and S. Hu, "Research on Prognosis for Engines by LSTM Deep Learning Method," *2019 Prognostics and System Health Management Conference, PHAI-Qingdao 2019*, pp. 1–8, 2019, doi: 10.1109/PHM-Qingdao46334.2019.8942976.

[152] B. Chinomona, C. Chung, L. K. Chang, W. C. Su, and M. C. Tsai, "Long short-term memory approach to estimate battery remaining useful life using partial data," *IEEE Access*, vol. 8, pp. 165419–165431, 2020, doi: 10.1109/ACCESS.2020.3022505.

[153] Y. Su, F. Tao, J. Jin, T. Wang, Q. Wang, and L. Wang, "Failure Prognosis of Complex Equipment with Multistream Deep Recurrent Neural Network," *Journal of Computing and Information Science in Engineering*, vol. 20, no. 2, Jan. 2020, doi: 10.1115/1.4045445/1068285.

[154] H. Wu and J. Zhao, "Fault detection and diagnosis based on transfer learning for multimode chemical processes," *Computers & Chemical Engineering*, vol. 135, p. 106731, Apr. 2020, doi: 10.1016/J.COMPCHEMENG.2020.106731.

[155] S. Zhang, K. Bi, and T. Qiu, "Bidirectional Recurrent Neural Network-Based Chemical Process Fault Diagnosis," *Industrial and Engineering Chemistry Research*, vol. 59, no. 2, pp. 824–834, 2020, doi: 10.1021/acs.iecr.9b05885.

[156] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and*

Data Mining, vol. Part F1296, pp. 665–674, 2017, doi: 10.1145/3097983.3098052.

[157] K. Deng *et al.*, "A Remaining Useful Life Prediction Method with Automatic Feature Extraction for Aircraft Engines," *Proceedings - 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE 2019*, vol. 2019-Janua, pp. 689–692, 2019, doi: 10.1109/TrustCom/BigDataSE.2019.00097.

[158] P. Lin and J. Tao, "A novel bearing health indicator construction method based on ensemble stacked autoencoder," *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019*, pp. 1–9, 2019, doi: 10.1109/ICPHM.2019.8819405.

[159] P. Lv, Y. Yu, Y. Fan, X. Tang, and X. Tong, "Layer-constrained variational autoencoding kernel density estimation model for anomaly detection," *Knowledge-Based Systems*, p. 105753, Mar. 2020, doi: 10.1016/J.KNOSYS.2020.105753.

[160] M. S. Kim, J. P. Yun, S. Lee, and P. Park, "Unsupervised Anomaly detection of LM Guide Using Variational Autoencoder," *2019 11th International Symposium on Advanced Topics in Electrical Engineering, ATEE 2019*, pp. 1–5, 2019, doi: 10.1109/ATEE.2019.8724998.

[161] C. Su, L. Li, and Z. Wen, "Remaining useful life prediction via a variational autoencoder and a time-window-based sequence neural network," *Quality and Reliability Engineering International*, vol. 36, no. 5, pp. 1639–1656, Jul. 2020, doi: 10.1002/QRE.2651.

[162] D. Verstraete, E. Droguett, and M. Modarres, "A deep adversarial approach based on multi-sensor fusion for semi-supervised remaining useful life prognostics," *Sensors (Switzerland)*, vol. 20, no. 1, 2020, doi: 10.3390/s20010176.

[163] D. Wulsin, J. Blanco, R. Mani, and B. Litt, "Semi-supervised anomaly detection for EEG waveforms using deep belief nets," *Proceedings - 9th International Conference on Machine Learning and Applications, ICMLA 2010*, pp. 436–441, 2010, doi: 10.1109/ICMLA.2010.71.

[164] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2306–2318, 2017, doi: 10.1109/TNNLS.2016.2582798.

[165] Z. Zhang and J. Zhao, "A deep belief network based fault diagnosis model for complex chemical processes," *Computers & Chemical Engineering*, vol. 107, pp. 395–407, Dec. 2017, doi: 10.1016/J.COMPCHEMENG.2017.02.041.

[166] Y. Wang, Z. Pan, X. Yuan, C. Yang, and W. Gui, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," *ISA Transactions*, vol. 96, pp. 457–467, Jan. 2020, doi: 10.1016/J.ISATRA.2019.07.001.

[167] G. Hu, H. Li, Y. Xia, and L. Luo, "A deep Boltzmann machine and multi-grained scanning forest ensemble collaborative method and its application to industrial fault diagnosis," *Computers in Industry*, vol. 100, pp. 287–296, Sep. 2018, doi: 10.1016/J.COMPIND.2018.04.002.

[168] T. Xia, Y. Song, Y. Zheng, E. Pan, and L. Xi, "An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation," *Computers in Industry*, vol. 115, p. 103182, Feb. 2020, doi: 10.1016/J.COMPIND.2019.103182.

[169] Q. An, Z. Tao, X. Xu, M. El Mansori, and M. Chen, "A data-driven model for milling tool remaining useful life prediction with convolutional and stacked LSTM network," *Measurement*, vol. 154, p. 107461, Mar. 2020, doi: 10.1016/J.MEASUREMENT.2019.107461.

[170] G. Yue, G. Ping, and L. Lanxin, "An End-to-End model based on CNN-LSTM for Industrial Fault Diagnosis and Prognosis," *IEEE Proceedings of IC-NIDC 2018*, pp. 3–7, 2018.

[171] P. Park, P. Di Marco, H. Shin, and J. Bang, "Fault detection and diagnosis using combined autoencoder and long short-term memory network," *Sensors (Switzerland)*, vol. 19, no. 21, pp. 1–17, 2019, doi: 10.3390/s19214612.

[172] D. H. Rothenberg, "Alarm management for process control : a best-practice guide for design, implementation, and use of industrial alarm systems."

[173] J. Hu, S. Cai, and L. Zhang, "Alarm Management Technology and Its Progress in Process Industries," vol. 134, no. CAAI, pp. 278–281, 2017, doi: 10.2991/caai-17.2017.63.

[174] B. R. Mehta and Y. J. Reddy, "Alarm management systems," *Industrial Process Automation Systems*, pp. 569–582, Jan. 2015, doi: 10.1016/b978-0-12-800939-0.00021-8.

[175] J. S. Alford, J. Kindervater, and R. Stankovich, "Alarm management for regulated industries," *Chemical Engineering Progress*, vol. 101, no. 4, pp. 25–30, 2005.

[176] J. Wang, F. Yang, T. Chen, and S. L. Shah, "An Overview of Industrial Alarm Systems: Main Causes for Alarm Overloading, Research Status, and Open Problems," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 1045–1061, 2016, doi: 10.1109/TASE.2015.2464234.

[177] P. Goel, A. Datta, and M. S. Mannan, "Industrial alarm systems: Challenges and opportunities," *Journal of Loss Prevention in the Process Industries*, vol. 50, pp. 23–36, 2017, doi: 10.1016/j.jlp.2017.09.001.

[178] Y. Cheng, I. Izadi, and T. Chen, "Pattern matching of alarm flood sequences by a modified Smith–Waterman algorithm," *Chemical Engineering Research and Design*, vol. 91, no. 6, pp. 1085–1094, 2013, doi: https://doi.org/10.1016/j.cherd.2012.11.001.

[179] V. Rodrigo, M. Chioua, T. Hagglund, and M. Hollender, “Causal analysis for alarm flood reduction,” *IFAC-PapersOnLine*, 2016, doi: 10.1016/j.ifacol.2016.07.269.

[180] W. Hu, T. Chen, and S. L. Shah, “Detection of Frequent Alarm Patterns in Industrial Alarm Floods Using Itemset Mining Methods,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 9, pp. 7290–7300, 2018, doi: 10.1109/TIE.2018.2795573.

[181] T. Niyazmand and I. Izadi, “Pattern mining in alarm flood sequences using a modified PrefixSpan algorithm,” *ISA Transactions*, vol. 90, pp. 287–293, Jul. 2019, doi: 10.1016/J.ISATRA.2018.12.050.

[182] J. Shang and T. Chen, “Early Classification of Alarm Floods via Exponentially Attenuated Component Analysis,” *IEEE Transactions on Industrial Electronics*, vol. 0046, no. c, pp. 1–1, 2019, doi: 10.1109/tie.2019.2949542.

[183] M. Lucke, M. Chioua, C. Grimholt, M. Hollender, and N. F. Thornhill, “Advances in alarm data analysis with a practical application to online alarm flood classification,” *Journal of Process Control*, vol. 79, pp. 56–71, Jul. 2019, doi: 10.1016/J.JPROCONT.2019.04.010.

[184] M. Fullen, P. Schüller, and O. Niggemann, “Semi-supervised Case-based Reasoning Approach to Alarm Flood Analysis,” pp. 53–61, 2020, doi: 10.1007/978-3-662-59084-3_7.

[185] A. Adhitya, S. F. Cheng, Z. Lee, and R. Srinivasan, “Quantifying the effectiveness of an alarm management system through human factors studies,” *Computers & Chemical Engineering*, vol. 67, pp. 1–12, Aug. 2014, doi: 10.1016/J.COMPCHEMENG.2014.03.013.

[186] W. Hu, T. Chen, and G. Meyer, “Constructing workflow models of alarm responses via trace labeling and dependency analysis,” *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2019*, pp. 1615–1620, 2019, doi: 10.23919/SICE.2019.8859934.

[187] J. Hu and Y. Yi, “A two-level intelligent alarm management framework for process safety,” *Safety Science*, vol. 82, pp. 432–444, Feb. 2016, doi: 10.1016/J.SSCI.2015.10.005.

[188] P. Goel, E. N. Pistikopoulos, M. S. Mannan, and A. Datta, “A data-driven alarm and event management framework,” *Journal of Loss Prevention in the Process Industries*, vol. 62, p. 103959, Nov. 2019, doi: 10.1016/J.JLP.2019.103959.

[189] A. M. P. Chacón and F. P. G. Márquez, “False Alarms Management by Data Science,” *Data Science and Digital Business*, pp. 301–316, Jan. 2019, doi: 10.1007/978-3-319-95651-0_15.

[190] A. Hidri, A. Selmi, and M. S. Hidri, “Discovery of Frequent Patterns of Episodes within a Time Window for Alarm Management Systems,” *IEEE Access*, vol. 8, pp. 11061–11073, 2020, doi: 10.1109/ACCESS.2020.2965647.

[191] B. R. Hollifield and E. Habibi, “The alarm management handbook : a comprehensive guide,” 2006.

[192] Y. Wang and M. Rausand, “Reliability analysis of safety-instrumented systems operated in high-demand mode,” *Journal of Loss Prevention in the Process Industries*, vol. 32, pp. 254–264, Nov. 2014, doi: 10.1016/J.JLP.2014.09.007.

[193] M. A. Lundteigen and M. Rausand, “Instrumented Systems: Where to Direct Future Research?,” *Process Safety Progress*, vol. 25, no. 4, pp. 326–330, 2010, doi: 10.1002/prs.

[194] H. Guo and X. Yang, “A simple reliability block diagram method for safety integrity verification,” *Reliability Engineering and System Safety*, 2007, doi: 10.1016/j.ress.2006.08.002.

[195] Y. Misumi and Y. Sato, “Estimation of average hazardous-event-frequency for allocation of safety-integrity levels,” *Reliability Engineering & System Safety*, vol. 66, no. 2, pp. 135–144, Nov. 1999, doi: 10.1016/S0951-8320(99)00030-7.

[196] L. Beckman, “Easily assess complex safety loops,” *Chemical Engineering Progress*, vol. 97, no. 3, pp. 57–59, 2001.

[197] J. R. Belland and D. Wiseman, “Using fault trees to analyze safety-instrumented systems,” *Proceedings - Annual Reliability and Maintainability Symposium*, vol. 2016-April, pp. 1–6, 2016, doi: 10.1109/RAMS.2016.7448056.

[198] J. V. Bukowski, “Incorporating process demand into models for assessment of safety system performance,” *Proceedings - Annual Reliability and Maintainability Symposium*, vol. 00, no. C, pp. 577–581, 2006, doi: 10.1109/RAMS.2006.1677435.

[199] Y. Langeron, A. Barros, A. Grall, and C. Bérenguer, “Combination of safety integrity levels (SILs): A study of IEC61508 merging rules,” *Journal of Loss Prevention in the Process Industries*, vol. 21, no. 4, pp. 437–449, Jul. 2008, doi: 10.1016/J.JLP.2008.02.003.

[200] Y. Liu and M. Rausand, “Reliability assessment of safety instrumented systems subject to different demand modes,” *Journal of Loss Prevention in the Process Industries*, vol. 24, no. 1, pp. 49–56, Jan. 2011, doi: 10.1016/J.JLP.2010.08.014.

[201] F. Innal, Y. Dutuit, A. Rauzy, and J. P. Signoret, “New insight into the average probability of failure on demand and the probability of dangerous failure per hour of safety instrumented systems,” *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 224, no. 2, pp. 75–86, 2010, doi: 10.1243/1748006XJRR278.

[202] Y. Liu and M. Rausand, “Reliability effects of test strategies on safety-instrumented systems in different demand modes,” *Reliability Engineering & System Safety*, vol. 119, pp. 235–243, Nov. 2013, doi: 10.1016/J.RESS.2013.06.035.

[203] M. Catelani, L. Ciani, and M. Venzi, “Sensitivity analysis with MC simulation for the failure rate evaluation and reliability assessment,” *Measurement*, vol. 74, pp. 150–158, Oct. 2015,

doi: 10.1016/J.MEASUREMENT.2015.07.003.

[204] G. Kaczor, S. Młynarski, and M. Szkoda, “Verification of safety integrity level with the application of Monte Carlo simulation and reliability block diagrams,” *Journal of Loss Prevention in the Process Industries*, vol. 41, pp. 31–39, May 2016, doi: 10.1016/J.JLP.2016.03.002.

[205] F. Khan, S. Rathnayaka, and S. Ahmed, “Methods and models in process safety and risk management: Past, present and future,” *Process Safety and Environmental Protection*, vol. 98, pp. 116–147, Nov. 2015, doi: 10.1016/J.PSEP.2015.07.005.

[206] F. Khan, H. Abunada, D. John, and T. Benmosbah, “Development of risk-based process safety indicators,” *Process Safety Progress*, vol. 29, no. 2, pp. 133–143, 2010, doi: 10.1002/prs.10354.

[207] M. T. Amin, F. Khan, and M. J. Zuo, “A bibliometric analysis of process system failure and reliability literature,” *Engineering Failure Analysis*, vol. 106. Elsevier Ltd, p. 104152, Dec. 01, 2019, doi: 10.1016/j.engfailanal.2019.104152.

[208] Y. Dai, H. Wang, F. Khan, and J. Zhao, “Abnormal situation management for smart chemical process operation,” *Current Opinion in Chemical Engineering*, vol. 14, pp. 49–55, Nov. 2016, doi: 10.1016/J.COCHE.2016.07.009.

[209] “Guidelines for Hazard Evaluation Procedures, 3rd Edition | AIChE.” https://www.aiche.org/resources/publications/books/guidelines-hazard-evaluation-procedures-3rd-edition (accessed Dec. 05, 2021).

[210] B. J. Tyler, “HAZOP study training from the 1970s to today,” *Process Safety and Environmental Protection*, vol. 90, no. 5, pp. 419–423, Sep. 2012, doi: 10.1016/J.PSEP.2012.07.003.

[211] Z. H. Qureshi, “A Review of Accident Modelling Approaches for Complex Critical Sociotechnical Systems,” *12th Australian Workshop on Safety Related Programmable Systems (SCS’07), Adelaide*, vol. 86, pp. 47–59, 2008.

[212] A. Al-shanini, A. Ahmad, and F. Khan, “Accident modelling and analysis in process industries,” *Journal of Loss Prevention in the Process Industries*, vol. 32, pp. 319–334, Nov. 2014, doi: 10.1016/J.JLP.2014.09.016.

[213] G. Fu, X. Xie, Q. Jia, Z. Li, P. Chen, and Y. Ge, “The development history of accident causation models in the past 100 years: 24Model, a more modern accident causation model,” *Process Safety and Environmental Protection*, vol. 134, pp. 47–82, Feb. 2020, doi: 10.1016/J.PSEP.2019.11.027.

[214] E. Hollnagel, “Understanding accidents - From root causes to performance variability,” *IEEE Conference on Human Factors and Power Plants*, no. July, pp. 1–6, 2002, doi:

10.1109/hfpp.2002.1042821.

[215] A. Yousefi, M. Rodriguez Hernandez, and V. Lopez Peña, "Systemic accident analysis models: A comparison study between AcciMap, FRAM, and STAMP," *Process Safety Progress*, vol. 38, no. 2, p. e12002, Jun. 2019, doi: 10.1002/PRS.12002.

[216] N. Khakzad, F. Khan, and P. Amyotte, "Dynamic safety analysis of process systems by mapping bow-tie into Bayesian network," *Process Safety and Environmental Protection*, vol. 91, no. 1–2, pp. 46–53, Jan. 2013, doi: 10.1016/J.PSEP.2012.01.005.

[217] X. Meng, G. Chen, G. Zhu, and Y. Zhu, "Dynamic quantitative risk assessment of accidents induced by leakage on offshore platforms using DEMATEL-BN," *International Journal of Naval Architecture and Ocean Engineering*, vol. 11, no. 1, pp. 22–32, Jan. 2019, doi: 10.1016/J.IJNAOE.2017.12.001.

[218] M. Taleb-Berrouane, F. Khan, and P. Amyotte, "Bayesian Stochastic Petri Nets (BSPN) - A new modelling tool for dynamic safety and reliability analysis," *Reliability Engineering & System Safety*, vol. 193, p. 106587, Jan. 2020, doi: 10.1016/J.RESS.2019.106587.

[219] J. Tixier, G. Dusserre, O. Salvi, and D. Gaston, "Review of 62 risk analysis methodologies of industrial plants," *Journal of Loss Prevention in the Process Industries*, vol. 15, no. 4, pp. 291–303, Jul. 2002, doi: 10.1016/S0950-4230(02)00008-6.

[220] P. K. Marhavilas, D. Koulouriotis, and V. Gemeni, "Risk analysis and assessment methodologies in the work sites: On a review, classification and comparative study of the scientific literature of the period 2000-2009," *Journal of Loss Prevention in the Process Industries*, vol. 24, no. 5. Elsevier, pp. 477–523, Sep. 01, 2011, doi: 10.1016/j.jlp.2011.03.004.

[221] A. Necci, V. Cozzani, G. Spadoni, and F. Khan, "Assessment of domino effect: State of the art and research Needs," *Reliability Engineering and System Safety*, vol. 143, pp. 3–18, Aug. 2015, doi: 10.1016/j.ress.2015.05.017.

[222] V. Villa, N. Paltrinieri, F. Khan, and V. Cozzani, "Towards dynamic risk analysis: A review of the risk assessment approach and its limitations in the chemical process industry," *Safety Science*, vol. 89. Elsevier B.V., pp. 77–93, Nov. 01, 2016, doi: 10.1016/j.ssci.2016.06.002.

[223] S. Kabir and Y. Papadopoulos, "Applications of Bayesian networks and Petri nets in safety, reliability, and risk assessments: A review," *Safety Science*, vol. 115. Elsevier B.V., pp. 154–175, Jun. 01, 2019, doi: 10.1016/j.ssci.2019.02.009.

[224] Y. Y. Haimes, "Risk modeling, assessment, and management," Accessed: Dec. 05, 2021. [Online]. Available: https://www.wiley.com/en-us/Risk+Modeling%2C+Assessment%2C+and+Management%2C+4th+Edition-p-9781119017981.

[225] H. Li, R. Sun, W. J. Lee, K. Dong, and R. Guo, “Assessing risk in chinese shale gas investments abroad: Modelling and policy recommendations,” *Sustainability (Switzerland)*, vol. 8, no. 8, 2016, doi: 10.3390/su8080708.

[226] N. Paltrinieri, L. Comfort, and G. Reniers, “Learning about risk: Machine learning for risk assessment,” *Safety Science*, vol. 118, pp. 475–486, Oct. 2019, doi: 10.1016/j.ssci.2019.06.001.

[227] J. Hegde and B. Rokseth, “Applications of machine learning methods for engineering risk assessment – A review,” *Safety Science*, vol. 122, no. September 2019, p. 104492, 2020, doi: 10.1016/j.ssci.2019.09.015.

[228] L. Tanguy, N. Tulechki, A. Urieli, E. Hermann, and C. Raynal, “Natural language processing for aviation safety reports: From classification to interactive analysis,” *Computers in Industry*, vol. 78, pp. 80–95, May 2016, doi: 10.1016/j.compind.2015.09.005.

[229] X. Xiang, C. Yu, and Q. Zhang, “On intelligent risk analysis and critical decision of underwater robotic vehicle,” *Ocean Engineering*, vol. 140, pp. 453–465, 2017, doi: https://doi.org/10.1016/j.oceaneng.2017.06.020.

[230] R. M. Ayo-Imoru and A. C. Cilliers, “Continuous machine learning for abnormality identification to aid condition-based maintenance in nuclear power plant,” *Annals of Nuclear Energy*, vol. 118, pp. 61–70, Aug. 2018, doi: 10.1016/j.anucene.2018.04.002.

[231] A. Mamudu, F. Khan, S. Zendehboudi, and S. Adedigba, “Dynamic risk assessment of reservoir production using data-driven probabilistic approach,” *Journal of Petroleum Science and Engineering*, vol. 184, p. 106486, Jan. 2020, doi: 10.1016/j.petrol.2019.106486.

[232] D. Zhou, X. Zhuang, H. Zuo, J. Cai, and H. Bao, “Hazard identification and prediction system for aircraft electrical system based on SRA and SVM:,” *https://doi.org/10.1177/0954410019894121*, vol. 234, no. 4, pp. 1014–1026, Jan. 2020, doi: 10.1177/0954410019894121.

[233] R. Alexander and T. Kelly, “Supporting systems of systems hazard analysis using multi-agent simulation,” *Safety Science*, vol. 51, no. 1, pp. 302–318, Jan. 2013, doi: 10.1016/j.ssci.2012.07.006.

[234] C. Y. Duan, X. Q. Chen, H. Shi, and H. C. Liu, “A New Model for Failure Mode and Effects Analysis Based on k-Means Clustering Within Hesitant Linguistic Environment,” *IEEE Transactions on Engineering Management*, pp. 1–11, 2019, doi: 10.1109/TEM.2019.2937579.

[235] Z. Wang and S. Li, “Data-driven risk assessment on urban pipeline network based on a cluster model,” *Reliability Engineering and System Safety*, vol. 196, p. 106781, Apr. 2020, doi: 10.1016/j.ress.2019.106781.

[236] H. Zhang *et al.*, “Novel naïve Bayes classification models for predicting the chemical Ames mutagenicity,” *Toxicology in Vitro*, vol. 41, pp. 56–63, Jun. 2017, doi: 10.1016/j.tiv.2017.02.016.

[237] N. Dogru and A. Subasi, “Traffic accident detection using random forest classifier,” *2018 15th Learning and Technology Conference, L and T 2018*, pp. 40–45, 2018, doi: 10.1109/LT.2018.8368509.

[238] S. S. Kamble, A. Gunasekaran, and S. A. Gawankar, “Sustainable Industry 4.0 framework: A systematic literature review identifying the current trends and future perspectives,” *Process Safety and Environmental Protection*, vol. 117, pp. 408–425, Jul. 2018, doi: 10.1016/J.PSEP.2018.05.009.

[239] J. Lee, I. Cameron, and M. Hassall, “Improving process safety: What roles for digitalization and industry 4.0?,” *Process Safety and Environmental Protection*, vol. 132, pp. 325–339, 2019, doi: 10.1016/j.psep.2019.10.021.

[240] A. Angelopoulos *et al.*, “Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects,” *Sensors (Switzerland)*, vol. 20, no. 1, pp. 1–34, 2020, doi: 10.3390/s20010109.

[241] C. Xu, S. Zhao, and F. Liu, “Sensor fault detection and diagnosis in the presence of outliers,” *Neurocomputing*, vol. 349, pp. 156–163, Jul. 2019, doi: 10.1016/J.NEUCOM.2019.01.025.

[242] M. T. Amin, S. Imtiaz, and F. Khan, “Process system fault detection and diagnosis using a hybrid technique,” *Chemical Engineering Science*, vol. 189, pp. 191–211, Nov. 2018, doi: 10.1016/j.ces.2018.05.045.

[243] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, “A review of process fault detection and diagnosis: Part III: Process history based methods,” *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 327–346, Mar. 2003, doi: 10.1016/S0098-1354(02)00162-X.

[244] Z. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques-part I: Fault diagnosis with model-based and signal-based approaches,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015, doi: 10.1109/TIE.2015.2417501.

[245] K. Zhang, K. Qian, Y. Chai, Y. Li, and J. Liu, “Research on fault diagnosis of Tennessee Eastman process based on KPCA and SVM,” *Proceedings - 2014 7th International Symposium on Computational Intelligence and Design, ISCID 2014*, vol. 1, pp. 490–495, 2015, doi: 10.1109/ISCID.2014.234.

[246] J. N. Goetz, A. Brenning, H. Petschko, and P. Leopold, “Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling,” *Computers & Geosciences*, vol. 81, pp. 1–11, Aug. 2015, doi: 10.1016/J.CAGEO.2015.04.007.

[247] T. Sorsa and H. N. Koivo, “Application of artificial neural networks in process fault diagnosis,” *Automatica*, vol. 29, no. 4, pp. 843–849, Jul. 1993, doi: 10.1016/0005-1098(93)90090-G.

[248] M. Ge, R. Du, G. Zhang, and Y. Xu, “Fault diagnosis using support vector machine with an application in sheet metal stamping operations,” *Mechanical Systems and Signal Processing*, vol. 18, no. 1, pp. 143–159, Jan. 2004, doi: 10.1016/S0888-3270(03)00071-2.

[249] L. B. JACK and A. K. NANDI, “FAULT DETECTION USING SUPPORT VECTOR MACHINES AND ARTIFICIAL NEURAL NETWORKS, AUGMENTED BY GENETIC ALGORITHMS,” *Mechanical Systems and Signal Processing*, vol. 16, no. 2–3, pp. 373–390, Mar. 2002, doi: 10.1006/MSSP.2001.1454.

[250] S. Mahadevan and S. L. Shah, “Fault detection and diagnosis in process data using one-class support vector machines,” *Journal of Process Control*, vol. 19, no. 10, pp. 1627–1639, Dec. 2009, doi: 10.1016/J.JPROCONT.2009.07.011.

[251] X. Gao and J. Hou, “An improved SVM integrated GS-PCA fault diagnosis approach of Tennessee Eastman process,” *Neurocomputing*, vol. 174, pp. 906–911, Jan. 2016, doi: 10.1016/J.NEUCOM.2015.10.018.

[252] B. Jin, D. Li, S. Srinivasan, S. K. Ng, K. Poolla, and A. Sangiovanni-Vincentelli, “Detecting and diagnosing incipient building faults using uncertainty information from deep neural networks,” *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019*, pp. 1–8, 2019, doi: 10.1109/ICPHM.2019.8819438.

[253] H. Zhao and Z. Lai, “Neighborhood preserving neural network for fault detection,” *Neural Networks*, vol. 109, pp. 6–18, Jan. 2019, doi: 10.1016/J.NEUNET.2018.09.010.

[254] H. Chen, P. Tiňo, and X. Yao, “Cognitive fault diagnosis in Tennessee Eastman Process using learning in the model space,” *Computers and Chemical Engineering*, vol. 67, pp. 33–42, 2014, doi: 10.1016/j.compchemeng.2014.03.015.

[255] V. A. Sotiris, P. W. Tse, and M. G. Pecht, “Anomaly Detection Through a Bayesian Support Vector Machine,” *IEEE TRANSACTIONS ON RELIABILITY*, vol. 59, no. 2, pp. 277–286, 2010.

[256] D. Fernández-Francos, D. Martínez-Rego, O. Fontenla-Romero, and A. Alonso-Betanzos, “Automatic bearing fault diagnosis based on one-class ν-SVM,” *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 357–365, Jan. 2013, doi: 10.1016/J.CIE.2012.10.013.

[257] K. Yan, Z. Ji, and W. Shen, “Online fault detection methods for chillers combining extended kalman filter and recursive one-class SVM,” *Neurocomputing*, vol. 228, pp. 205–212, Mar. 2017, doi: 10.1016/j.neucom.2016.09.076.

[258] R. Chalapathy, A. K. Menon, and S. Chawla, “Anomaly Detection using One-Class Neural

Networks," no. August, pp. 19–23, 2018, [Online]. Available: http://arxiv.org/abs/1802.06360.

[259] C. Alippi, I. Fellow, S. Ntalampiras, and M. Roveri, "for Distributed Sensor Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1213–1226, 2013, doi: 10.1109/TNNLS.2013.2253491.

[260] G. Karatzinis, Y. S. Boutalis, and Y. L. Karnavas, "Motor Fault Detection and Diagnosis Using Fuzzy Cognitive Networks with Functional Weights," *2018 26th Mediterranean Conference on Control and Automation (MED)*, pp. 709–714, 2018, doi: 10.1109/MED.2018.8443043.

[261] G. A. Papakostas, D. E. Koulouriotis, A. S. Polydoros, and V. D. Tourassis, "Towards Hebbian learning of Fuzzy Cognitive Maps in pattern classification problems," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10620–10629, Sep. 2012, doi: 10.1016/J.ESWA.2012.02.148.

[262] W. Froelich, "Towards improving the efficiency of the fuzzy cognitive map classifier," *Neurocomputing*, vol. 232, pp. 83–93, Apr. 2017, doi: 10.1016/J.NEUCOM.2016.11.059.

[263] J. J. Downs and E. F. Vogel, "A Plant-wide Industrial Problem Process," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993, doi: 10.1016/0098-1354(93)80018-I.

Chapter 3 Cognitive Fault Detection and Diagnosis Using Artificial Intelligent

Preface

A version of this manuscript has been published in the Computer and chemical engineering Journal. In this chapter, a cognitive fault detection and diagnosis methodology has been developed using unsupervised process data. Anomaly detection and classification machine learning algorithms were used in the methodology. This chapter contributes to the targeted 2nd objective of the thesis defined in chapter 1.

Abstract

Fault detection and classifications using supervised learning algorithms are widely studied; however, lesser attention is given to fault detection using unsupervised learning. This work focused on the integration of unsupervised learning with cognitive modeling to detect and diagnose unknown fault conditions. It is achieved by integrating two techniques: (i) incremental one class algorithm to identify anomaly conditions and introduce a new state of fault to the current fault states if an unknown fault occurs, and (ii) dynamic shallow neural network to learn and classify the fault state. The proposed framework is applied to the well-known Tennessee Eastman process and achieved significantly better results compared to results reported by in literature. Laboratory experiments are also performed using a pilot-scale system to test the validity of the approach. The results confirm the proposed framework is an effective way to detect and classify known and unknown faults in process operations.

3.1 Introduction

Safety and reliability are essential for chemical process industries. Ensuring safety requires continuous monitoring of an operation and efficient detection of abnormalities. Online monitoring and fault detection and diagnosis are becoming more and more attractive research and application areas [1]–[3].

Model-based and data-driven methods are commonly used approaches to detect and diagnose faults [2]. The model-based approaches heavily depend on the mathematical model of the system and its components. To detect fault using the model-based approach, the developed mathematical model predicts the model output and compares it with plant data. The resulting residual is used to detect fault conditions.

The model-based approaches are limited to linear and some specific nonlinear models, and the model-based fault detection methods used filter-based models such as the Kalman filter (KF), extended KF, and particle filters, which are effective tools for fault detection [2], [4]. However, these methods assume prior knowledge of measurement noise statistics. Therefore, when outliers exist in measurement, the KF based fault detection approaches cannot distinguish between fault and outlier. Apart from fault detection, fault diagnosis is quite a challenge for the model-based approaches. The model-based fault diagnosis is categorized as deterministic systems, stochastic systems, discrete-event, and hybrid systems, as well as network and distributed diagnosis systems [5]. A nonlinear stochastic dynamic process system can develop combined with particle filters and interactive multiple model estimation [6]. The proposed method detects and provides fault information accurately for known faults. For a complex system defining the mathematical models and defining fault conditions beforehand are challenging [6].

The data-driven approach requires good quality historical data from the system. The data-driven fault detection and diagnosis are categorized as traditional statistical methods and novel machine learning approaches [7]. Principle component analysis (PCA) and statistical pattern classifiers are commonly used statistical approaches. Among the machine learning approaches, the neural network (NN) and support vector machines (SVM) are more commonly used.

However, most of the data-driven research efforts were dedicated to the supervised learning methods, where the data-driven model has enough information about all the different types of faults. The supervised learning methods heavily rely on prior knowledge of the fault type, several functional forms, and fault conditions. Due to the system's complexity, process data features exhibit nonlinear behavior. Many researchers currently follow different machine learning approaches to develop a model based on prior knowledge gained from nonlinear data. The nonlinear data-driven approach can use the neural network with a multilayer perceptron and a hyperbolic tangent as the nonlinear element to detect and diagnose faults [8]. The SVM fault diagnosis method suggested a better success rate compared to the neural network in the case of a test application [9]. NN and SVM fault detection methodologies were critically reviewed by Jack and Nandi (2002), and improvement of the overall generalization performance was demonstrated using a genetic algorithm [10]. The one-class support vector machine is also proposed for fault detection and diagnosis in a processing plant [11]. Gao et al. (2016) suggested PCA integrated (to reduce the feature dimension) multiclass SVM coupled with the grid search method (for improved optimization) based process supervision and fault diagnosis [12]. Yin and Hou (2016) reviewed the research and development of fault diagnosis and monitoring approach based on SVM and showed the advantages in generalization performance and for small samples. The Monte Carlo dropout can also be used to enhance the supervised learning pipeline; the resulting neural network can detect and diagnose unseen initial fault states [13]. Zhao and Lai (2019) proposed a novel statistical feature extraction method, namely, the neighborhood preserving neural network for nonlinear data-driven fault detection techniques through preserving the local geometric structure of normal process data [14].

More recently, hybrid approaches are being commonly used to detect and diagnose faults. Hybrid approaches are generally a combination of model-based approaches and data-driven approaches. Amin et al. proposed a hybrid approach to integrate the PCA and T^2 statistics with a Bayesian Network (BN) model [3]. Galagedarage and Khan proposed a combination of the Hidden Markov Model (HMM) and BN, where the HMM detects the abnormalities based on historical data and the BN diagnoses the fault and finds the root cause [1]. In fault detection and diagnosis, hybrid approaches are mainly used to find the root cause.

The developed methodology used in this section is able to detect the unknown fault states by using an incremental one-class neural network and classify the known fault by using a shallow neural network. A similar approach was investigated by Chen et al. using the concept of cognitive fault diagnosis by using a reservoir neural network and one class support vector machine (OC SVM) without the prior knowledge of fault states and signatures [15].

The main contributions of the present work include: i) development of an incremental one class NN algorithm to detect unknown faults, ii) use of a shallow neural network to learn the known and unknown faults and develop a model to classify the different fault conditions, and iii) testing the effectiveness of the proposed algorithm and model using a fault-finding experimental setup (RT 580) and the benchmark Tennessee Eastman process.

3.2 Background and Related Work

Most fault detection algorithms are based on mathematical models, where the dynamic behavior of a system is modeled using mathematical equations and compared with the real system behavior.

However, for a complex process system, such a model may not be accurate or available. When the mathematical models are more complex or unavailable, data-driven approaches are used. These approaches highly depend on data, including data for all different fault conditions. However, in a complex process system or a newly developed process, getting all possible fault states' data remains a challenge.

Recent research on fault detection and diagnosis mainly focuses on detecting unknown faults by using very limited knowledge or data. OCSVM is used to detect the anomalies and integrated the output to calibrate the posterior probabilities to manage the false alarms [16]. OCSVM also used to discriminate between a normal and a faulty condition for automated bearing fault detection and diagnosis [17]. Yin et al. applied the robust one-class SVM to detect the fault conditions using only the normal data for training [18]. A novel hybrid method was proposed by Yan et al. to detect faults and perform online classification without any faulty training data [19]. They developed a framework using the extended Kalman filter model and the recursive SVM. One class neural network (OC-NN) proposed by Chalapathy is similar to the OCSVM, OC- NN combines the ability of deep networks to extract a progressively rich representation of data along with the one class objective [20].

3.2.1 Cognitive Fault Detection and Diagnosis

Cognitive fault detection and diagnosis is a forthcoming research area in many engineering disciplines. Alippi et al. proposed a Hidden Markov model-based cognitive fault detection and diagnosis methods for distributed sensor networks [21]. The fuzzy cognitive network (FCN) based fault detection approach was introduced by Karatzinis et al.; FCN is an operational extension of a Fuzzy Cognitive Map (FCM). FCM has been used by various researchers for pattern recognition and decision making [22]. Papakostas et al. introduced the FCM for the pattern classification problem, and Frolich proposed an improved FCM based classifier [23], [24].

However, the most recent research on cognitive fault classifications is based on supervised learning. Chen *et al.* proposed a memory-based approach to detecting the unknown faults based on learning on the model space and one-class SVM for fault detection [15].

This work introduces the unsupervised NN approach for unknown fault detection in algorithm 1 and general shallow NN for known fault classification by using supervised learning in algorithm 2. Section 3.2.2 describes the one-class NN method for unknown fault prediction, and section 3.2.3 describes the shallow NN-based classification for known fault classification.

3.2.2 One Class Neural Network

OC NN is similar to the OCSVM that separates all the data points from the origin. OC- NN can exploit and improve features obtained from unsupervised learning specifically for anomaly detection. In this approach, the model is not trained with predicted output. Unlike supervisor learning, this model is trained and tested using process data. OC- NN evaluates data samples based on decision scores to differentiate anomalies in complex data sets where the decision boundary between normal and anomalies is highly nonlinear.

Chalapathy et al. proposed a shallow neural network with one output model to detect anomalies [20]. The suggested model is shown in Figure 3.1.

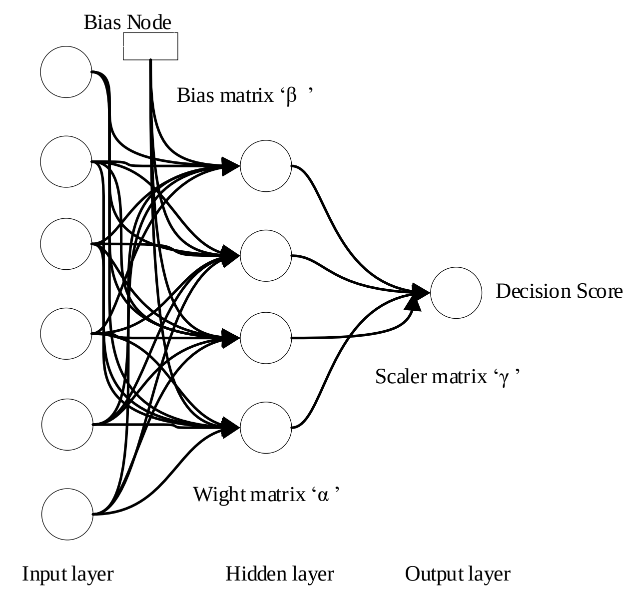


Figure 3.1 One Class Neural Network Model

In the above neural network, ‘γ’ is the scalar matrix from the hidden to the output layer, ‘α’ is the weight matrix from the input to the hidden layer, F is the feature space or the variables that are used in the NN input nodes, ‘β’ is the bias matrix, g is the NN activation function, and N is the number of samples from the origin.

The OC-NN objective is formulated as,

$$\min_{\gamma,\alpha,\beta} \frac{1}{2}\|\gamma\|_2^2 + \frac{1}{2}\|\alpha\|_F^2 + \frac{1}{\alpha}.\frac{1}{N}[\textstyle\sum_{n=1}^{N} \max(0, \beta - \langle\gamma, g(\alpha \boldsymbol{X}_{\boldsymbol{n}:})\rangle) - \beta] \tag{3-1}$$

From Equation 3.1, bias ‘β,’ weight matrix ‘α,’ and the output matrix ‘γ’ can be optimized by a minimization approach. Chalapathy et al. described the optimization algorithm for OC- NN by optimizing ‘α’ and ‘γ’ separately while updating the ‘β’ based on optimized ‘α’ and ‘γ’ [20].

$$\min_{\gamma,\alpha} \frac{1}{2}\|\gamma\|_2^2 + \frac{1}{2}\|\alpha\|_F^2 + \frac{1}{\alpha}.\frac{1}{N}[\textstyle\sum_{n=1}^{N} \max\,(0, (\,\beta - [\gamma, g(\alpha \boldsymbol{x}_{\boldsymbol{n}:})])] \tag{3-2}$$

Similarly, the optimization for β,

$$\underset{\beta}{argmin}\left[\frac{1}{N\alpha} \cdot \sum_{n=1}^{N} \max\left(0, \left(\beta - [< \gamma, g(\alpha \boldsymbol{x}_{\boldsymbol{n}:}) >]\right) - \beta\right]. \quad (3\text{-}3)$$

In OC- NN algorithm, the input layer feeds the set of multivariate samples and calculates the data sample decision score by a single output for a single data sample [20]. Equation 3.2 shows the optimization of 'α' and 'γ' using the 'β' value. Similarly, Equation 3.3 shows the optimization of 'β' by using the optimized 'α' and 'γ' values. Initial decision scores are managed to compute for each data sample based on the optimized γ, α, and β values. By using the decision score, normal and abnormal data samples are classified. Finally, with the optimized model, positive decision score data will identify as a normal condition, and a negative decision score will be defined as an abnormal condition.

In this section, an incremental OC-NN algorithm is developed using a dynamically changing NN model. For example, all the data samples with a negative decision score are counted as the number of anomalies in the data frame and compared with the marginal level to detect abnormal conditions. Once an abnormality is found by the OC- NN, the model identifies the newly detected fault condition. Detected fault features are identified by operational knowledge, and incremental OC-NN model learned and trained itself to detect the unknown faults. Also, the classification NN updates itself for the classification by including the newly detected fault. The complete algorithm is detailed in section 3.3.

3.2.3 Fault Classification

In this work, shallow NN is used to classify the faults states based on a supervised learning approach. NN model output dynamically changes based on detected unknown fault conditions. Therefore, the developed model learns and tests itself according to the detected faults by algorithm 1. For example, if an unknown fault is detected by the OC-NN model, the shallow NN output is augmented by 1. Detected unknown faults are labeled as a new fault condition and are updated by the NN. Figure 3.2 shows the proposed NN model for classification.

If a known fault occurs, identified fault states are classified based on the trained model.

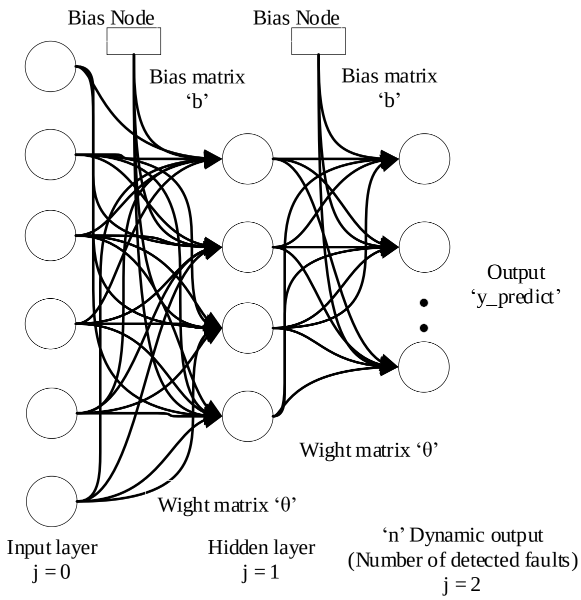


Figure 3.2 Dynamic output Neural Network.

The dynamic output neural network objectives are formulated as,

$$a_i^j = g(x_i.\theta_i^j + b^j) \quad (3\text{-}4)$$

Where, $'a_i^j'$ is an activation of node $'\mathrm{i}'$ in layer 'j', $'\theta_i^j'$ is the weight matrix from layer 'j' to 'j+1', b^j is the bias matrix in each layer. Initial data samples are defined as (x_i) in the input layer, and for hidden and output layers (j – 1), node value is defined as $'x_i'$.

In the proposed model, a neural network with backpropagation and sigmoid activation function is used to optimize the weight matrix $'\theta'$ and bias $'\mathrm{b}'$. For a dynamic output classification model, a shallow NN output label is updated with identified new fault conditions. Both proposed NNs are fully connected and receive information from each constructed node. The complete algorithm is presented in section 3.3.

3.3 Cognitive Fault Diagnosis Methodology

This section describes the complete framework of the proposed model. NN-based anomaly detection is used in this section to detect the anomalies from the origins. The OC-NN modified as incremental OC-NN is used to detect the unknown faults from existing faults and non-fault conditions. To classify the captured faults, a dynamic output NN is modified and developed based on a shallow neural network and backpropagation.

The complete proposed framework is shown in Figure 3.3. Initially, OC- NN model and shallow classification NN models are trained by using nonfaulty data. Time-dependent data samples are framed by moving ‘n’ sampling window and tested for the anomaly. In the tested data frame, all the anomaly samples are counted, and the total number is compared with a predefined marginal level. From the experiment, the marginal level varies with the noise level in the data samples. If trained and tested data has a 5% noise level, expected results are obtained when 20% anomalies are considered as the marginal level. However, with a 10% noise level in training and test data, 25% of anomalies provide expected results within a data frame.

If the number of data points marked as anomaly exceeds the marginal level, the tested data frame is detected as having an unknown fault, and the OC- NN model is updated by using the detected unknown fault data frame and previously trained data frame for unknown fault detection. At the same time, the shallow NN output node is incremented and labeled with a new fault condition for further fault classification.

In the incremental OC- NN test, if counted anomaly detection points remain within the marginal level, the data frame is identified as having a known fault condition. Therefore, the data frame directly feeds into the shallow NN to classify the fault condition.

The proposed incremental OC-NN structure is shown in Figure 3.1. The NN model is developed based on the features in data samples as an input layer, 50 hidden layers, and one output layer.

The proposed shallow classification NN structure is shown in Figure 3.2. The NN is initially modeled using the data features as the input layer, 60 hidden layers, and a single output layer to classify the no-fault condition. However, the model learns itself and dynamically changes its output based on the number of detected faults.

The sigmoid neural network activation function is used in both algorithms.

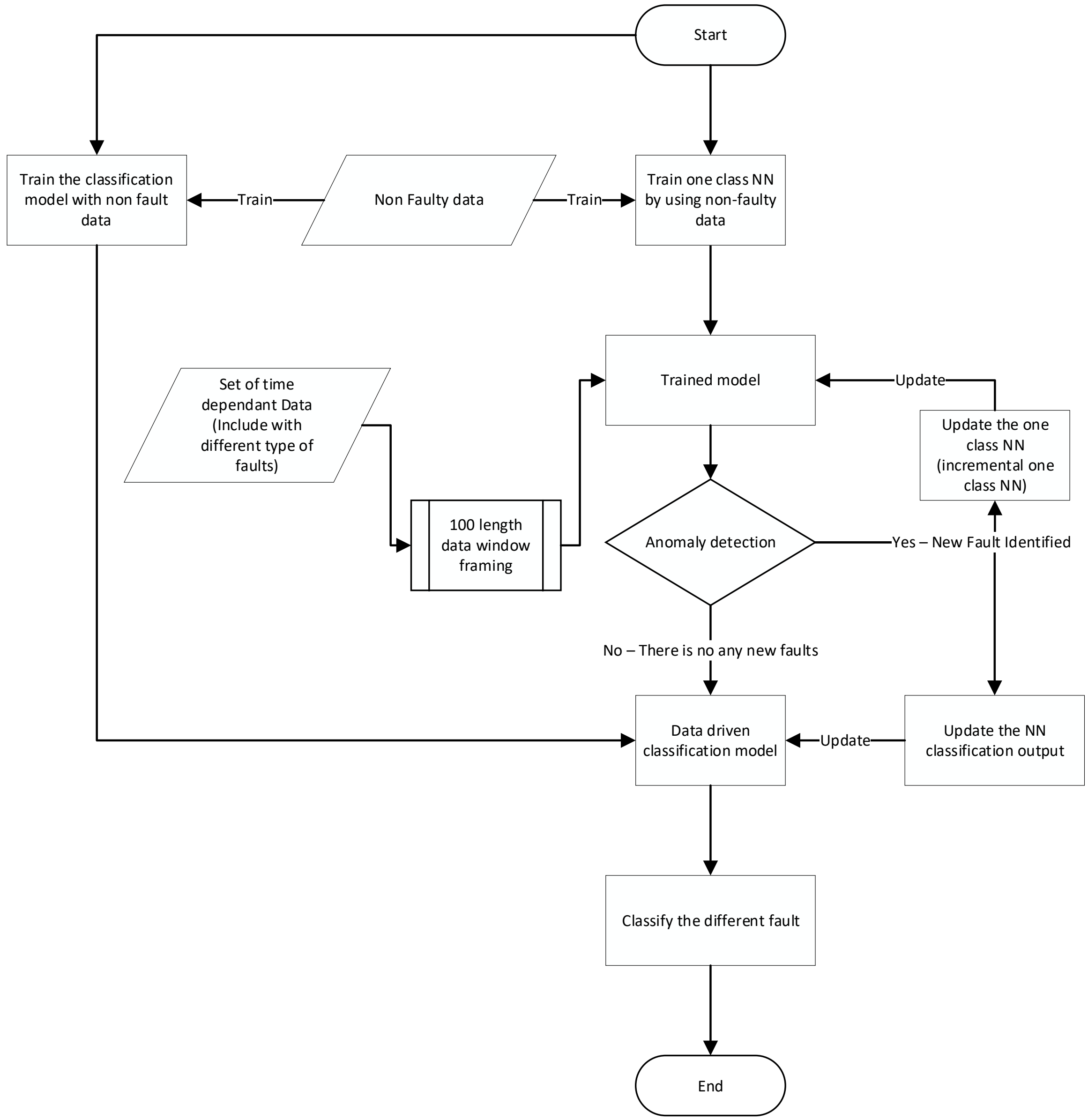


Figure 3.3 The Framework to develop cognitive fault diagnosis model

Incremental OC-NN parameters γ, α, and β are optimized using the method proposed by Chalapathy [20] and are detailed in section 3.2. An incremental one-class algorithm is summarized in algorithm 1. The number of iterations was defined based on a trial-and-error method. From experiment 500 to 1000, iterations gave an acceptable range; this may vary based on a different type of data.

Shallow neural network parameters θ and b values are optimized using backpropagation and Adam optimizer. Algorithm 2 summarizes the dynamic output shallow NN. The number of optimizing epochs was defined based on a trial-and-error method. Approximately 300 to 700 iterations gave a better-optimized result from the experiment. However, this approximation is more suited to the selected data, and it may vary with the different data samples and features.

Algorithm 1 Incremental OC- NN

```
1   Input    Nonfaulty data, 'n' sample window time-dependent data.
2   Output   Set of decision scores (for one data frame) / update OC- NN
3   while    (1: 'n' sample window) :
4            Define 'β' value
5            for    (1: No of epochs)
6                   Update ('α' and 'γ'); optimize the NN weights and output layer
7                   Optimize the r-value using updated 'α' and 'γ'.
8            end
9            Compute the decision score for each sample: d(n)
10           if     decision score (d(n) ≥ 0) then
11                  Normal data point
12           else
13                  Anomaly_count = Anomaly_count + 1
14  end
15           After 'n' samples
16           if     Anomaly_count > Marginal level
17                  Detect fault and run 2nd 'n' window using 3 – 13
18                  if          Anomaly_count > Marginal level
19                              Update the incremental OC- models
20                              Update dynamic output NN layer
21                  else        Define the unknown fault as false alarm
22           else   Jump to Dynamic NN, for fault classification
23  Repeat step 3 for next windowed samples.
```

If the incremental OC-NN identifies new faults, the shallow classification NN will update the output layer and train the model with the newly detected fault by running lines 6 to 10 in algorithm 2 to classify the same fault in the future. Initially, no-fault data is used to train the model with one output node. Moving forward, a sliding ‘n’ window in the data set will detect the new/unknown faults. Whenever the unknown fault is detected, the sampling window is used to train the shallow NN.

Algorithm 2 Dynamic output Shallow NN.

```
1  Input:    Nonfaulty data or ‘n’ window time-dependent data.
2  Output:   Dynamic output NN
3            Initialize the θ and b
4            Define the input layer (number of features) and the hidden layer
5            Update output layer (based on incremental OC- NN)
6            for (1: No of epochs)
7                   | Optimize θ and b
8            end
9  return {Dynamic output NN}
```

3.3.1 Experimental Test of the Developed Algorithm

The developed algorithm is tested on a laboratory-scale pilot plant. Armfield RT 580 Fault finding system liquid level and flow rate cascading experiment have been done to collect no-fault and fault data. The details of the experimental setup are presented in the RT580 experimental manual. In this experimental process, the tank level and the tank input flow rate variables are used to monitor the fault condition.

The experimental data plot is shown in Figure 3.4. Fault 1 was introduced between 500 – 900 samples, and fault 2 was introduced between 1200 – 1500 samples. According to the experimental setup, fault 1 defines the deviation in the process liquid level in the tank, and fault 2 defines the deviation in the flow rate.

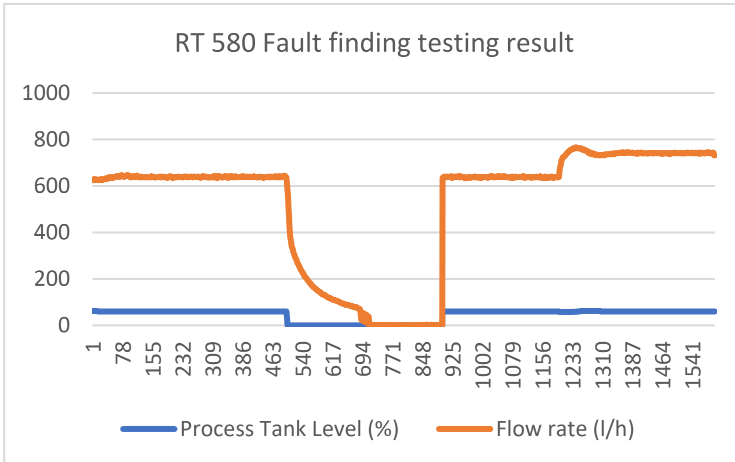


Figure 3.4 RT 580 Experimental Result.

The complete test process is shown in Figure 3.5. In this experiment, 1500 data samples with two feature data (process tank level and flow rate) are used to test the algorithm. An incremental OC-algorithm defined in algorithm 1 was used to detect the anomaly sample points in the data frame. Initially, nonfaulty 300 data samples were fed into the algorithms to train the model. By training the models, the incremental OC- NN model parameters γ, α, and β were optimized using algorithm 1, and the shallow NN model parameters θ, and b were optimized using algorithm 2.

After 300 samples, data were divided into 100 sample frames and fed into the trained model to detect anomaly points. In the data frame, if the number of anomalies is less than the marginal level, the model defines the tested data frame as a known fault state and sends the data frame to algorithm 2 for classification.

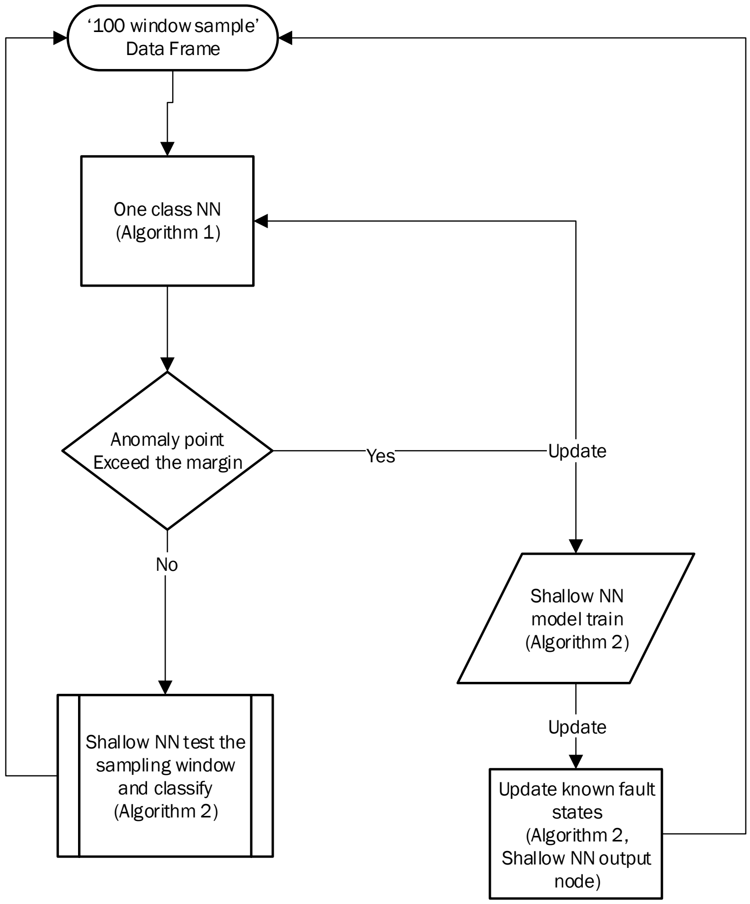


Figure 3.5 Experimental test flow diagram.

If the number of anomaly points is larger than the marginal level, incremental OC-NN and shallow NN would get updated based on the newly detected unknown faulty data frame. In this experiment, two data frames were used to confirm the unknown fault condition before updating the NN models. Therefore 200 samples were used in the testing to update the models. Table 3.1 shows the training dataset used in algorithm 1.

Table 3-1 Incremental NN model and dataset for RT 580 experiment.

Incremental Model	**Data Set for algorithm 2**
Initial	300 non-Faulty data
Model 1	Initial + 200 F1[2]
Model 2	Model 1 + 200 F2[2]

[2] Dataset identified by the one class NN for unknown faults.

Table 3.2 shows the RT 580 experimental data fault detection test result based on the developed algorithm.

Table 3-2 Test fault and classification model update.

Fault	Fault ID	Introduced fault sample	Fault detection	Classification Model update (for algorithm 2)
Fault 1	1	500 - 900 sample	Between 500 – 600 samples	Model updated with 500 – 700 samples
Fault 2	2	1200 - 1500 sample	Between 1200 – 1300 samples	Model updated with 1200 – 1400 samples

The cognitive fault detection and classification results are shown in Table 3.3. From the developed algorithms, faults can be isolated and classified as a different fault condition. However, different fault characteristics initially should be defined by human knowledge with data. In this experiment, fault id 1 defined as a fault 1 at 500 samples. With experience in data at the 500^{th} sample fault identified in the process tank level. Same as fault id 2 defined as a fault 2 at 1200 samples. With experience in the data, 1200^{th} sample's fault identified in the flow rate level.

Table 3-3 Experimental result with classification and model develop.

Sample window	Results	Sample window	Results
0-2	Trained the algorithm 1 and 2	8	98 % fault 1 classification 2% Nonfaulty classification
3-4	100% Nonfaulty classification	9 – 11	97% Nonfaulty classification
5	Unknown fault detected and algorithm 1 updated.	12	Unknown fault detected and algorithm 1 updated.
5-6	Label as fault 1 and trained algorithm 2.	12-13	Label as fault 2 and trained algorithm 2.
7	100% fault 1 classified	14-15	97% fault 2 classified

3.4 Application of the Proposed Models and Algorithm

This section describes the use of the developed model for a complex industrial problem. The Tennessee Eastman challenge process was used for this study. Fault detection and diagnosis performance were investigated by using the developed model, and the results were compared with other recently developed algorithms.

Down and Vogel proposed the TE process that provides realistic industrial process control and monitoring model [25]. The process model shown in Figure 3.6 consists of five major units: reactor, condenser, compressor separator, and stripper. TE process contains 41 measured variables and 12 manipulated variables. Based on the real chemical process, the TE produces two products from four reactants. It contains eight different feed components labeled (A-H). The gaseous reactants A, C, D, and E, and the inert B are fed to the reactor where the liquid products G and H are formed. The species F is a by-product of the process.

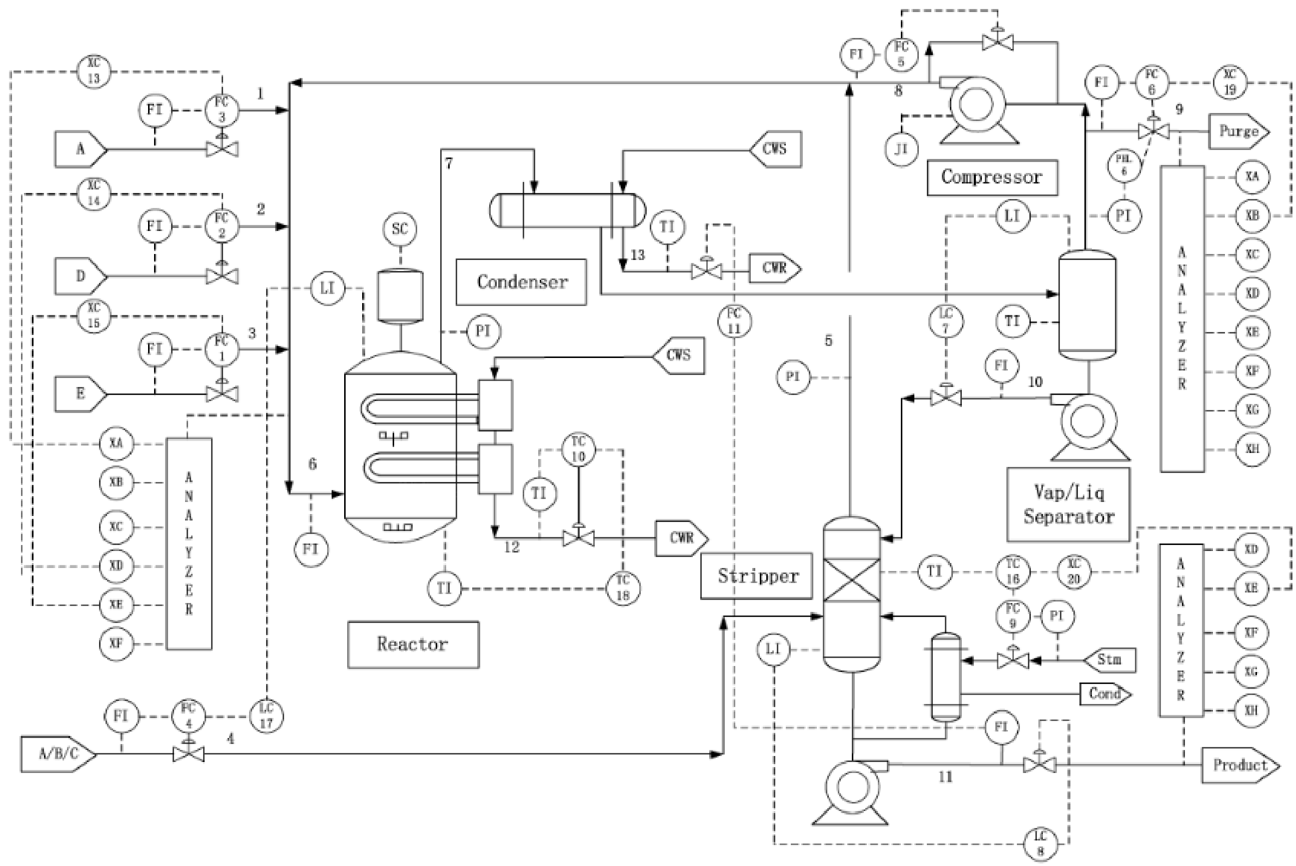


Figure 3.6 TE process flow [25]

Out of the 41 measured input variables, 22 variables are (XMEAS1 to XMEAS22) continue process measurements, and the other 19 variables (XMEAS 23 to XMEAS 41) are composition measurements. There are 21 process faults in the TE process, as summarized in Table 3.4.

Out of the 21 faults in the TE process, IDV4, IDV9, and IDV11 are a good representation of overlapping information, and also, they are difficult to classify [6]. Therefore, to test the proposed algorithm, TE process faults IDV 1, 4, 5, 6, and 11 have been selected to perform the experiment.

MATLAB Simulink was used to simulate the nonfaulty and faulty data with a specified time frame. And Python tensor flow library was used to implement the proposed algorithm.

Table 3-4 Tennessee Eastman process faults. [25]

Variable	Description	Type
IDV1	A/C feed ratio, B composition constant (Stream 4)[3]	Step
IDV2	B composition, A/C ratio constant (Stream 4)	Step
IDV3	D feed temperature (Stream 2)	Step
IDV4	Reactor cooling water inlet temperature[3]	Step
IDV5	Condenser cooling water inlet temperature[3]	Step
IDV6	A feed loss (Stream 1) [3]	Step
IDV7	C header pressure loss - reduced availability (Stream 4) [3]	Step
IDV8	A, B, C feed composition (Stream 4)	Random variation
IDV9	D feed temperature (Stream 2)	Random variation
IDV10	C feed temperature (Stream 4)	Random variation
IDV11	Reactor cooling water inlet temperature	Random variation
IDV12	Condenser cooling water inlet temperature	Random variation
IDV13	Reaction kinetics	Slow drift
IDV14	Reactor cooling water valve	Sticking
IDV15	Condenser cooling water valve	Sticking
IDV16	Unknown	
IDV17	Unknown	
IDV18	Unknown	
IDV19	Unknown	
IDV20	Unknown	
IDV21	The valve for Stream 4 was fixed at the steady-state position	Constant position

3.5 Results and Discussion

In this chapter, the proposed method is tested using the TE data. 3300 Data samples with 22 measurement variables/data features are used for testing. Out of 20 different fault types, five

[3] Fault conditions that are tested by the algorithm.

selected faults were used to test the developed algorithm. IDV 4, 11 are especially focused due to the data overlapping along with faults, IDV 1, IDV 5, and IDV 6.

One thousand nonfaulty data samples and 500 data samples in IDV 1 and IDV 5 faults, and 400 data samples in IDV 6 and IDV 11 faults were generated as time-series data to test the proposed models.

3.5.1 Incremental One-Class NN Test Results

The proposed incremental OC-NN model was initially trained by using 500 no-fault data samples. The rest of the time series data samples are fed into the moving 100 sample windows and divided as data frames. Each frame is tested by an optimized NN model to generate the decision scores for each sample separately. Based on the decision score, data samples in the frame are distinguished as normal or abnormal data points. Each abnormal point within a data frame is counted, and the total number is compared with an acceptable, marginal level. If the counted anomaly points exceed the marginal level, the data frame is detected as having an unknown fault condition, and the NN model is updated for further detection. In the incremental one-class NN model, known and unknown fault are detected without any label or prior knowledge in each data frame. Therefore, the unsupervised learning approach proposed in the model is tested based on each sample's decision scores.

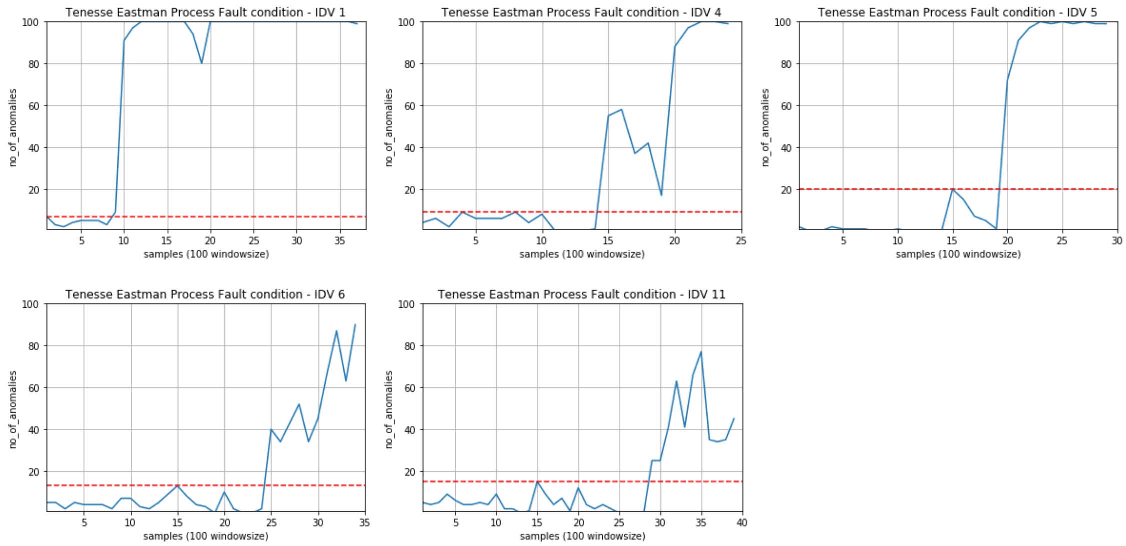


Figure 3.7 Incremental OC- NN results

Incremental OC-NN test results are shown in Figure 3.7. Without any further knowledge, the first unknown fault (IDV 1) was detected at the 10th sampling window. Once the unknown fault is detected, the model is updated with 500 no-fault data samples and 200 detected fault data samples to detect unknown faults in the future. The complete model updating framework is shown in Table 3.5.

Table 3-5 Incremental NN model and dataset.

Incremental Model	**Data Set**
Initial	500 NF
Model 1	Initial + 200 F1[4]
Model 2	Model 1 + 200 F2[4]
Model 3	Model 2 + 200 F3[4]
Model 4	Model 3 + 200 F4[4]

[4] Data frame identified by the incremental one class NN for unknown faults

3.5.2 Cognitive Fault Detection and Diagnostic Test Results

To test the developed cognitive fault detection and diagnosis method, generated TE Time-dependent data samples are used.

Table 3-6 Test fault detection and model update.

Fault	Fault ID	Introduced fault sample	Fault detection	Models update
IDV 1	1	1000 sample	Between 1000 – 1100 samples	Models updated with 1000 – 1200 samples
IDV 4	2	1500 sample	Between 1500 – 1600 samples	Models updated with 1500 – 1700 samples
IDV 5	3	2000 sample	Between 2000 – 2100 samples	Models updated with 2000 – 2200 samples
IDV 6	4	2500 sample	Between 2500 – 2600 samples	Models updated with 2500 – 2700 samples
IDV 11	5	2900 sample	Between 2900 – 3000 samples	Models updated with 2900 – 3100 samples

In the cognitive approach, the initial incremental OC-NN and the shallow neural network models were trained by 500 no-fault data samples. The rest of the 2800 samples were divided into moving 100 samples window data frames. Each frame was fed into the incremental OC-NN model to detect the unknown fault conditions. If incremental OC- NN detects the unknown fault condition, both models will get trained and updated with the detected unknown fault data, and the process is repeated. Table 3.6 shows the incremental OC-NN fault detection and both model updates time frame for the generated 3300 data samples. With the incremental one-class NN testing, if the data frame is identified as having a known fault condition, the data frame is fed into the shallow NN and classified as a detected known fault state. In this approach, fault detection is achieved by the unsupervised OC-NN model, and fault classification is achieved by the supervised shallow NN.

Table 3.7 shows the incremental OC-NN and the shallow NN model updates and results. In this experiment, we tested the cognitive approach to detect the unknown fault condition without prior knowledge. Further experimental results prove that all the updates and results will be generated based on the unknown fault detection and classification by the developed algorithms. In a way, the developed algorithms were updated based on the newly detected unknown fault conditions, and the classification was continued for known fault states until the data frame ended.

Table 3-7 Data frame moving samples and obtained result.

Sample window	**Model update and Results**
0-5	Trained algorithm 1 and 2
6 - 9	100% Nonfaulty classification
10	Unknown fault detected and algorithm 1 updated.
10 - 12	Label as fault 1 and trained algorithm 2.
12 - 14	99% fault 1 classified
15	Unknown fault detected and algorithm 1 updated.
15 - 17	Label as fault 2 and trained algorithm 2.
17 – 19	100 % classified as fault 2.
20	Unknown fault detected and algorithm 1 updated
20 -22	Label as fault 3 and trained algorithm 2.
22 – 24	100 % classified as fault 3.
25	Unknown fault detected and algorithm 1 updated.
25 – 27	Label as fault 4 and trained algorithm 2.
27 – 28	99% classified as fault 4.
29	Unknown fault detected and algorithm 1 updated
29 – 31	Label as fault 5 and trained algorithm 2.
31 – 33	99% classified as fault 5.

3.5.3 Comparison of the Proposed Models with Recent Studies

The proposed model effectively detects the fault within an expected window frame. In this approach, we divided the dataset into data frames and fed the frames into the NN models. Therefore, the NN model response will get faster than commonly used NN models. In real-time, smaller sampling intervals and a high-performance computer will give a better result in terms of detection time. This study used 100 sample windows running with intel core i5 with 4 GB RAM, and took 80-sec running time for every 100 samples.

Tables 3.8 and 3.9 demonstrate the strength of the developed methodology. Unknown faults are detected within the expected time frame. However, due to the data sampling frame, a minimal delay will there be for detection.

Table 3-8 Performance comparison of the proposed system (Samples).

Fault	**Fault ID**	**Incremental OC- NN model (maximum #[5] of samples)**	**HMM model (# of samples) (Galagedarage et al. (2019))**
IDV 1	1	100	**92**
IDV 4	2	**100**	325
IDV 5	3	100	**90**
IDV 6	4	**100**	241
IDV 11	5	**100**	335

Almost 100% accuracy is shown in the NN classification model with 200 samples of training data.

[5] # - Number of

Table 3-9 Classification ratio by using Shallow Neural Network (%).

Fault	Fault ID	Classification accuracy (%)		
		Dynamic output NN model (Proposed Model)	**SVM Jing et al. (2014)**	**PCA**
IDV 1	1	**99**	93	97
IDV 4	2	**100**	65	92
IDV 5	3	**100**	73	98
IDV 6	4	**99**	77	99
IDV 11	5	**99**	30	83

The main contribution and usage of the proposed model are as follow. This work is a straightforward method to apply machine learning techniques to detect and classify fault conditions and update the model over time. The main goal of this method is to learn and update the model without prior knowledge, which will be engaged to learn the fault over time and update the model itself. From the experimental result and case study, it is shown that the developed algorithm is capable of detecting the unknown fault within the time frame and classifying the known fault with high accuracy.

The proposed approach is flexible to accommodate different sampling times for time-dependent data. For example, for a corrosion experiment, larger time interval (Week or Month), or for a process industry, very short time interval (millisecond or second) data can be fed into the model in real-time to distinguish the normal and abnormal behavior as well as to classify faults.

Further developing this method to find the causalities or fault features for the variable deviation would be beneficial to develop a state of solution for fault detection and diagnosis using a comprehensive machine learning concept.

3.6 Conclusions

This section introduces a cognitive approach to fault detection and diagnosis. The proposed model is able to update itself based on the detected unknown fault conditions and keep on updating further whenever an unknown fault is detected. This framework is proposed based on the incremental one-class NN algorithm and the shallow NN with dynamically changing output nodes. An unsupervised learning approach is used by the incremental one-class NN to detect the fault condition without any prior knowledge, and the shallow NN uses a supervised learning approach to classify the known fault conditions.

As the main objective, this paper investigates the fault without prior knowledge of the fault numbers and type. Laboratory experiments and simulated data are used to test the developed method. The results confirm the benefits of the algorithm in cognitive fault detection. This framework is recommended for data with limited knowledge of fault. This framework can be extended to find the root cause of each identified fault. Also, with the appropriate numerical/mathematical model of the system, this framework can be expanded to distinguish between uncertainty and actual fault.

References

[1] M. Galagedarage Don and F. Khan, "Process fault prognosis using hidden markov model-bayesian networks hybrid model," *Industrial and Engineering Chemistry Research*, vol. 58, no. 27, pp. 12041–12053, 2019, doi: 10.1021/acs.iecr.9b00524.

[2] C. Xu, S. Zhao, and F. Liu, "Sensor fault detection and diagnosis in the presence of outliers," *Neurocomputing*, vol. 349, pp. 156–163, Jul. 2019, doi: 10.1016/J.NEUCOM.2019.01.025.

[3] M. T. Amin, S. Imtiaz, and F. Khan, "Process system fault detection and diagnosis using a hybrid technique," *Chemical Engineering Science*, vol. 189, pp. 191–211, Nov. 2018, doi:

10.1016/j.ces.2018.05.045.

[4] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, “A review of process fault detection and diagnosis: Part III: Process history based methods,” *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 327–346, Mar. 2003, doi: 10.1016/S0098-1354(02)00162-X.

[5] Z. Gao, C. Cecati, and S. X. Ding, “A survey of fault diagnosis and fault-tolerant techniques-part I: Fault diagnosis with model-based and signal-based approaches,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015, doi: 10.1109/TIE.2015.2417501.

[6] K. Zhang, K. Qian, Y. Chai, Y. Li, and J. Liu, “Research on fault diagnosis of Tennessee Eastman process based on KPCA and SVM,” *Proceedings - 2014 7th International Symposium on Computational Intelligence and Design, ISCID 2014*, vol. 1, pp. 490–495, 2015, doi: 10.1109/ISCID.2014.234.

[7] J. N. Goetz, A. Brenning, H. Petschko, and P. Leopold, “Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling,” *Computers & Geosciences*, vol. 81, pp. 1–11, Aug. 2015, doi: 10.1016/J.CAGEO.2015.04.007.

[8] T. Sorsa and H. N. Koivo, “Application of artificial neural networks in process fault diagnosis,” *Automatica*, vol. 29, no. 4, pp. 843–849, Jul. 1993, doi: 10.1016/0005-1098(93)90090-G.

[9] M. Ge, R. Du, G. Zhang, and Y. Xu, “Fault diagnosis using support vector machine with an application in sheet metal stamping operations,” *Mechanical Systems and Signal Processing*, vol. 18, no. 1, pp. 143–159, Jan. 2004, doi: 10.1016/S0888-3270(03)00071-2.

[10] L. B. Jack And A. K. Nandi, “Fault Detection Using Support Vector Machines And Artificial Neural Networks, Augmented By Genetic Algorithms,” *Mechanical Systems and Signal Processing*, vol. 16, no. 2–3, pp. 373–390, Mar. 2002, doi: 10.1006/MSSP.2001.1454.

[11] S. Mahadevan and S. L. Shah, “Fault detection and diagnosis in process data using one-class support vector machines,” *Journal of Process Control*, vol. 19, no. 10, pp. 1627–

1639, Dec. 2009, doi: 10.1016/J.JPROCONT.2009.07.011.

[12] X. Gao and J. Hou, “An improved SVM integrated GS-PCA fault diagnosis approach of Tennessee Eastman process,” *Neurocomputing*, vol. 174, pp. 906–911, Jan. 2016, doi: 10.1016/J.NEUCOM.2015.10.018.

[13] B. Jin, D. Li, S. Srinivasan, S. K. Ng, K. Poolla, and A. Sangiovanni-Vincentelli, “Detecting and diagnosing incipient building faults using uncertainty information from deep neural networks,” *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019*, pp. 1–8, 2019, doi: 10.1109/ICPHM.2019.8819438.

[14] H. Zhao and Z. Lai, “Neighborhood preserving neural network for fault detection,” *Neural Networks*, vol. 109, pp. 6–18, Jan. 2019, doi: 10.1016/J.NEUNET.2018.09.010.

[15] H. Chen, P. Tiňo, and X. Yao, “Cognitive fault diagnosis in Tennessee Eastman Process using learning in the model space,” *Computers and Chemical Engineering*, vol. 67, pp. 33–42, 2014, doi: 10.1016/j.compchemeng.2014.03.015.

[16] V. A. Sotiris, P. W. Tse, and M. G. Pecht, “Anomaly Detection Through a Bayesian Support Vector Machine,” *IEEE TRANSACTIONS ON RELIABILITY*, vol. 59, no. 2, pp. 277–286, 2010.

[17] D. Fernández-Francos, D. Martínez-Rego, O. Fontenla-Romero, and A. Alonso-Betanzos, “Automatic bearing fault diagnosis based on one-class ν-SVM,” *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 357–365, Jan. 2013, doi: 10.1016/J.CIE.2012.10.013.

[18] S. Yin, X. Zhu, and C. Jing, “Fault detection based on a robust one class support vector machine,” *Neurocomputing*, vol. 145, pp. 263–268, Dec. 2014, doi: 10.1016/j.neucom.2014.05.035.

[19] K. Yan, Z. Ji, and W. Shen, “Online fault detection methods for chillers combining extended kalman filter and recursive one-class SVM,” *Neurocomputing*, vol. 228, pp. 205–212, Mar. 2017, doi: 10.1016/j.neucom.2016.09.076.

[20] R. Chalapathy, A. K. Menon, and S. Chawla, “Anomaly Detection using One-Class Neural Networks,” no. August, pp. 19–23, 2018, [Online]. Available: http://arxiv.org/abs/1802.06360.

[21] C. Alippi, I. Fellow, S. Ntalampiras, and M. Roveri, "for Distributed Sensor Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1213–1226, 2013, doi: 10.1109/TNNLS.2013.2253491.

[22] G. Karatzinis, Y. S. Boutalis, and Y. L. Karnavas, "Motor Fault Detection and Diagnosis Using Fuzzy Cognitive Networks with Functional Weights," *2018 26th Mediterranean Conference on Control and Automation (MED)*, pp. 709–714, 2018, doi: 10.1109/MED.2018.8443043.

[23] G. A. Papakostas, D. E. Koulouriotis, A. S. Polydoros, and V. D. Tourassis, "Towards Hebbian learning of Fuzzy Cognitive Maps in pattern classification problems," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10620–10629, Sep. 2012, doi: 10.1016/J.ESWA.2012.02.148.

[24] W. Froelich, "Towards improving the efficiency of the fuzzy cognitive map classifier," *Neurocomputing*, vol. 232, pp. 83–93, Apr. 2017, doi: 10.1016/J.NEUCOM.2016.11.059.

[25] J. J. Downs and E. F. Vogel, "A Plant-wide Industrial Problem Process," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993, doi: 10.1016/0098-1354(93)80018-I.

Chapter 4: Autonomous Fault Diagnosis and Root Cause Analysis Using Machine Learning Techniques.

Preface

A version of this manuscript has been published to the Industrial & Engineering chemistry research Journal. In this chapter, Autonomous root cause analysis methodology has bee been developed using a neural network permutation algorithm. Also, the developed methodology detects and diagnosis the fault condition online using self-updating models. This chapter contributes the targeted 3rd objective of the thesis defined in chapter 1.

Abstract

In this era of Industry 4.0, there are continuing efforts to develop fault detection and diagnosis methods that are fully autonomous; these methods are self-learning, with little or no human intervention. This work proposes a methodology for the autonomous diagnosis of the root cause of detected fault in a complex processing system. The Methodology comprises steps to detect and classify any newly encountered fault, classify the known faults, and find the root cause of the detected fault condition. The one-class SVM model is used in the framework to detect the unlabeled fault, and the neural network is used for fault classification and root cause analysis. The developed Methodology is capable of self-updating the fault database by detecting and diagnosing any new fault condition. A permutation algorithm is applied in the neural network framework to extract the variable's contribution to the classified fault condition. Also, the Spearman rank correlation approach is used to investigate and justify the data correlation and causation. The proposed framework is tested using a continuous stirred tank heater and the benchmark Tennessee Eastman process.

4.1 Introduction

In process systems, a deviation of an observed variable from an acceptable range is defined as an abnormal event. A process can be brought back to its desired operating state if an abnormal event is managed promptly and properly. However, due to poor abnormal situation management, the process industry suffers from financial and environmental losses; in extreme cases, this may lead to injuries and fatalities. In the past and current decades, numerous studies have been done to investigate as well as to prevent abnormal events and provide solutions for effective abnormal situation management (ASM).

The key components of an ASM framework are process monitoring, fault detection, and diagnosis. Physical models, data-driven models, and knowledge-based models are widely used in process systems' monitoring and fault diagnosis [1], [2]. Using these approaches, physical models are difficult to implement due to the complexity and extreme dimensionality of the industrial processes [1]. A knowledge-based model requires human experts in the industry to develop and train the model and to define the normal and abnormal events [3]. The data-driven approaches do not rely on the system's physical model but require good quality historical data [2]. In the past decades, mostly supervised learning methods have been used in data-driven models, requiring a large volume of labelled historical data with sufficient information about the normal and abnormal operations.

However, a vast portion of the historical data is unlabelled in the process industry. In Industry 4.0 era, a key challenge is to develop autonomous fault detection and diagnosis model that will be self-learning for any newly encountered fault. To develop an autonomous fault diagnosis system, the FDD method should be capable of classifying the normal and abnormal conditions from unlabelled data and linking the observed phenomena to the root cause.

Several models and frameworks have been proposed to develop smart process monitoring and maintenance systems over the years. The framework for a smart controller solution for the process industry was recently developed based on self-awareness, self-diagnosis, self prognosis, and self-healing [4]. Machine learning, root cause analysis, and case-based reasoning are heavily integrated with other common frameworks [4], [5]. The concept of cognitive fault diagnosis based on reservoir NN and OC- SVM approach was used to detect a fault condition using unlabelled data [6]. Also, OC- NN and shallow NN approaches are integrated to detect the unknown fault condition and classify the known fault condition by the models [7]. However, both approaches investigated fault without analyzing the clear root cause of the detected fault condition.

The root cause analysis of an abnormal event is another challenge in the multivariate complex process system due to the data correlation, which has been a very active research topic in the last two decades. Root cause analysis methods have been used for studying plant-wide oscillation [8]. Among the reviewed methods, the spectral envelope method was found to be commonly used to determine the causal analysis in the frequency domain [9]. A model-based graphical approach, namely, the adjacency matrix method, has been used, but the method relies to a great extent on the process model; therefore, it has limited use for complex systems [9]. Granger causality analysis is another simple approach to measure the causal effect among the process variables [10]. The transfer entropy method also can be used to define the root cause by finding a suspicious variable among all process variables by utilizing a reconstruction-based contribution method [11]. The Bayesian network (BN) approach is also widely used in process systems to analyze the root cause due to the model's flexibility with requirements for historical data [12], [13]. However, the existing statistical, probabilistic, and stochastic root cause analysis approaches detailed above still lack the capability to be adopted to autonomous fault diagnosis and the growing complexity of the Industry 4.0 environment.

Therefore, from the literature, it is obvious that to develop a smart process plant, it is important to develop self-learning cognitive models to detect, diagnose and define the root cause to maintain the process systems in line with Industry 4.0 concepts.

The main objective of the present work includes i. develop a process monitoring system that can detect an unlabelled fault state from the raw sensor data. ii. Perform root cause analysis for complex process system data using a neural network permutation algorithm, iii. Integrate well-known one-class SVM and NN models to detect the unknown fault condition and classify the known fault condition, iv. Also, this work presents the Methodology to self-update the model whenever a new fault is identified by the FDD algorithm. Figure 4.1 illustrates the proposed integrated model for the autonomous FDD.

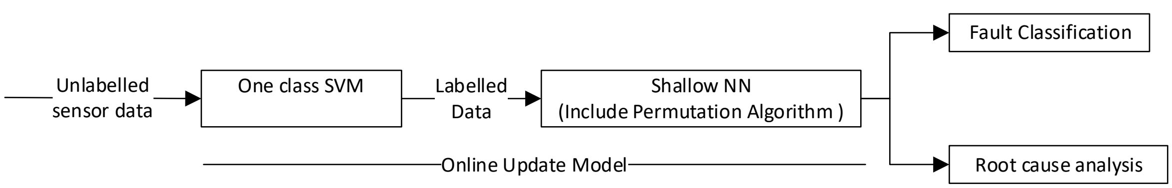


Figure 4.1: FDD integrated model to detect and diagnose unlabeled fault conditions.

The main contribution of this work includes an investigation of autonomous fault diagnosis and root cause analysis, using a machine learning algorithm on the process systems without prior knowledge of the fault state, by integrating OC-SVM to capture the unknown fault condition and NN to classify the known fault condition and determine the root cause for the classified fault. To the authors' knowledge, there is no existing work on autonomous fault diagnosis and root cause analysis on the TE process. All existing work on fault diagnosis and root cause rely on prior knowledge of the fault pattern. This work also contributes to the machine learning approach to learn the dynamically constructed fault dictionary, including the fault's root cause in real-time, by self-updating the models in online conditions.

The rest of the section is organized as follows. The well-known mechanism of OC- algorithm and classification algorithm is introduced for primary fault detection and diagnosis. Then, the proposed Methodology that consists of the correlation test and neural network model is implemented with root cause diagnosis, using a permutation algorithm described in detail. To develop and test the proposed Methodology, a continuous stirred tank heater (CSTH) process system with five variables is used. The TE process system with 22 variables is used for a benchmark to test and evaluate the proposed method. Finally, major findings, conclusions and further work of this study are summarized.

4.2 Background

Considering Industry 4.0 and the predicted future trends in the implementation of autonomous fault detection and diagnosis, it is important to detect a fault condition using unlabeled data when data may exhibit less information about the fault condition. Anomaly detection and clustering in machine learning algorithms are widely used in unsupervised models [14]. Also, to diagnose the fault condition, it is essential to investigate the fault by classifying the fault condition and defining the root cause for each fault condition. To address the identified challenges, in the proposed Methodology, an FDD model is configured to detect the unknown fault using one-class SVM and to analyze the root cause for the detected fault using NN and a permutation algorithm.

4.2.1 One Class SVM for Unlabelled Fault Detection

In the one-class SVM method, a hyperplane is constructed in the SVM instead of a hypersphere. The hyperplane separates the target class data points with the maximal margin from the origin, where all the outliers are assumed to fall on the plane through the origin [15]. The one-class SVM has been successfully implemented for process system fault detection. Malfunction in a chiller system [16], various faults' detection in Tennessee Eastman process data [17], and the temperature sensor fault case study in a microreactor system [18] are a few recent applications.

OC- SVM maps the training data points into the feature space corresponding to the kernel to separate them from the origin with a maximum margin. For the test data points, the value of OC-SVM decision function g(x) is determined by evaluating the side of the hyperplane where it falls in feature space. To evaluate the outlier, the decision function g(x) will return the value of +1 for the target class region and -1 for outliers. The decision function of g(x) is given by:

$$g(x) = sign\ (w.\varphi(x_i) - \rho) \tag{4-1}$$

where, 'φ' is a nonlinear mapping function to represent a data point 'x_i' in feature space. 'w' and 'ρ' are the hyperplane characteristics of the algorithm, which has a maximal distance from the origin. Therefore, the main goal of an algorithm is to determine the hyperplane to separate the target class from the origin. The following quadric programming minimization equation is used to separate the target class samples from the origin.

$$\min_{w,\varepsilon_i,\rho} \frac{1}{2}\|w\|^2 - \rho + \frac{1}{NC}\sum_{i=1}^{N} \varepsilon_i \tag{4-2}$$

Subject to $w.\varphi x_i \geq \rho - \varepsilon_i, \forall i \qquad \varepsilon_i \geq 0$

In Equation 4.2, N is the number of training samples. $'\varepsilon_i'$ is a slack variable for data point x_i that allows x_i to locate outside the decision boundary and not meet the margin requirement at a cost.

The variable 'C' is the asymptotic fraction of outliers allowed in the training dataset, C $\in [0,1]$. Scholkopf et al. (2001) proved that C is both the upper bound on the fraction of outliers and the lower bound on the fraction of support vectors [15]. Also, they proved that support vectors are the points residing in the exterior of data distribution; hence the interior data points are discarded after the training[15], [19]. Therefore, the one-class SVM can be trained using the selected data points.

In Equation 4.1, the nonlinear function φ connects the input domain X and the feature space (F). The inner product of φ is computed by evaluating a kernel function and used for nonlinearization,

$$k(x_i, x_j) = \varphi(x_i).\varphi(x_j) \tag{4-3}$$

Appropriate selection of kernel function avoids the explicit mapping between the $X \rightarrow F$. The commonly used kernel functions are discussed in[15], [16], [19], [20].

In the past and in recent years, support vector data description (SVDD) and OC- SVM have been used in many applications to determine the anomaly points and detect the operation deviation. For the past two decades, one-class support vector classifiers have been the most popular research topic in data-driven fault detection methods [19]. However, one of the main drawbacks observed in OC-SVM is that training data may contain some outliers due to the data noise that will reduce the performance of the classifier. To overcome this issue, robust one-class SVM with Gaussian-based penalization is proposed by Prayoonpitak and Wongsa [20] to eliminate the outlier from the training dataset. Further, to eliminate the suspected outliers from the training data, the concept of automatic adjustment in the tuning parameters is proposed iteratively, using a bisection algorithm

[17]. Arunthavanathan et al. [7] proposed method based on the number of anomalies in the training data frame using one-class SVM, the margin level depended on the data noise level.

An incremental one-class SVM was initially proposed based on the batch algorithm to detect the multiple deviation points [18]. An incremental one-class approach was developed by using the learning by the model algorithm. In the FDD application, incremental learning uses each one class learner to represent each fault condition by updating the model [6]. Furthermore, to detect the unknown fault condition, an incremental one-class NN approach was proposed by Chalapathy et al. [21].

4.2.2 Fault Classification Using NN.

Neural network fault classification methods were initially proposed using the multilayer, feedforward, analog perception, and the generalized delta learning rule [22]. A supervised learning approach based on back propagation learning was proposed by Venkatasubramanian et al. [23], and the feasibility of this approach is demonstrated through a neural network-based fault diagnosis case study.

In autonomous fault detection and diagnosis, dynamic fault classification has received more attention in recent years. Dynamic fault classification models run in an adaptive mode, learn the detected fault condition data pattern automatically, and use the fault signature to classify the fault condition in the future. Lit et al. proposed the dynamic fault classification method based on an artificial neural network with a moving window to segment training, and the subsequently trained network is applied in the online process data [24]. A two-stage neural network model was proposed by Maki and Loparo, the first stage detects the dynamic trend of each measurement, and the second stage detects and diagnoses the fault. In the proposed model, a moving time window technique is used to track dynamic data and detect the transient state of the fault [25]. Also, for the dynamic fault classification, neural network-based unmeasurable operating variables in chemical processes, using the moving time window approach, are proposed by Yang et al. [26].

However, all these approaches have used the moving window to test fault detection. To detect the dynamic fault condition, dynamic output neural network was proposed by Arunthavanathan et al. [27]. The dynamic output neural network is formulated based on the number of detected faults in real-time and the number of neural network outputs modeled in the output layer. The complete model is described as follows:

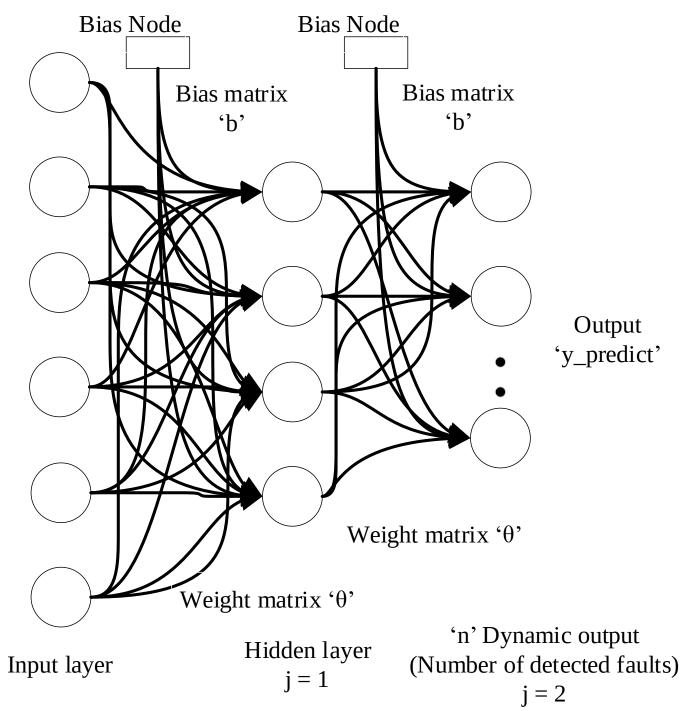


Figure 4.2 Dynamic output neural network. [27]

The dynamic output neural network is illustrated in Figure 4.2, and the objectives are formulated as follows:

$$w_i^j = g(x_i.\theta_i^j + b^j) \qquad (4\text{-}4)$$

where $'w_{i}^{j}{}'$ is activation of node ‘i’ in layer ‘j’, including hidden and output layers. Initially, for an input layer, $'x_i'$ is an input variable adopted in the input layer, and for further hidden and output layers, $'x_i'$ is an updated activation node, $'\theta_i^j{}'$ is the weight matrix for each layer, and $'b^j{}'$ is a bias matrix in each layer. $'g'$ is a nonlinear activation function. To determine the appropriate input and output relationship, a different activation function is used. The SoftMax activation function is used in the proposed Methodology to evaluate the fault classification index. The transformation is shown in Equation 4.3 and is finally connected to the SoftMax function to determine the network output, as shown in Equation 4.5.

$$S(y_i) = \frac{\mathrm{e}^{(y_i)}}{\sum_{j=1}^{N} \mathrm{e}^{(y_j)}} \tag{4-5}$$

In Equation 4.5, $'y_i'$ is an input vector and can take any value between -/+ ∞. $'y_j'$ is a normalized output vector with a value range between 0 and 1. $'S(y_i)'$ is a softmax transformation of y_i, and ‘N’ defined the nodes in the output layer.

4.2.3 Root Cause Analysis Based on Permutation Feature Importance.

Permutation feature importance was initially proposed to estimate the feature importance in the trained models. Breiman [28] proposed a permutation algorithm in a random forest approach. In the proposed model, permutation importance is defined as the mean decrease in accuracy of the trained model when each feature is permuted [28]. This approach disturbs the relationship between input variables and output targets; therefore, the drop in model accuracy is defined as the target classification dependence on the impact variable.

To rank the importance of input parameters, Sung [29] proposed a method to measure the importance of input by observing the change from mean squared error (MSE) when the input is deleted from the neural network [29]. However, when noisy and redundant inputs are present, this method fails to determine the correct observation. To overcome this issue, de Oña and Garrido [30] proposed a method to introduce noise to one of the inputs, while all other remaining variables are kept at the original noise level in the machine learning model, and the MSE between model output before and after the noise addition is compared to determine the cause variable [30].

In our proposed algorithm, while shuffle the data points in one variable, all other variables data points are fixed and the NN accuracy is tested before and after shuffling, finally MSE is used to determine the important variable. Complete algorithm development is described in section 4.3.1.3.

4.3 The Methodology for Autonomous Fault Detection

As described in Figure 4.3, this work proposes a fully autonomous workflow to detect, diagnose and find the root cause for an abnormal event.

In offline conditions, one-class SVM and Neural Network models are trained by using nonfaulty data. However, in the multivariant raw sensor data sets, data points are far from each other. Therefore, before applying the data to the algorithms, a standard scaler is used to remove the mean and scale the data points between 0 and 1.

The Spearman correlation test is used to determine the highly correlated variables among the multivariant data sets and remove one of the highly correlated variables before training the models. Section 4.4.1 describes the correlation test and the outcome result.

In online conditions, the trained one-class SVM and NN models are used for real-time monitoring. The raw sensor data is scaled using a standard scaler and sampled using a moving window, and tested for anomaly by using one-class SVM. The number of anomalous points are counted within the window and compared with the predefined margin. If the number of anomalous data points exceeds the margin, the tested data window is identified as having unknown fault symptoms. The newly identified faulty data is concatenated with the faulty data in the existing data repository. The concatenated data is used to retrain the one-class SVM model. Also, the detected data frame is labelled as a new fault condition and used to train the NN with an updated output node.

In the anomaly test, if the counted anomaly detection points remain within the allowable margin, the data frame is identified as having the known fault condition. Therefore, without any model updates, the data frame is fed into the NN model to classify the fault condition, and the NN permutation algorithm is used to identify the root cause for the detected fault condition. In this approach, models are trained in offline conditions only at the initial stage, and subsequently, the model updates them by itself in online condition whenever the new fault is identified.

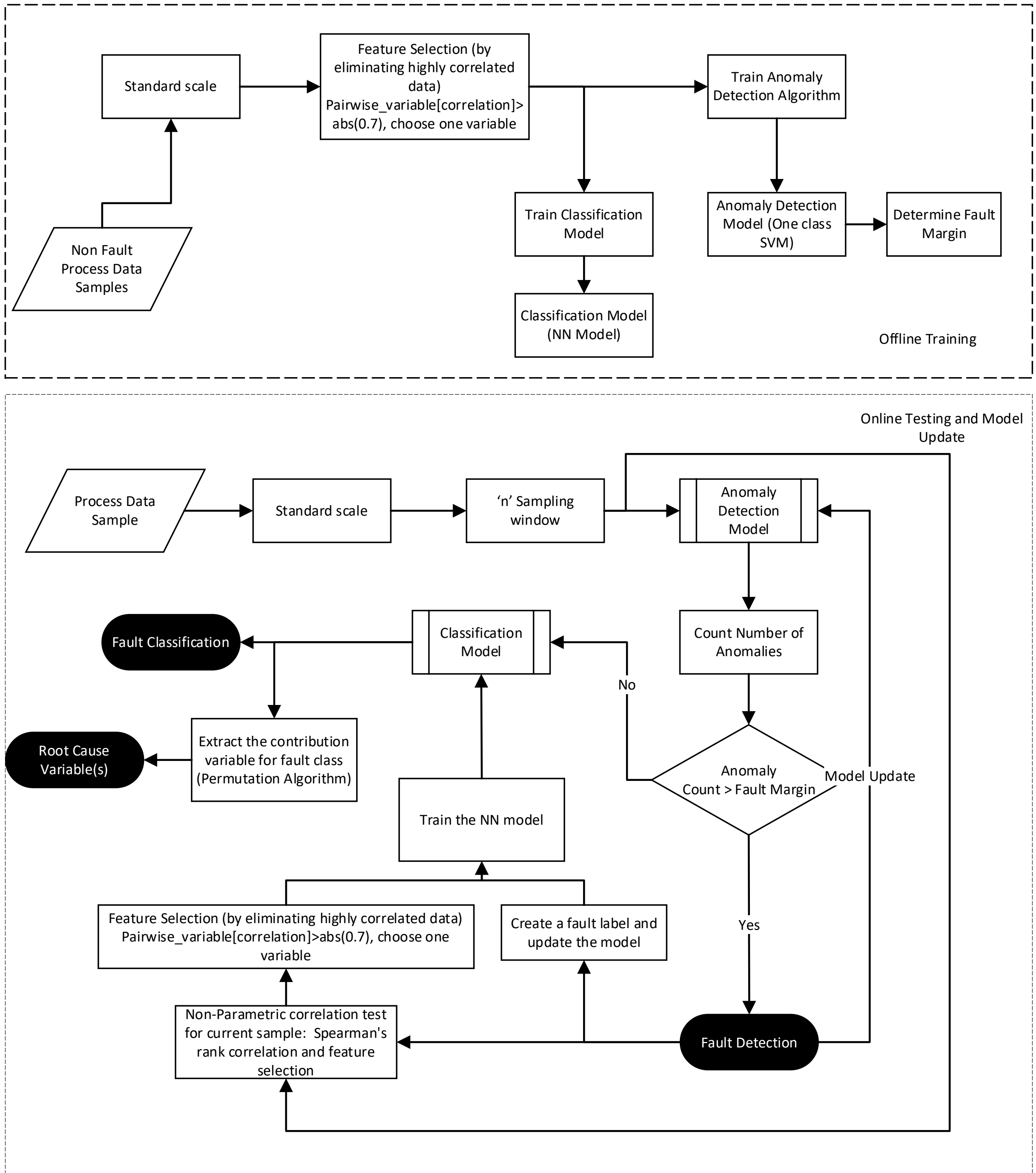


Figure 4.3 The proposed Methodology

4.3.1 The Proposed Framework

The proposed framework is divided into three major algorithms. Algorithm 1 is most similar to the one-class SVM proposed by Chen et al. [6]. However, it modified to learn by itself to update the model whenever a new fault condition is detected in the online condition. Algorithm 2, similar to the dynamic output neural network proposed by Arunthavanathan et al. [27]. The proposed model was modified for online updates with the monitored NN hyperparameters such as learning rate and a number of epochs for better accuracy and less loss. During the online updates, existing NN and one-class SVM parameters are recalled and used as an initial value to update the models. Finally, algorithm 3 is proposed in this work to define the root cause by detecting the important variable for the classification of the well-trained NN model in algorithm 2.

4.3.1.1 Incremental One-Class SVM

OC- SVM is developed in algorithm 1 to determine the decision score +1 to -1 by using Equation 4.1. To optimize the w and ρ parameters, the gamma function and radial basis function kernel with a 1e-3 learning rate are used in the model architecture. To detect the fault condition, 100 sampling moving windows are fed into the OC- SVM and the number of outliers tested within the window. The length of the window depends on the nature of the process systems and the frequency of sensing the data. The fault margin is developed based on the experimental result in normal operating conditions. In the proposed algorithm, OC- SVM marginal level is determined by the number of anomalies counts in the data frame. The results are reported with different noise levels in the training data; the margin was increased to 20% for 5% noise level, and the margin was set to 25% for 10% noise level in the data. Therefore, the OC- SVM fault margin is developed as follows,

$$Margin = \max(anomaly_{count}) + \delta * \max\,(anomaly_{count}) \qquad (4\text{-}6)$$

In the Equation 4.6, $\max\,(anomaly_{count})$ defines the maximum number of anomalies in the nonfaulty training dataset, and δ defines the noise margin.

Algorithm 1: Incremental one-class SVM

Offline training of model/Online Update

1a. **Input** Nonfaulty preprocess data with nonfaulty label / detected faulty data with the updated label.

2a. **Output** optimized OC- SVM /updated OC- SVM decision function.

3a. **for** (1: length_data)**:**

4a. Optimize the w and ρ in the SVM decision function defined in Equation 4.1.

end

5a. Run the validation nonfault preprocess data set to validate the decision function.

6a. Set the #anomaly margin using validation data.

Online/Real-time test

1b. **Input** Preprocess data sampled using ‘n’ window.

2b. **Output** set of decision score/anomaly count for each windowed sample.

3b. **for** (1: length_data)**:**

4b. Compute the decision score for each sample: g(n)

5b. if (g(n) == -1) then

6b. Anomaly_count = Anomaly_count+1

7b. else

8b. Normal operation condition.

end

9b. **if** (Anomaly_count > Margin)**:**

10b. Abnormal operation detected and new fault label created.

11b. Update the incremental OC- Model using the current and past sampling window.

12b. Update the dynamic output NN (Algorithm 2a) using current and past sampling windows.

13b. **else:**

14b Run the algorithm 2 Online/Real-time test for fault classification.

4.3.1.2 Dynamic Output Neural Network

To determine the root cause using the permutation algorithm to obtain the feature importance in a neural network, it is important to train the NN with minimal mean squared error. Algorithm 2 is developed by appropriately selecting the parameters.

A dynamic output NN model architect with the number of features in the input layers, dynamically changing output layer, and one hidden layer with 68 neurons. A ReLu activation function is used in the hidden layer, and SoftMax activation provides a better solution for binary one-hot encoding classification. An Adam optimizer with an initial 1e-3 learning rate is used to optimize the parameters using backpropagation, and categorical cross-entropy is used as a loss function to determine and validate the developed model. When learning and updating the NN model online, the Adam optimizer learning rate is optimized by running it with the range of the learning rate. The number of epochs for the NN learning also optimized using model accuracy and the mean squared error.

Algorithm 2: Dynamic output NN

Offline train model / Online Update

1a. **Input** Nonfaulty preprocessed data with a nonfaulty label (one-hot encoding) / detected faulty data with the updated label.

2a. **Output** Dynamic output NN model.

3a. Randomly initialize the weights

4a. Implement the forward propagation and determine the cost function.

5a. **for** (1: length_data)**:**

6a. Perform forward propagation and back propagation to optimize the w and b in Equation 4.3.

end

7a. Verify the backpropagation by using MSE and determine the appropriate learning rate for the optimization algorithm.

8a. NN_Model is developed for label prediction.

Online/Real-time classification

1b. **Input** Preprocess data is sampled using the 'n' window.

2b. **Output** Fault classification for a known and labeled fault condition.

3b. Develop/updated NN model (input sample)

4b. **for** (1: length_input sample)**:**

5b. Predict the one-hot encoding classification for each sample.

6b. Convert the predicted one-hot encoding label to a classified label.

end

7b. Using the classified label, define the test accuracy.

4.3.1.3 Permutation Algorithm for Root Cause Analysis

A permutation feature importance algorithm highly depends on fitted machine learning models. Therefore, it is important to implement high accuracy predictive models and cross-validate prior to the algorithm development. In the proposed method, algorithm 2 dynamic output NN model fitted with the permutation algorithm to determine the feature importance.

Algorithm 3: Permutation algorithm to determine the impact variable in NN classification.

1 **Input** Trained and Fitted shallow Neural Network model validates mean absolute error (MAE).

2 **Output** High impact variable due to the fault occurrence.

3 **for** (j = 1: number_of_column/variable):

4 Generate permuted test data: $(p_{j,i}, y_i)_{i=1}^{n}$, where $p_j \in x$ randomly shuffled jth column variable from the dataset; n is the number of samples from a previous window and current data window.

5 Generate prediction target from the developed model: Data set real output y_x, and permuted prediction, $\hat{y}_{p_{j,i}}$.

6	Permutation importance score calculation using mean absolute error: Perm_MAE $(y_x, \hat{y}_{p_j}) = \frac{1}{n}\sum_{i=1}^{n}\left\|y_{x,i} - \hat{y}_{p_{j,i}}\right\|$ (4-7) Permutation score for jth column variable $= \frac{\|\text{Perm\_MAE} - \text{MAE}\|}{MAE\ x\ 100}$ (4-8)
	end
7	Analyze the permutation importance score of each variable.

Using the randomly shuffled predictor as a permutation, the permutation importance score compares the importance of each feature to an identically distributed predictor and reduces potential bias. Additionally, in this approach, the score is produced using the test data, which provides a more accurate portrait of how the model is affected due to the variable changes. The disadvantages of the permutation test observed by Hooker and Mentch [31], based on the tree-based algorithm. For highly correlated variables with more than 85% of correlation, rate are discussed in various publications [31], [32].

However, to overcome the identified issue in the permutation algorithm, it is important to analyze the data correlation prior to applying the neural network model and eliminate one of the highly correlated variables from the data. Also, it is important to train the NN model with over 95% accuracy to obtain the appropriate root cause variable. Therefore, in the proposed method before training the NN, the Spearman correlation test has been done.

4.4 Algorithm Development and Testing

To test the proposed framework, CSTH model datasets were initially used in the testing. The CSTH is a commonly used subsystem in most of process systems. Thornhill et al. (2008) [33] proposed the CSTH simulation model. The developed model is highly nonlinear, and it contains real disturbance data [33]. In a simple CSTH model, cold and hot water are mixed and heated using steam through a heating coil. Finally, the cold and hot water are drained off through a long pipe, as shown in Figure 4.4. The input of the model steam and cold water valves are operated by an electronic signal in the range of 4-20mA. The outputs are measured for temperature, level, and cold water flow rate, nominally in the range of 4-20mA [33].

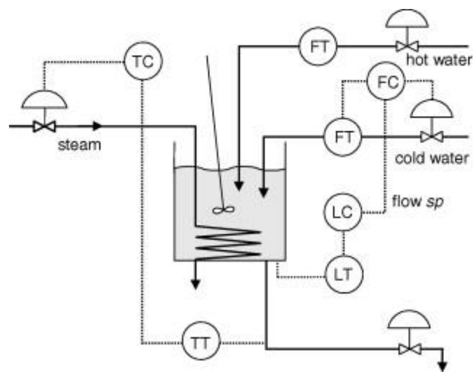


Figure 4.4 The continuous stirred tank heater [33]

Two fault scenarios, as detailed in Table 4.1, have been considered in this experiment to test the model input fault condition. In F1 fault condition, fault is applied in the steam valve position, and in F2 fault condition, the fault is introduced in the cold-water valve. In each fault condition, 1400 nonfaulty samples and 100 faulty samples are generated for the experiment.

Table 4-1 Fault description in the CSTH

Fault ID	**Fault Description**
F1	Steam valve position
F2	Step change in cold water valve

4.4.1 Experimental Procedure

Step by step algorithm development and the experimental procedure are given below. In offline, the raw sensor simulated data are preprocessed initially using zero mean and unit variance. Scaled data then fed into the Spearman correlation test to eliminate one of the correlated variables. In the online and real-time processes, all moving window samples are preprocessed using the same mean, and the unit variant is determined in the offline condition with nonfaulty data.

Step 1: Initially, 300 data samples from the non-faulty condition data were used to train algorithms 1a and 2a offline.

When training the NN model, it is important to develop the model with high accuracy. Therefore, the Adam optimizer learning rate is tested over the range of learning rates.

By properly selecting the learning rate and a number of epochs in the NN model, accuracy can be maintained over 95%. Figure 4.5 shows the different learning rates and the number of epochs' impact on the accuracy. For the current CSTH system, a 0.1 learning rate with 150 epochs gives a better performance for the initial trained and validated data.

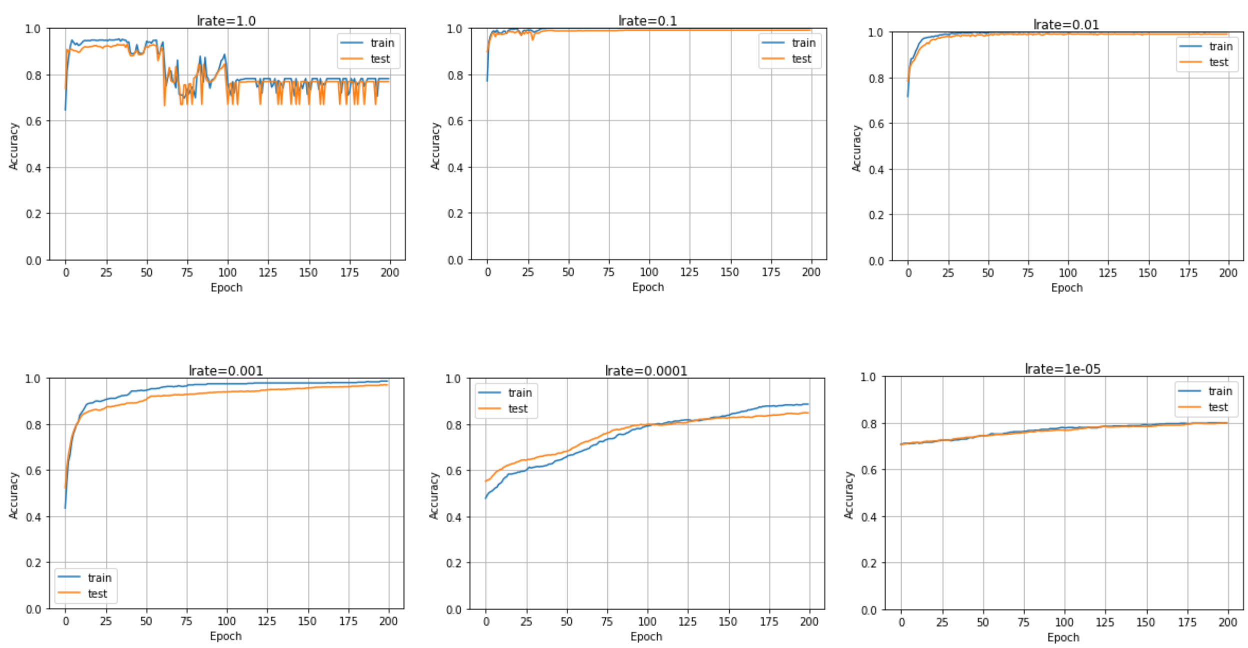
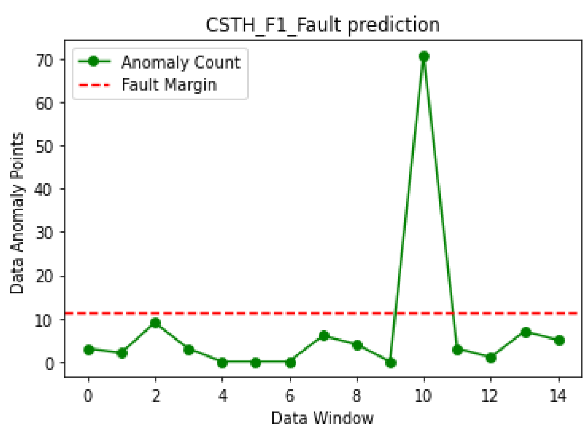


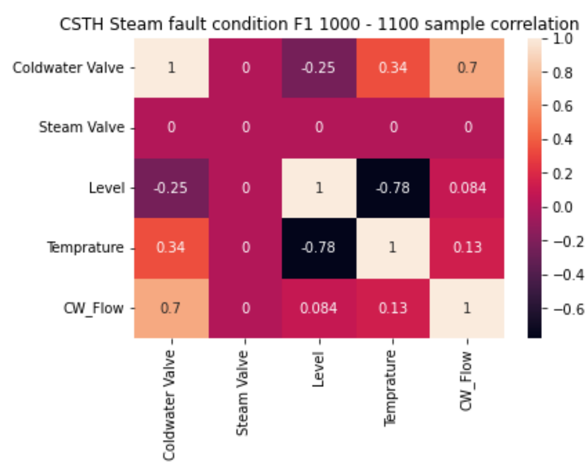
Figure 4.5 Proposed NN model accuracy and loss over the number of iterations.

Step 2: Preprocessed multivariate sensor data (with multiple sensors) are windowed with 100 samples and fed into the trained one-class SVM described in algorithm 1b to determine the process deviation. If a number of anomalies in the data sampling window exceeds the marginal level, the data window is defined as a possible fault condition and updates the one-class SVM and dynamic output NN. In the tested data sample window, a number of anomalies are within the margin, data window is fed into the dynamic output NN to classify the fault condition.

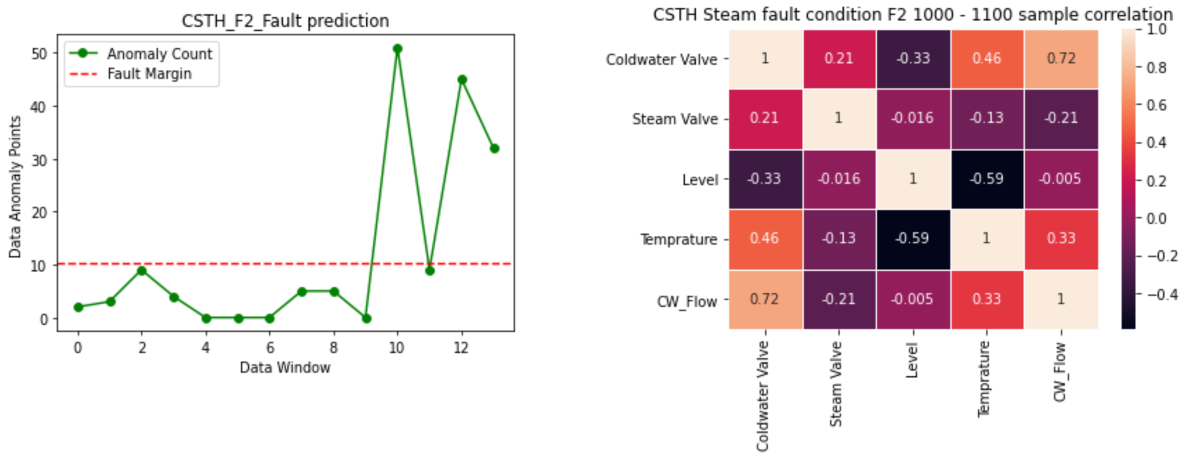
Once the new fault is detected online, the data window is forced to do the correlation test and remove one of the correlated variables from the dataset before training the dynamic output NN. This test has been done to overcome the permutation algorithm issues. From the experiment, a Spearman correlation over 0.7 is defined as a high correlation in the proposed model. Any correlation value of 0.7 or less does not affect the permutation test in the proposed model.

The one-class SVM results for CSTH fault F1 and fault F2 are shown in Figures 4.6a and 4.6b. In both cases, the fault is introduced at the 1000th sample and detected accordingly. Further, as shown in Figure 4.6, based on the fault condition F1, the Spearman correlation test does not recognize any high correlation variables. As in fault condition F2, the cold-water flow and water flow valve correlation are recognized as high correlations due to the step changes in cold water.





(a)



(b)

Figure 4.6 OC- SVM test and Spearman correlation test for CSTH fault.

Step 3: After testing the anomaly condition using one-class SVM, and if the data samples are under the anomaly margin, samples are fed into the NN to classify the fault condition.

Step 4: The classification NN model is further investigated with a permutation algorithm, as described in algorithm 3. The NN model uses a permutation algorithm to test each sampling window to define the most contributed variable for fault classification after determining and eliminating the correlated variable. By using this method, for each fault condition, the cause variable or root cause is determined online.

For the CSTH tested fault condition, the root cause variable is correctly identified by the proposed algorithm. As shown in Figure 4.7 and Table 4.2, the root cause for fault F1 is defined as a steam valve variable, where the cause for the operation deviation is determined as the steam valve position. Similarly, the root cause for fault F2 is determined as a cold water valve, where the cause of fault F2 is the step-change in the cold water valve.

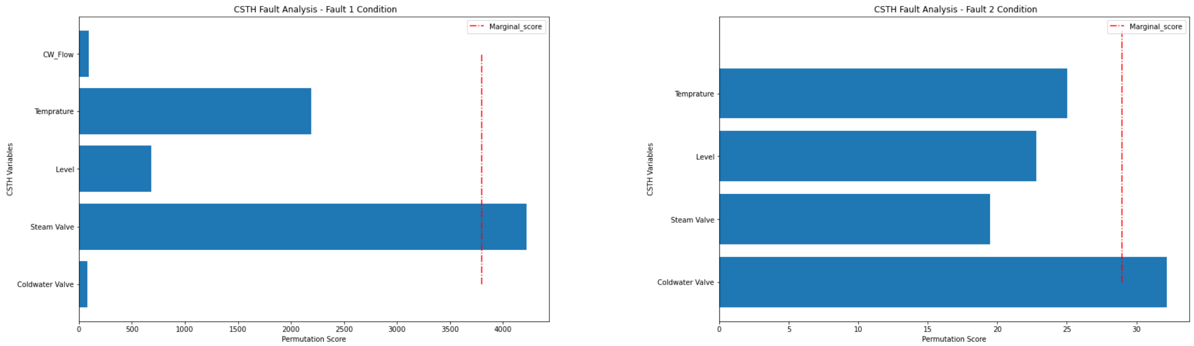


Figure 4.7 Fault F1 and F2 root cause analysis using permutation algorithm.

Finally, the proposed model is evaluated by the CSTH process model and the dataset. Identified fault symptoms are detected at the same sampling window, and the root cause for the data deviation is detected accurately by the NN permutation algorithm.

Table 4-2 Fault Comparison

Fault ID	**Fault Description**	**Root Cause / Contribution of variable**	
		NN permutation algorithm	**Elapsed Time (sec)**
F1	Steam valve stiction	Steam Valve	4.1
F2	The step-change in cold water	Cold-water valve	4.2

Furthermore, the proposed framework is developed using a python platform and tested using a core i7 intel microprocessor, with 3.0 GHz speed and16 GB RAM configuration. As shown in Table 4.2, diagnosing the fault condition took less than 5 sec. This time duration is calculated based on NN running time to diagnose the fault condition online. However, Elapsed time does not include the one-class SVM run time and model update time.

4.5 Application and Benchmarking

The application of the proposed approach to detect and diagnose the fault condition autonomously is tested on the benchmark TE process data. Fault diagnosis performance of the proposed method and its run-time are compared with those of the recently proposed symbolic dynamic-based normalized transfer entropy model [34].

TE process data are generated using a simulated model developed by Down and Vogel (1993) [35]. As shown in Figure 4.8, the TE process consists of a reactor, a condenser, a compressor, a separator, and a stripper. By feeding the A, C, D, and E to the reactor liquid, products G and H are produced. The species F is a by-product of the process.

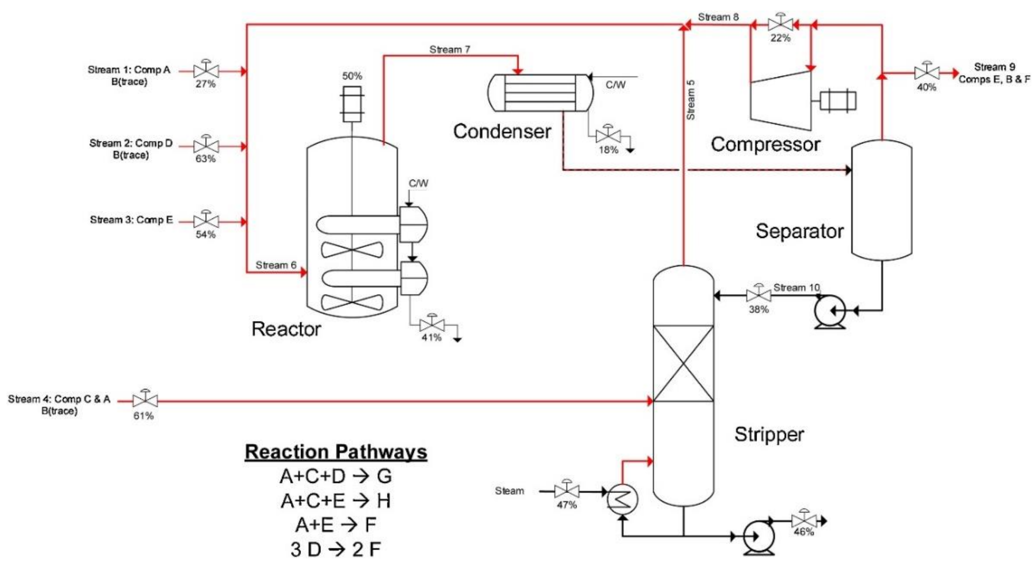


Figure 4.8 Tennessee Eastman process flow [36]

In the TE process, a total of 53 measured variables can be generated in the simulation process. Out of the 53 measured variables, 22 process variables are continuous process measurement, 19 variables are composition measurement, and the remaining 12 variables are manipulated variables. To test and apply the proposed framework, 22 continuous variables, listed in Table 4.3, are used.

Table 4-3 TE process continues process variables.

Index	**Description**	**Index**	**Description**
XMEAS1	A feed (stream 1)	XMEAS12	Separator level
XMEAS2	D feed (stream 2)	XMEAS13	Separator pressure

XMEAS3	E feed (stream 3)	XMEAS14	Separator underflow (stream 10)
XMEAS4	Total feed (stream 4)	XMEAS15	Stripper level
XMEAS5	Recycle flow (stream 8)	XMEAS16	Stripper pressure
XMEAS6	Reactor feed rate (stream 6)	XMEAS17	Stripper underflow (stream 11)
XMEAS7	Reactor pressure	XMEAS18	Stripper temperature
XMEAS8	Reactor level	XMEAS19	Stripper stream flow
XMEAS9	Reactor temperature	XMEAS20	Compressor work
XMEAS10	Purge rate (stream 9)	XMEAS21	Reactor cooling water outlet temperature
XMEAS11	Separator temperature	XMEAS22	Condenser cooling water outlet temperature D

In the TE simulation process, 21 different process faults can be generated. Faults IDV 1 to 8 are related to step changes in the related variables[42]. Faults IDV 9 to 12 are related to random variation of the variables. Slow drift in a reaction kinetic fault is demonstrated in IDV 13. Faults IDV 14, 15, and 21 are related to sticky valves. The remaining Faults IDV 16 to 20 are faults with unknown causes. Table 4.4 summarizes the faults that are tested using the proposed models, and faults are selected in each common type.

Table 4-4 Selected TE process fault condition for testing.

Fault ID	**Description Variable**	**Type**
IDV1	A/C feed ratio, B composition constant (Stream 4)	Step
IDV4	Reactor cooling water inlet temperature	Step
IDV5	Condenser cooling water inlet temperature	Step
IDV6	A feed loss (Stream 1)	Step
IDV11	Reactor cooling water inlet temperature	Random variation
IDV12	Condenser cooling water inlet temperature	Random variation
IDV14	Reactor cooling water valve	Sticking
IDV15	Condenser cooling water valve	Sticking

To demonstrate the algorithms, 1500 data samples are generated from each fault condition. In the generated samples, 1000 to 1100 samples are under normal operations, and the remaining 400 to 500 samples are under fault conditions.

As described by the algorithm steps in section 4.1, 500 nonfaulty preprocessed samples are used to train the one-class SVM and dynamic NN offline. In the online real-time process, 100 samples of a moving window are generated to follow the remaining steps.

4.5.1 Proposed Algorithm Test

Proposed stand-alone fault detection and diagnosis with the proper root cause analysis test result are demonstrated with the two different stages. OC- SVM, described in the algorithm 1 testing result, demonstrates the fault detection using unsupervised data, and the NN model described in algorithms 2 and 3 analysis results demonstrates the fault classification and the root cause for the classified fault and fault diagnosis time.

4.5.1.1 One Class SVM Algorithm Test

Figure 4.9 shows the one-class SVM testing result for each selected fault condition. For IDV12, the Fault was introduced at 1100 samples and detected in 12th sampling window. The rest of the faults are detected in the 11th sampling window and where actual faults are introduced at the 1000th sample.

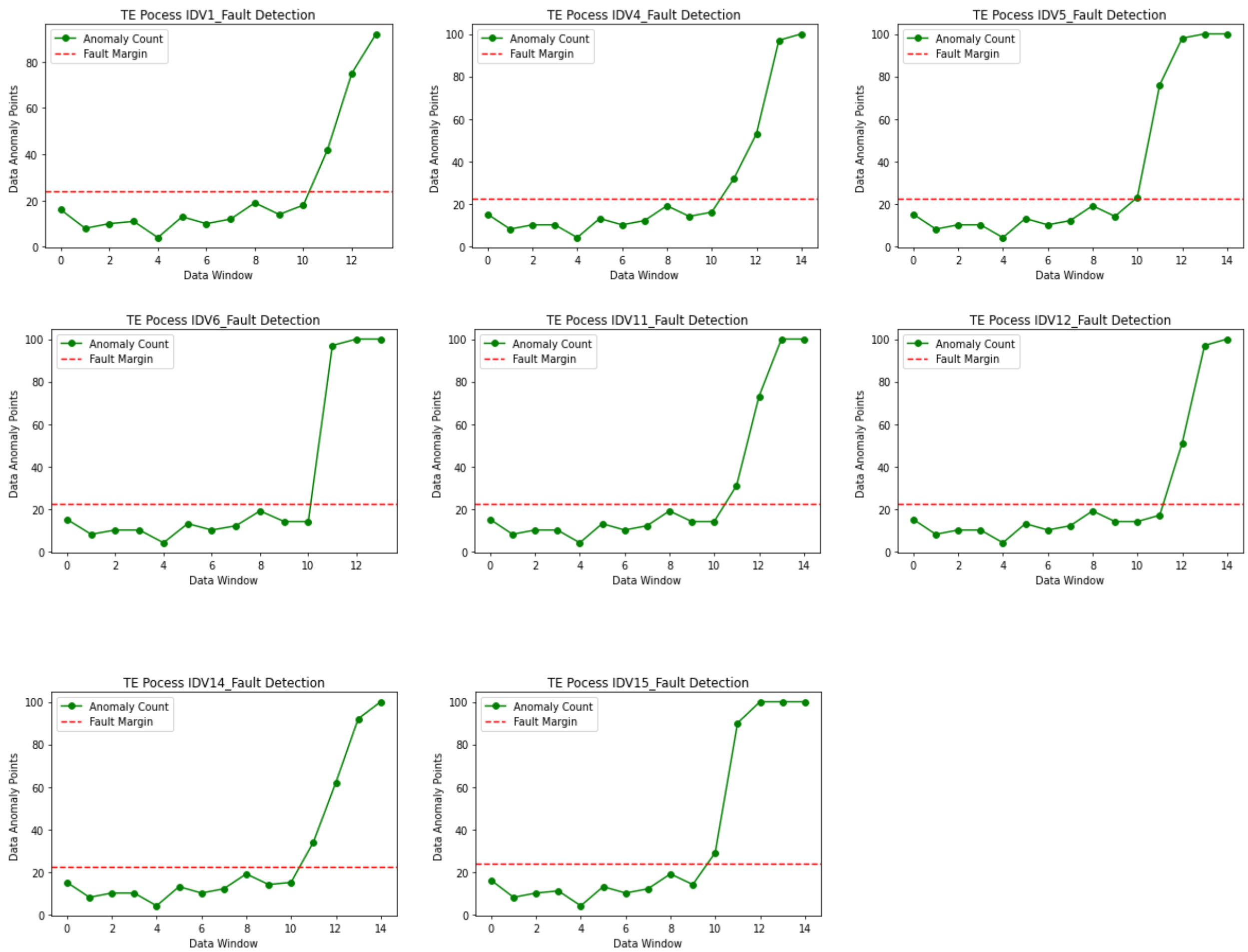


Figure 4.9 TE process data fault detection using one-class SVM (Algorithm 1)

4.5.1.2 Autonomous and Self-Update Test

The autonomous fault detection and classification test results are shown in Figure 4.10 and Table 4.4. To test the autonomy and model self-learning, TE process faults condition data are concatenated to provide different and same fault conditions over the different operating points, as shown in Table 4.5.

Table 4-5 TE fault condition for automated test

Data sample	**Fault condition**
1-1000	No-fault condition

1001-1500	Fault 1
1501-2000	No-fault condition
2001-2500	Fault 4 condition
2501-3000	Fault 11
3001-3400	Fault 4
3401-3700	Fault 1

Fault and fault free combined data are sampled using 100 sampling windows and fed into the one-class SVM to detect the unknown fault condition. As shown in Figure 4.10, in test 1, non-faulty data have trained the OC-SVM and NN offline, and the test results are shown for 1 – 1000 samples and 1501 – 2000 samples detected as a normal fault condition. In test 2, after the fault 1 condition is detected at the 1001 sample point, the OC-SVM and NN model self-learn using normal and fault 1 data. As shown in Figure 4.10 in test 2, 1001 – 1500 samples and 3401 – 3900 samples are detected. In test 3, after fault 4 is detected at the 2001 sample point, the NN model self-updates using no-fault, fault 1, and fault 4 data. As shown in the result, 2000 -2500 samples and 3001 – 3400 data samples are detected.

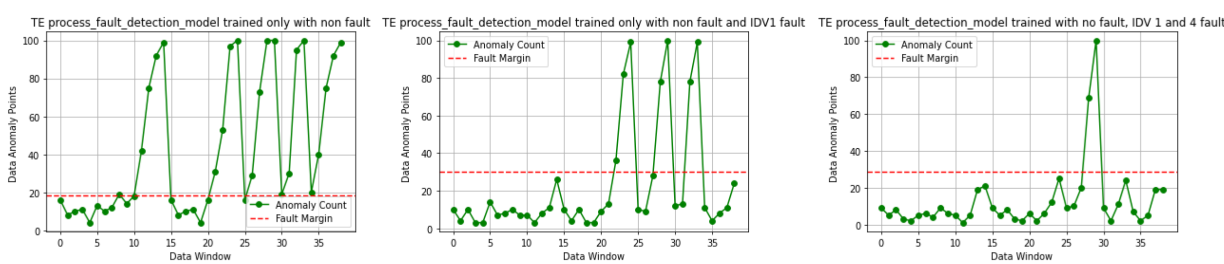


Figure 4.10 Autonomous and model self-update test.

In this way, whenever a new fault is detected, OC-SVM is updated to further detect the new fault condition, and NN is updated with a new label to classify the same fault condition. Also, Figure 4.10 illustrates that once the fault is detected and updated by the one-class SVM, the same fault will not be detected further as a new fault condition. Table 4.6 summarizes the model update, detection, and classification test for this concatenated TE process data.

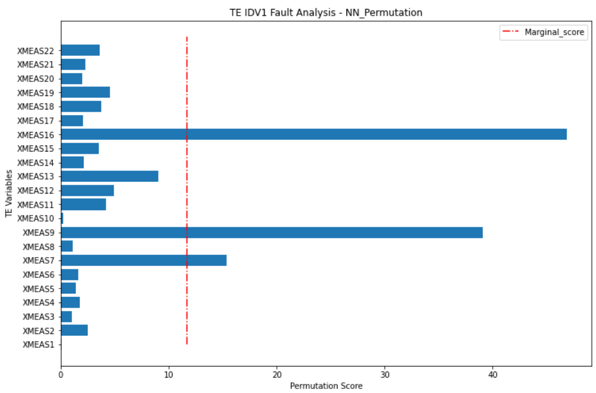
Table 4-6 Data samples and obtained results using one-class SVM and NN

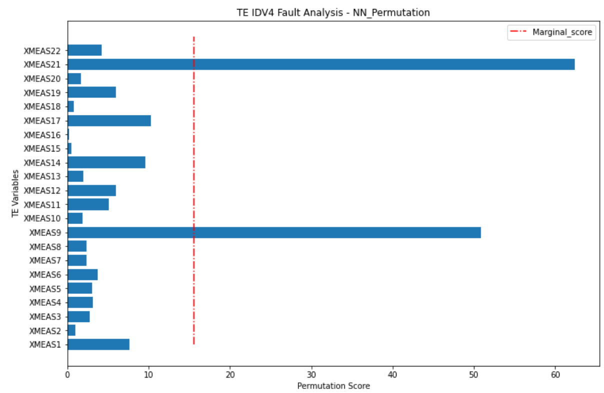
Data sample	Model update and results
1 – 600	Offline trained OC-SVM and NN.
601 – 1000	No-fault classified (100%)
1001 – 1300	Fault 1 detected and OC-SVM and NN updated.
1301 – 1500	Fault 1 classified using NN. (98%)
1501 – 2000	No fault classified using NN. (97%)
2001 – 2300	Fault 4 was detected and OC-SVM and NN were updated.
2301 – 2500	Fault 4 classified (97%)
2501 – 2800	Fault 11 detected and model update.
2801 - 3000	Fault 11 classified (96%)
3001 – 3400	Fault 4 classified (94%)
3401 – 3700	Fault 1 classified (96%)

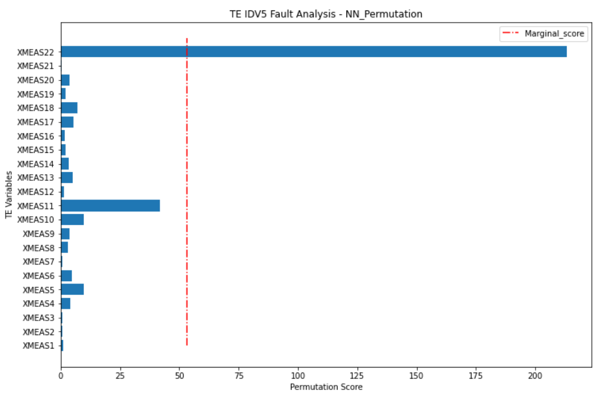
4.5.1.3 Permutation and Root Cause Analysis Test

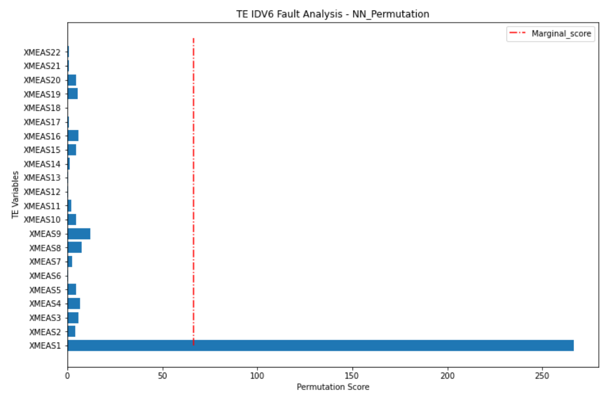
The NN model permutation algorithm testing result is illustrated in Figure 4.11. As mentioned in section 4.1 in step 4, after the one-class SVM test, the moving data window is fed into the NN to demonstrate the fault classification and determine the permutation's importance. The permutation score for each variable is calculated using algorithm 3. Using the number of experiments, 25% of the maximum permutation score is defined as a marginal score to analyze the root cause for the detected failure condition. The proposed algorithm testing result shows the accurate root cause analysis for the detected fault condition. For example, as illustrated in Figure 4.11, the IDV 5 result, the root cause for the fault is analyzed as the XMAS22 variable. From Tables 4.3 and 4.4

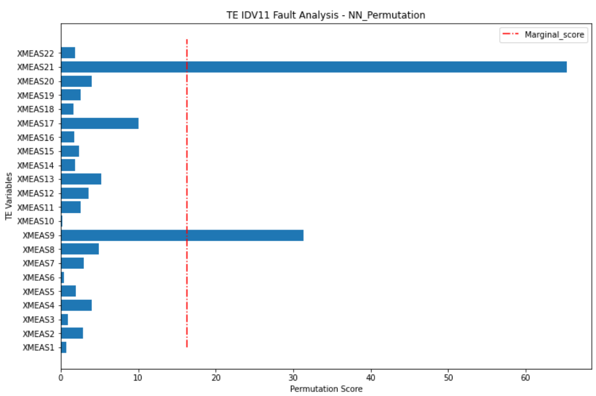
in the supporting information, the IDV 5 fault condition in the TE process system is defined as 'condenser cooling water inlet temperature,' and the analyzed root cause 'XMEAS22' variable correctly predicts the cause variable by the permutation algorithm.

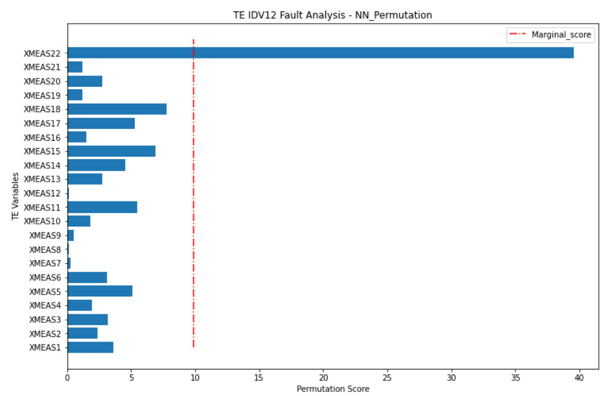












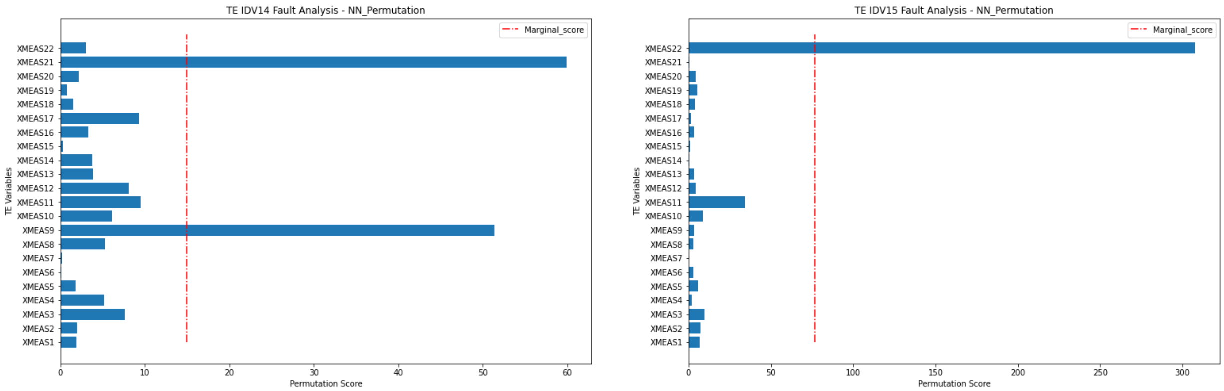


Figure 4.11 TE process fault root cause analysis

The proposed NN permutation algorithm results are compared with recently developed fault diagnosis and root cause analysis methodologies. As shown in Table 4.7, the proposed NN permutation algorithm accurately detect the root cause for each fault condition. Moreover, compared to recently developed transfer entropy and the SDNTE approach, the developed model shows a faster response to diagnose and define the root cause for each fault condition. In Table 4.7, elapsed time for the proposed algorithm defines the time duration to attain the fault diagnosis and root cause variable after the fault detection.

Table 4-7 TE process Diagnosis and root cause analysis comparison

Fault ID	**Fault Description**	**Root Cause / Contribution of variable**							
		Proposed Method		**Rashid et al (2018) [34]**				**Amin et al (2018) [37]**	**He et al (2014) [38]**
		NN-Permutation		**Transfer Entropy**		**SDNTE**		**PCA, $T^{2,}$ and BN**	**RBMCA+Fuzzy-SGD**
		Root Cause Variable	Elapsed Time (s)	Root Cause Variable	Elapsed Time (s)	Root Cause Variable	Elapsed Time (s)	Root cause Variable	Root cause Variable
IDV1	A/C feed ratio, B composition constant	XMEAS16 XMEAS9 XMEAS13 XMEAS7	6.0	Not Tested		XMEAS4			XMEAS21 XMEAS23 XMEAS7
IDV4	Reactor cooling water inlet temperature	XMEAS21 XMEAS9	6.0			XMEAS9			XMEAS9 XMEAS26
IDV5	Condenser cooling water inlet temperature	XMEAS22 XMEAS11	6.1			XMEAS11			XMEAS22 XMEAS11
IDV6	A feed loss	XMEAS1	6.3			XMEAS6			-
IDV11	Reactor cooling water inlet temperature	XMEAS21 XMEAS9	7.1	XMEAS21, XMEAS32	68.9	XMEAS21, XMEAS32	16.2	XMEAS9	XMEAS37 XMEAS9
IDV12	Condenser cooling water inlet temperature	XMEAS22	7.8	XMEAS22	79.9	XMEAS22	18.1	XMEAS11	-
IDV14	Reactor cooling water valve	XMEAS21 XMEAS9	6.3	XMEAS21, XMEAS32	69.7	XMEAS32	17.8	XMEAS9	-
IDV15	Condenser cooling water valve	XMEAS22 XMEAS11	7.1	XMEAS11, XMEAS22, XMEAS33	71.2	XMEAS11, XMEAS33	18.2	XMEAS11	-

- PCA- Principal component analysis, SDNTE - symbolic dynamic-based normalized transfer entropy, RBMCA - Reconstruction based multivariate contribution analysis, and SGD – Signed directed graph.

4.6 Conclusions

In this study, a novel autonomous approach is proposed for online fault detection, diagnosis, and root cause analysis using machine learning approaches that can be integrated with Industry 4.0 concepts. Self-learning and online updating one-class SVM and NN models are used to detect the unknown faults and classify the known fault conditions. Specifically, the NN permutation algorithm has been combined with the input data correlation test for the first time in this work to determine the root cause of the detected abnormal condition. In addition, online updating, a self-adjusting learning rate, and the number of iterations for NN optimization have been proposed to achieve better accuracy in the NN classification. Also, the required computational time is tested and compared. In the proposed algorithm, NN with a smaller number of layers is optimized to shorten the computation time and meet the requirements for online applications.

The proposed model is evaluated using CSTH process data with 5 variables and TE process data with 22 variables. The results confirm the benefit of the algorithm compared with recently developed algorithms. The proposed framework is recommended for Industry 4.0 process system abnormal situation management to diagnose the root cause of faults in real-time. The proposed framework can be further integrated on the Internet of Things (IoT) to analyze process fault conditions in real-time and in a remote monitoring setting. Also, when implementing the proposed Methodology in a large-scale process plant, with the appropriate communication protocol in the IoT, the root cause for a detected fault can be easily isolated by the process subsystems.

Reference

[1] S. Dey and J. A. Stori, "A Bayesian network approach to root cause diagnosis of process variations," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 1, pp. 75–91, 2005, doi: 10.1016/j.ijmachtools.2004.06.018.

[2] A. Alzghoul, B. Backe, M. Löfstrand, A. Byström, and B. Liljedahl, "Comparing a knowledge-based and a data-driven method in querying data streams for system fault detection: A hydraulic drive system application," *Computers in Industry*, vol. 65, no. 8, pp. 1126–1135, Oct. 2014, doi: 10.1016/J.COMPIND.2014.06.003.

[3] C. M. Fan and Y. P. Lu, "A bayesian framework to integrate knowledge-based and data-driven inference tools for reliable yield diagnoses," *Proceedings - Winter Simulation Conference*, pp. 2323–2329, 2008, doi: 10.1109/WSC.2008.4736337.

[4] Y. Cohen and G. Singer, "A smart process controller framework for Industry 4.0 settings," *Journal of Intelligent Manufacturing*, no. 0123456789, 2021, doi: 10.1007/s10845-021-01748-5.

[5] L. Ming and J. Zhao, "Review on chemical process fault detection and diagnosis," *2017 6th International Symposium on Advanced Control of Industrial Processes, AdCONIP 2017*, pp. 457–462, 2017, doi: 10.1109/ADCONIP.2017.7983824.

[6] H. Chen, P. Tiňo, and X. Yao, "Cognitive fault diagnosis in Tennessee Eastman Process using learning in the model space," *Computers and Chemical Engineering*, vol. 67, pp. 33–42, 2014, doi: 10.1016/j.compchemeng.2014.03.015.

[7] R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, "Fault detection and diagnosis in process system using artificial intelligence-based cognitive technique," *Computers and Chemical Engineering*, vol. 134, p. 106697, Mar. 2020, doi: 10.1016/j.compchemeng.2019.106697.

[8] P. Duan, T. Chen, S. L. Shah, and F. Yang, "Methods for Root Cause Diagnosis of Plant-Wide Oscillations," *AIChE Journal*, vol. 60, no. 6, pp. 2019–2034, 2014, doi: 10.1002/aic.14391.

[9] H. Jiang, R. Patwardhan, and S. L. Shah, "Root cause diagnosis of plant-wide oscillations

using the concept of adjacency matrix," *Journal of Process Control*, vol. 19, no. 8, pp. 1347–1354, 2009, doi: 10.1016/j.jprocont.2009.04.013.

[10] T. Yuan and S. J. Qin, "Root cause diagnosis of plant-wide oscillations using Granger causality," *Journal of Process Control*, vol. 24, no. 2, pp. 450–459, 2014, doi: 10.1016/j.jprocont.2013.11.009.

[11] W. Yu and F. Yang, "Detection of causality between process variables based on industrial alarm data using transfer entropy," *Entropy*, vol. 17, no. 8, pp. 5868–5887, 2015, doi: 10.3390/e17085868.

[12] G. Weidl, A. L. Madsen, and S. Israelson, "Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes," *Computers and Chemical Engineering*, vol. 29, no. 9, pp. 1996–2009, 2005, doi: 10.1016/j.compchemeng.2005.05.005.

[13] Q. X. Zhu, Y. Luo, and Y. L. He, "Novel Multiblock Transfer Entropy Based Bayesian Network and Its Application to Root Cause Analysis," *Industrial and Engineering Chemistry Research*, vol. 58, no. 12, pp. 4936–4945, 2019, doi: 10.1021/acs.iecr.8b06392.

[14] H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, "Unsupervised Learning Based On Artificial Neural Network: A Review," *2018 IEEE International Conference on Cyborg and Bionic Systems, CBS 2018*, pp. 322–327, 2019, doi: 10.1109/CBS.2018.8612259.

[15] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Piatt, "Support vector method for novelty detection," *Advances in Neural Information Processing Systems*, pp. 582–588, 2000.

[16] A. Beghi, L. Cecchinato, C. Corazzol, M. Rampazzo, F. Simmini, and G. A. Susto, "A one-class SVM based tool for machine learning novelty detection in HVAC chiller systems," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2014, vol. 19, no. 3, pp. 1953–1958, doi: 10.3182/20140824-6-za-1003.02382.

[17] Y. Xiao, H. Wang, W. Xu, and J. Zhou, "Robust one-class SVM for fault detection," *Chemometrics and Intelligent Laboratory Systems*, vol. 151, pp. 15–25, Feb. 2016, doi: 10.1016/j.chemolab.2015.11.010.

[18] C. Domeniconi and D. Gunopulos, “Incremental support vector machine construction,” *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 589–592, 2001, doi: 10.1109/icdm.2001.989572.

[19] S. Alam, S. K. Sonbhadra, S. Agarwal, and P. Nagabhushan, “One-class support vector classifiers: A survey,” *Knowledge-Based Systems*, vol. 196, p. 105754, 2020, doi: 10.1016/j.knosys.2020.105754.

[20] T. Prayoonpitak and S. Wongsa, “A Robust One -Class Support Vector Machine Using Gaussian -Based Penalty Factor and Its Application to Fault Detection,” *International Journal of Materials, Mechanics and Manufacturing*, vol. 5, no. 3, pp. 146–152, 2017, doi: 10.18178/ijmmm.2017.5.3.307.

[21] R. Chalapathy, A. K. Menon, and S. Chawla, “Anomaly Detection using One-Class Neural Networks,” no. August, pp. 19–23, 2018, [Online]. Available: http://arxiv.org/abs/1802.06360.

[22] J. C. Hoskins and D. M. Himmelblau, “Artificial neural network models of knowledge representation in chemical engineering,” *Computers & Chemical Engineering*, vol. 12, no. 9–10, pp. 881–890, Sep. 1988, doi: 10.1016/0098-1354(88)87015-7.

[23] V. Venkatasubramanian and K. Chan, “A neural network methodology for process fault diagnosis,” *AIChE Journal*, vol. 35, no. 12, pp. 1993–2002, 1989, doi: 10.1002/aic.690351210.

[24] R. Lit, J. H. Olsont, and D. L. Chester, “Dynamic Fault Detection and Diagnosis Using Neural Networks,” *IEEE*, pp. 1169–1174, 1990.

[25] Y. Maki and K. A. Loparo, “A neural-network approach to fault detection and diagnosis in industrial processes,” *IEEE Transactions on Control Systems Technology*, vol. 5, no. 6, pp. 529–541, 1997, doi: 10.1109/87.641399.

[26] S. H. Yang, B. H. Chen, and X. Z. Wang, “Neural network based fault diagnosis using unmeasurable inputs,” *Engineering Applications of Artificial Intelligence*, vol. 13, no. 3, pp. 345–356, 2000, doi: 10.1016/S0952-1976(00)00005-1.

[27] R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, “Fault detection and

diagnosis in process system using artificial intelligence-based cognitive technique," *Computers & Chemical Engineering*, vol. 134, p. 106697, Mar. 2020, doi: 10.1016/J.COMPCHEMENG.2019.106697.

[28] L. BREIMAN, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001, doi: 10.1201/9780429469275-8.

[29] A. H. Sung, "Ranking importance of input parameters of neural networks," *Expert Systems with Applications*, vol. 15, no. 3–4, pp. 405–411, 1998, doi: 10.1016/s0957-4174(98)00041-4.

[30] J. de Oña and C. Garrido, "Extracting the contribution of independent variables in neural network models: A new approach to handle instability," *Neural Computing and Applications*, vol. 25, no. 3–4, pp. 859–869, 2014, doi: 10.1007/s00521-014-1573-5.

[31] G. Hooker and L. Mentch, "Please Stop Permuting Features: An Explanation and Alternatives," May 2019, Accessed: Sep. 04, 2021. [Online]. Available: https://arxiv.org/abs/1905.03151.

[32] C. Strobl, A. L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, pp. 1–11, 2008, doi: 10.1186/1471-2105-9-307.

[33] N. F. Thornhill, S. C. Patwardhan, and S. L. Shah, "A continuous stirred tank heater simulation model with applications," *Journal of Process Control*, vol. 18, no. 3–4, pp. 347–360, 2008, doi: 10.1016/j.jprocont.2007.07.006.

[34] B. Rashidi, D. S. Singh, and Q. Zhao, "Data-driven root-cause fault diagnosis for multivariate non-linear processes," *Control Engineering Practice*, vol. 70, pp. 134–147, Jan. 2018, doi: 10.1016/j.conengprac.2017.09.021.

[35] J. J. Downs and E. F. Vogel, "A Plant-wide Industrial Problem Process," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993, doi: 10.1016/0098-1354(93)80018-I.

[36] I. A. Udugama, K. V. Gernaey, M. A. Taube, and C. Bayer, "A novel use for an old problem: The Tennessee Eastman challenge process as an activating teaching tool," *Education for*

Chemical Engineers, vol. 30, pp. 20–31, 2020, doi: 10.1016/j.ece.2019.09.002.

[37] M. T. Amin, S. Imtiaz, and F. Khan, "Process system fault detection and diagnosis using a hybrid technique," *Chemical Engineering Science*, vol. 189, pp. 191–211, Nov. 2018, doi: 10.1016/j.ces.2018.05.045.

[38] B. He, T. Chen, and X. Yang, "Root cause analysis in multivariate statistical process monitoring: Integrating reconstruction-based multivariate contribution analysis with fuzzy-signed directed graphs," *Computers and Chemical Engineering*, vol. 64, pp. 167–177, 2014, doi: 10.1016/j.compchemeng.2014.02.014.

Chapter 5 Process Fault Prognosis Using Deep Learning

Preface

A version of this manuscript has been published in the process safety environmental protection Journal. In this chapter, an early fault detection algorithm has been developed using CNN-LSTM integrated methodology. Also, the developed methodology used one-class SVM to analyze the fault condition from predicted synthetic data. This chapter contributes to the targeted 4th objective of the thesis defined in chapter 1.

Abstract

Early fault detection and fault prognosis are crucial functions to ensure safe process operations. Fault prognosis can detect and isolate early developing faults as well as predict fault propagation. To promptly detect potential faults in process systems, it is important to examine the fault symptoms as early as possible. In recent years, fault prognosis approaches have led to the remaining useful life prediction. Therefore, in a process system, advancing prognosis approaches will be beneficial for early fault detection in terms of process safety and to predict the remaining useful life, targeting the system's reliability. In data-driven models, early fault detection is regarded as a time-dependent sequence learning problem; the future data sequence is predicted using the previous data pattern. Studying recent years' research shows that a recurrent neural network (RNN) can solve the sequence learning problem. This work proposes an early potential fault detection approach by examining the fault symptoms in multivariate complex process systems. The proposed model has been developed using the Convolutional Neural Network (CNN)- Long Short-Term Memory (LSTM) approach to forecast the system parameters for future sampling windows' recognition and an unsupervised One-class-SVM used for fault symptoms' detection using

forecasted data window. The performance of the proposed method is assessed using Tennessee Eastman process time-series data. The results confirm that the proposed method effectively detects potential fault conditions in multivariate dynamic systems by detecting the fault symptoms early as possible.

5.1 Introduction

In process systems, when a fault occurs in a machine or equipment, it may take a significant time to detect it. However, when the fault occurs, it gradually leads to performance degradation until a failure occurs. A failure in a machine has the potential to seriously affect the safety and reliability of the plant. It is important to detect and identify a fault as early as possible and investigate its root causes, to prevent system failure. With the advent of Industry 4.0, the complex combinations of process, sensor, control, and information technology in process plants warrant the need to develop and apply autonomous failure prognosis and early detection techniques.

Early fault detection and condition monitoring techniques can be divided into three categories: model-based, data-based, and expert knowledge-based [1]. The model-based approaches are developed using the laws of physics of the system, and they give precise results when the physical system is described adequately with the right model assumptions [2]. However, due to the complexity and unknown dynamics of process systems, it is difficult to develop such mechanistic models. To overcome this issue, expert systems have been developed based on expert knowledge of the system and a failure data history. Information used in an expert system generally includes a knowledge base of expert opinions about the process accumulated over a long time and a basis of rules for applying this knowledge to a particular problem [3]. The rules are formulated based on human expertise and heuristic algorithms. Li et al. have reported that accurate physical models or expert models are not available in most of the process systems [4].

The data-driven approaches are traditionally used as feature extraction methods combined with statistical analysis for detecting a deviation in-process data [5]. However, in recent years, due to the requirements of autonomous operation in Industry 4.0, shallow and deep learning algorithms are more commonly used for data-driven analysis [6].

Due to the complexity in process systems, industrial data are highly correlated and multivariate. Therefore, it is necessary to develop prognosis models to handle multivariate data. In recent years deep neural network approaches have been used in multivariate data analysis. Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Autoencoder (AE), and Restricted Boltzmann Machine (RBM) approaches are widely applied in recent deep learning methods for fault diagnosis and prognosis.

In recent research, for detecting and diagnosing a fault, CNN approaches are used for feature extraction from 1-D sensor data [7]. Cheng et al., Liu et al., and Hoang et al. have proposed a similar 1-D CNN approach to extract features from raw data and integrate the CNN model with a fully connected neural network for fault classification and diagnosis [7]–[9]. Liu et al., and Hoang et al., have also suggested the use of an RNN based model to predict the fault condition using time-series process data [8], [9]. Zhang et al., and Wang et al., have proposed an RBM-based fault prognosis approach for chemical process systems [10], [11]. Lin and Tao, and Verstraete et al., have developed an autoencoder-based fault prognosis and health monitoring approach [12], [13]. DNN approaches are applied as supervised learning models to predict the known fault condition in these cited works.

To the best of our knowledge, most of the deep learning prognosis models forecast the fault condition based on available supervised data. However, when developing a deep learning methodology for fault prognosis and fault prediction, the following drawbacks and challenges are considered:

1. It is difficult to detect a fault condition promptly using unsupervised data.
2. Predicting the fault condition using offline risk calculation based on a normal dataset is not accurate.
3. Feature extracting and forecasting the multivariate process data to call attention to a system fault early pose problems.

To overcome the above-mentioned drawbacks, a multistep multivariate CNN-LSTM deep neural network interface with a one-class support vector machine (OC-SVM) is proposed in this article. As shown in Figure 5.1, in the proposed model, the CNN LSTM model uses CNN to extract the features of the input time data and uses LSTM to forecast the synthetic process data from the previous data samples. Finally, OC SVM is used to predict the fault symptoms using the forecasted data sample from the CNN LSTM model.

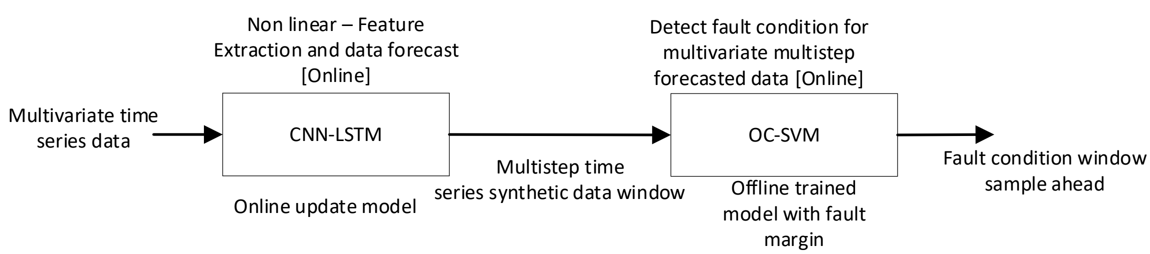


Figure 5.1 Model integration in the proposed method.

This section is organized as follows: Section 5.1 describes the fundamental objective of the proposed work, and section 5.2 reviews the background information. The proposed hybrid network for fault prognosis is described in section 5.3. Applications of the proposed method and test results are presented in sections 5.4 and 5.5, respectively. Some concluding remarks are presented in section 5.6.

5.2 Background and Related Work

Most of the recent fault prognosis methodologies have been developed based on data-based models. Deep learning has become one of the most important methods in the field of machine learning and has been recently used to forecast future data patterns. In recent years, as an emerging artificial machine learning method, deep learning approaches have been applied in process plant fault prediction and prognosis because of their powerful pattern learning and function mapping capabilities.

The use of ANN in process system fault detection and the diagnosis was initially proposed by Hoskin and Himmelblau [14]. Later, NN models were applied in many process systems for fault detection and diagnosis. Sorsa and Koivo proposed an artificial neural network-based methodology for the heat exchanger of a stirred tank reactor system to detect and classify 10 fault situations [15]. However, there are large quantities of temporal information available in a process system that contribute to fault prediction but cannot be extracted by ANN. A deep learning approach of the recurrent neural network (RNN) has a strong ability to capture hidden correlations. RNN has been applied to process big data in applications for image captioning, voice conversion, and natural language processing, and also for dealing with process faults. However, the original RNN has the problem of vanishing gradients because of the later node perception of the previous nodes decreases.

Long short-term networks were initially proposed by Hochreiter and Schmidhuber to solve the above-mentioned problem [16]. Compared to the conventional RNN, an LSTM network performs well in extracting the features of time series for a long-time span. Zhao et al. proposed a fault diagnosis method based on the LSTM network; the novel method can directly classify the raw process data without specific feature extraction and classifier design [17]. It is also able to adaptively learn dynamic information from raw data. Yue et al. proposed a CNN-LSTM model for industrial data for deep learning and transfer learning; in the approach, CNN extracts features automatically, and LSTM analyzes the new feature sequences from CNN [18]. Zheng et al. use a hybrid CNN-LSTM network for fault detection and diagnosis, and compared to the result with the LSTM method, this approach improves the accuracy of fault detection [19]. Park et al. proposed the LSTM-Autoencoder approach for fault detection and diagnosis, where an autoencoder is used to detect a rare event, and LSTM is used for classifying the different types of faults [20]. Xia et al. proposed an ensemble framework based on convolutional bidirectional LSTM with multiple time windows for accurately predicting RUL [21]. Han et al. proposed a fault diagnosis model based on the optimized LSTM network [22]. It showed that the number of hidden layer nodes in the LSTM network has a significant influence on the diagnosis result. Liu et al. proposed a novel model named LSS, which combines the advantages of an LSTM network with statistical process analysis to predict the fault condition [8].

The most commonly used deep learning algorithms in fault prognosis are CNN, recurrent neural network (RNN), and autoencoder network [23]. CNN architecture is used for feature extraction, and RNN is used for forecasting. In this section, the proposed CNN-LSTM and OC-SVM methodologies and background information are discussed.

5.2.1 CNN Approach

The convolutional neural network is a feedforward neural network used for data preprocessing and feature extraction. The CNN approaches have significant success in computer vision, natural language processes, and speech recognition applications [24].

A CNN structures are classified as:1-D CNN, 2-D CNN, and 3D CNN. 1-D CNN is focused on sequential data feature extraction, 2-D is used for image and text recognition, and 3-D CNN is mostly used for medical image processing [25]. To extract the features from time series sequential sensor data, the 1-D CNN approach is proposed in this article, and background information is discussed as follows.

The complete process of 1D CNN is described in Figure 5.2. Multidimensional time series sensor raw data will be convolute as a feature mapping. The extracted feature dimension after the convolution filter will be E*1; E depends on sensor data dimension, the size of the filter, and convolutional step length. In Figure 5.2, red and blue indicate the sample filter dimension. Suppose the number of filters is F, then the extracted feature dimension will be E * F.

Yue et al., and Pan et al., have reported that general CNN models include convolutional layers and pooling layers [26], [27]. Convolutional layers use a filter matrix for feature extraction and the pooling layer for feature dimensionality reduction, compressing the amount of data and parameters and reducing overfitting. According to Li et al., when data pass through the pooling layer, noticeable useful information can be filtered to some extent. In the proposed method, CNN is used only for feature mapping [4]. Therefore, the pooling layer is not implemented in the proposed methodology, where raw feature dimension is used for synthetic data prediction by using the LSTM network.

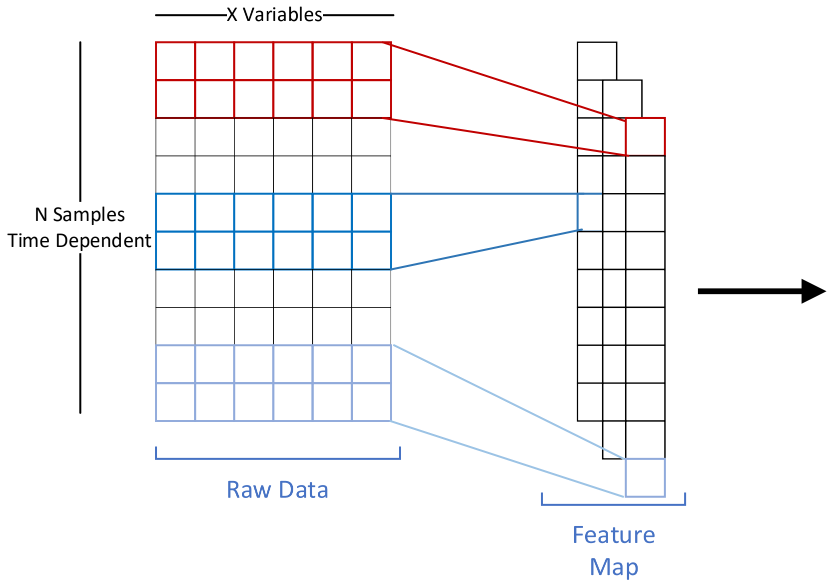


Figure 5.2: CNN network (Adapted from Li et al. (2020))

The mathematical expression of 1-D CNN approaches is presented:

One-dimensional sequential data input:

$x = [x_1, x_2, ..., x_N]$, where N is data length of sensor data sequences.

In the convolutional layer operation, the convolutional filter slides over each sample and executes the convolutional operation. The output dimension of the convolutional layer depends on the number of filters, and each sample is converted to a feature map, calculated as follows:

$$M_i = a(w^T X^{(i:i+N-1)} + b) \tag{5-1}$$

where $'w^T'$ denotes the transposition of filter kernel matrix 'w', 'b' denotes bias, $'a'$ denotes nonlinear activation and $X^{(i:i+N-1)}$ represents N observations from i^{th} time step to $(i+N-1)^{th}$ time step.

The time step window $X^{(i:i+N-1)}$ is calculated as follows:

$$X^{(i:i+N-1)} = X^{i} \oplus X^{i-1} \oplus X^{i-2} \oplus \dots \oplus X^{(i+N-1)} \tag{5-2}$$

$\oplus$ denotes linking concatenation of each sample into a longer embedding.

Sliding the filter window from the first point to the last point in the sample data will obtain the feature map of the filter, which is expressed as:

$$M_j = [M_1, M_2, \dots, M_{n-N+1}] \tag{5-3}$$

where ‘j’ is the jth filter kernel; however, in 1D CNN, multiple filter kernels can be applied in the convolutional layer with different filter lengths.

A large filter size generally leads to good accuracy [4]. However, due to the computational burden, it is difficult to agree on an appropriate filter size.

5.2.2 LSTM Approach

The recurrent Neural Network (RNN) is a commonly used neural network approach for time series prediction. RNN predicts time series data by allowing the state to circulate via its feedback connection. However, with long-time series predictions, RNNs typically suffer from the vanishing gradient problem. To overcome this issue, LSTM was introduced. LSTM is a modified structure of RNN that adds memory cells into hidden layers to control the memory information of the time series data [16].

The LSTM has three primary multiplication gate structures: input, forget, and output gates. LSTM uses multiplicative gates to store and access data over a long period of time. This eliminates the problem of a vanishing gradient. Input and forget gates are designed for controlling the state of the memory cell. Figure 5.3 shows the architecture of the LSTM block.

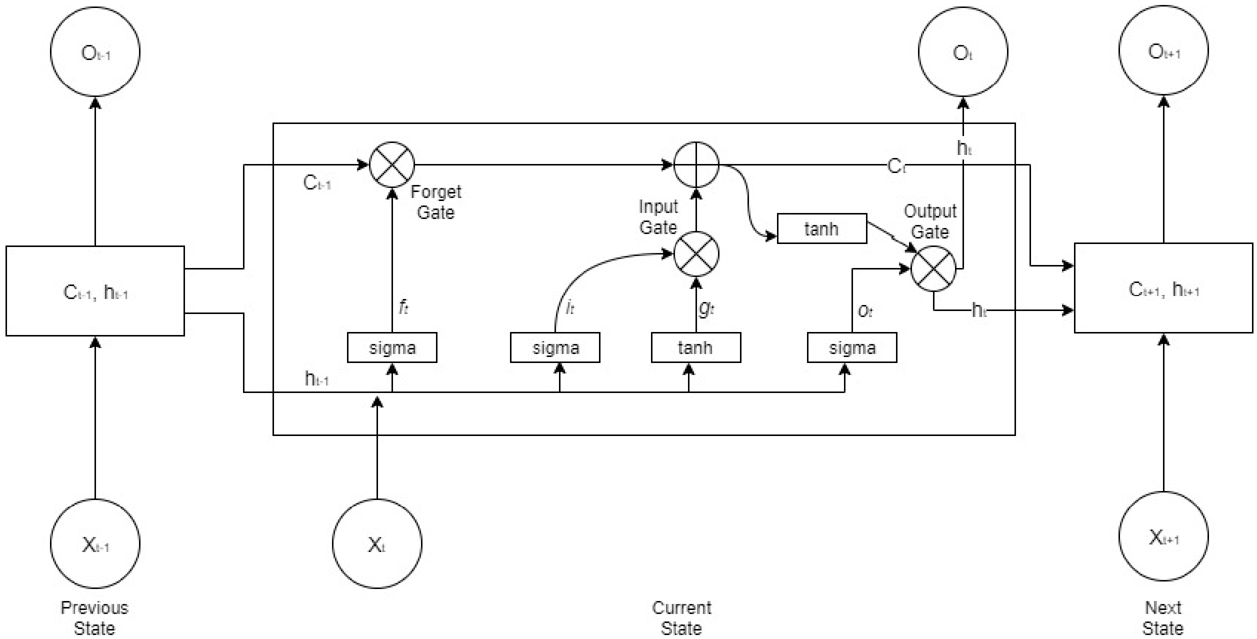


Figure 5.3: LSTM unit (Adapted from Xia et al. (2020)) [21]

Brief definitions of each gate are described below:

The Forget gate indicates how much memory of the last moment's cell can be saved. The output gate of the LSTM is designed for controlling how much information is output for cell status. The gate uses the output of the previous time step (h_{t-1}) and the input of the current time (X_t). The output of the gate makes the decision to clear the data by logic 0 and save the data by logic 1.

$$f_t = \sigma(W_f . x_t + W_f . h_{t-1} + b_f) \quad (5\text{-}4)$$

where $'W_f'$ is an input weight and $'b_f'$ is a bias weight to obtain the forget gate $'f_t'$.

The input gate determines how much input of the current moment can be saved to the cell state and controls the proportion of fusion of historical and current stimuli. Two phases comprise an input gate; i_t decides which values are updated and g_t generates a new value for the cell states:

$$i_t = \sigma(W_i . x_t + W_i . h_{t-1} + b_i) \quad (5\text{-}5)$$

$$g_t = tanh(W_g . x_t + W_g . h_{t-1} + b_g \quad (5\text{-}6)$$

where $'W_i' \ and \ 'W_g'$ are input weights and $'b_g' \ and \ 'b_f'$ are bias weights.

Hence, the updated value of the current cell state (C_t) is a combination of the forget gate and the input gate:

$$C_t = f_t \otimes C_{t-1} + i_t . g_t) \quad (5\text{-}7)$$

In Equation 5.7, values $'f_t'$ lie between 0 and 1. An element-wise multiplication of f_t by C_{t-1} (previous cell state) determines which elements need to be saved in the memory cell.

The output gate controls which parts of the current state should be read and sent to the output:

$$O_t = \sigma(W_o . x_t + W_o . h_{t-1} + b_o) \quad (5\text{-}8)$$

$$h_t = O_t \otimes tanh(C_t) \quad (5\text{-}9)$$

5.2.3 CNN-LSTM Approach

A hybrid CNN-LSTM model is constructed by combining CNN with LSTM to improve data forecasting. The proposed forecasting CNN-LSTM models multivariate time series data as inputs and multistep time series data as outputs. Figure 5.4 shows the proposed CNN-LSTM block diagram.

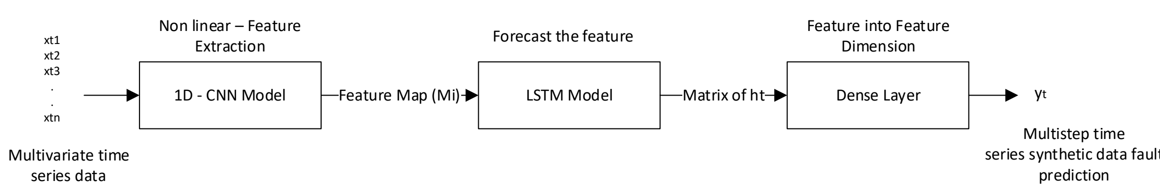


Figure 5.4: General CNN-LSTM model with a dense layer.

CNN and LSTM approaches are discussed in sections 5.2.1 and 5.2.2. However, when integrating the models, a 1D-CNN output M_i feature sample, as mentioned in Equations 5.1 and 5.4, is fed into the LSTM model current state input x_t, as mentioned in Equations 5.4 and 5.8. Finally, in the CNN-LSTM approach, the dense layer is used after the LSTM to transform the synthetic feature data by learning the features from all the combinations of the features of the previous layers. The dense layer is similar to the shallow neural network, which is used to do the matrix multiplication of the input vector from LSTM sequential layers with a weight matrix.

$$y = a(w_f . h + b) \tag{5-10}$$

In Equation 5.10, a is a nonlinear activation function, w_f is a dense layer weight matrix, and b is a bias vector. Finally, h is an input vector for the layer, which is an output matrix of h_t from the LSTM sequential network, as mentioned in Equation 5.9. Also, the dense layer optimizes the use of the supervisory learning approach, where the previous state of input features is supervised by current state inputs to forecast the synthetic feature data for the next state.

5.2.4 One-Class Support Vector Machine

A one-class support vector machine is an unsupervised anomaly detection learning algorithm that can be trained using normal data. When testing data samples deviate from normal data, sample points will be detected as anomalies. In general, OC-SVM is used to separate the data of one specific class. However, in this approach, OC-SVM is used to detect the fault condition early, using forecasted synthetic data from the CNN-LSTM model.

Schölkopf proposed the OC- SVM to separate all the data points from their origin and maximize the distance from this hyperplane to the origin [28]. The result is that the binary function captures the region in the input space from the probability density of the real-time data. Therefore, the model gives +1within a region and -1 for the region outlier.

$$f_x = \begin{cases} +1\ if\ x\ is\ in\ the\ region \\ -1\ if\ x\ is\ not\ in\ the\ region \end{cases}$$

Let x be a training input sample that fits into the one-class classifier. To separate the data set from the origin, OC-SVM needs to solve the following quadratic optimization problem:

$$min\frac{1}{2}\|w\|^2 + \frac{1}{vl}\sum_{i=1}^{l}\varepsilon_i - \rho \tag{5-11}$$

where ‘w’ and $'\rho'$ hyperplane parameters have maximal distance from the origin. The $'\varepsilon_i'$ is introduced to protect the SVM classifier from noisy overfitting data. Therefore, it allows some data points to lie within the margin. Finally, $'v'$ tunes the trade-off between the classification error on the training data and the margin maximization in OC-, and $'v'$ lies between 0 and 1.

This is subject to:

$$\left(w.\varPhi(x_i)\right) \geq \rho - \varepsilon_i \quad i = 1,2,\ldots l\ \varepsilon_i \geq 0 \tag{5-12}$$

where $'\Phi'$ is a nonlinear function,

Once $'w'\ and\ '\rho'$ solve this problem, then the one-class SVM decision function is defined as follows to use in the real-time testing:

$$f_{(x)} = sign\ (w.\Phi(x_i) - \rho) \tag{5-13}$$

One of the advantages of using one-class SVM is that the nonlinear decision boundary can be made to recognize the outlier. Therefore, due to the nonlinearity in fault detection and classification, one-class SVM has been commonly used in recent years.

5.3 Hybrid Network for Fault Prognosis

This section describes the framework of the proposed model. In the proposed methodology, the CNN-LSTM hybrid approach is used for feature extraction and forecasting time-dependent samples. Subsequently, OC- SVM is used as unsupervised anomaly detection to classify the faulty and nonfaulty conditions from forecasted data samples.

The complete proposed framework is shown in Figure 5.5. Initially, an OC-SVM model was trained using nonfaulty data, and the fault margin level was obtained by counting outliers in offline conditions. This margin is used as a predefined margin to classify the fault condition.

Multivariable time-dependent process data samples are framed by sliding sampling windows. The sampled data frame is fed into CNN for feature extraction. Due to the nonlinearity data set conversion, the ReLU activation function is used in the model. The output feature map from the CNN then inputs to the LSTM sequence network to forecast the features for the next sampling window. In the initial stage of the model development, several experiments are run to find the hyperparameters in the model to develop an accurate model to forecast the synthetic data samples.

OC- SVM has been developed to detect the unknown fault condition by detecting the abnormalities from forecasted synthetic feature samples. As mentioned in section 5.2.3, from the OC- SVM model, data sample observation for each sample, +1, indicates a normal operating condition, and -1 indicates an abnormal condition. Finally, the number of outliers in each window is counted and tested with a predefined fault margin. If the number of outliers exceeds the marginal level, the tested window is predicted to be in an abnormal condition. Otherwise, it is defined as in a normal operating condition.

In the proposed model, OC-SVM trains in offline conditions and defines the fault margin using the number of anomalies. Further, the sliding sampling window and CNN-LSTM approaches update the online process and forecast future time-dependent synthetic feature samples. Finally, the OC-SVM in the online model detects the abnormal behavior as early as possible using the synthetic samples obtained from the CNN-LSTM model. In a real-time system implementation, forecasting time will depend on model input sampling time and data processing time. Input sampling time highly depends on the frequency of sensor output data, and process time is dependent on the computation time of the CNN - LSTM model.

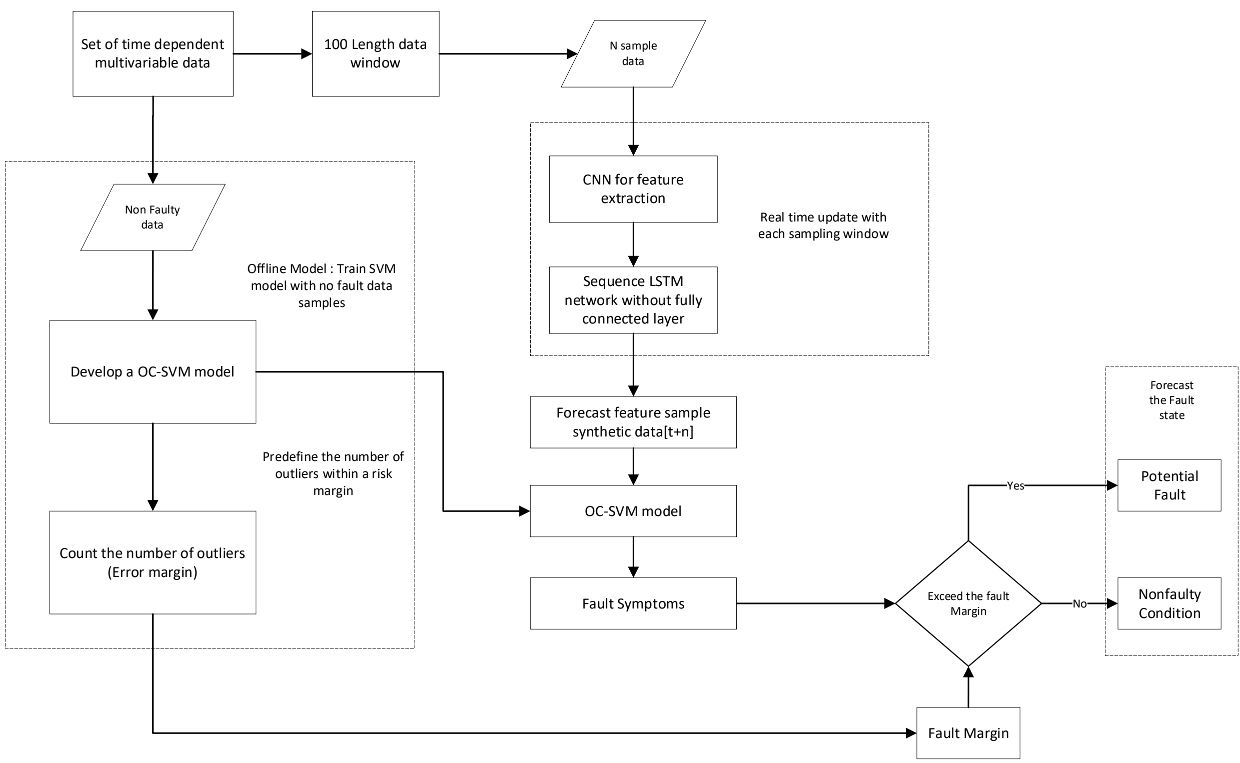


Figure 5.5: Framework of fault prognosis model.

5.3.1 Proposed CNN LSTM – One Class SVM Model Architecture.

The CNN LSTM sequence model architecture and the one-class SVM optimizer are described in this section. When applying the model online, preprocessed window sample data are directly fed into the one-class SVM to predict the actual fault condition. Subsequently, the sample window predicts the fault condition for forecasted synthetic data from CNN-LSTM by using the actual fault conditions and forecasted fault conditions model optimized in the initial stage of the design. The complete model architecture is illustrated in Figure 5.6.

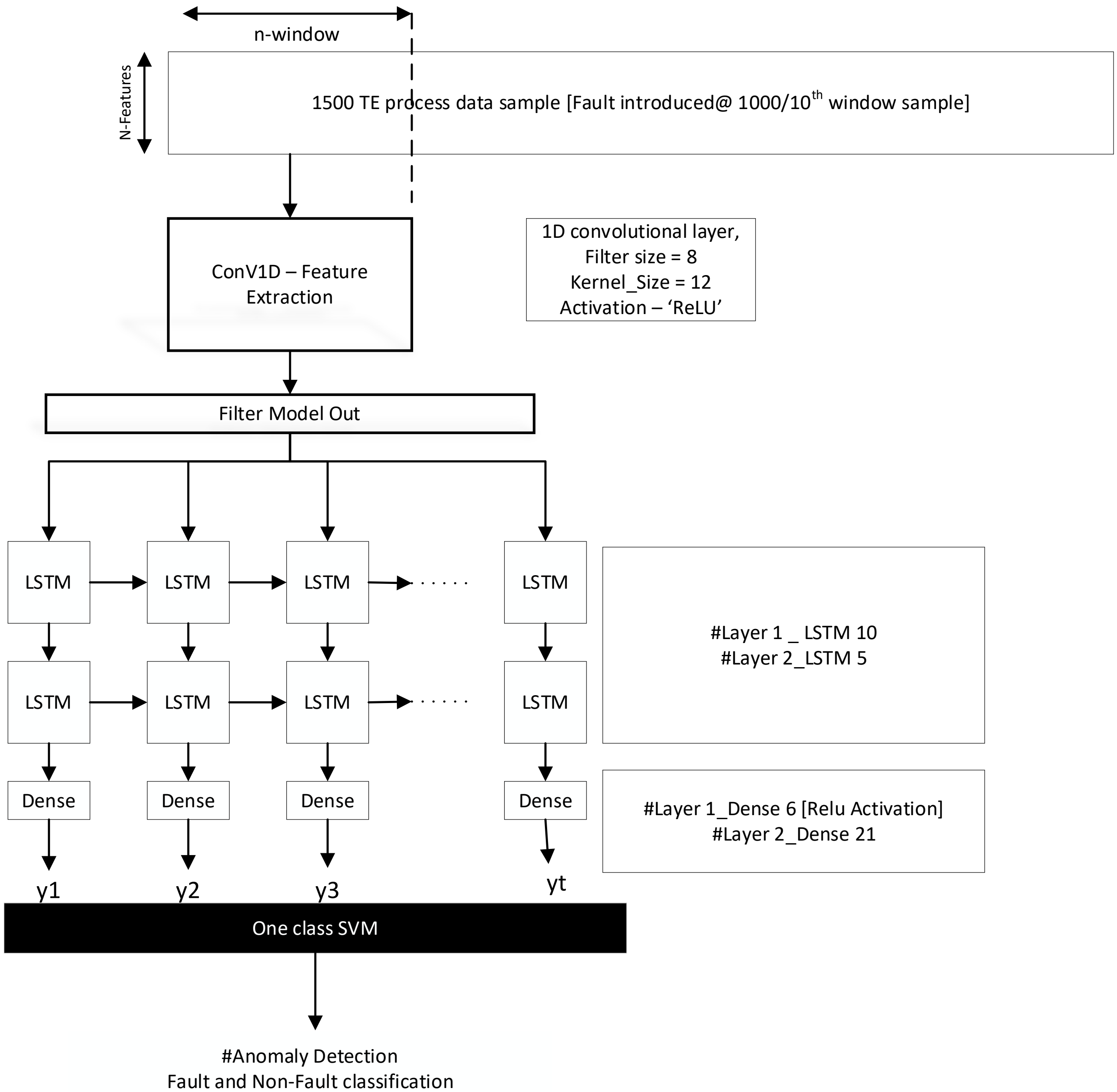


Figure 5.6: CNN LSTM – OC- SVM model for prognosis.

In the first stage, the standard CNN model is trained with stochastic gradient descent methods. 1D CNN model architecture is developed with 8 filters and 12 kernels. A two-layer LSTM sequence model architecture is developed with 10 neurons in layer 1 and 5 neurons in layer 2. A two-layer dense network is developed in the model with 6 neurons in layer 1 and ‘n’ neurons in layer 2 to obtain the ‘n’ number of synthetic feature samples.

Finally, OC-SVM is modelled using the Gamma function, and the radial basis function kernel with an 8e-4 learning rate is used to develop the model.

5.3.2 Proposed Model Algorithm

The proposed algorithm is developed to optimize the CNN-LSTM and the OC-SVM models in offline conditions using nonfaulty data. Initially, non-faulty data samples are standardized to zero median and unit variance. Pre-processed data are fed into the Phase I CNN-LSTM algorithm to forecast the data. As mentioned in Equations 5.1 and 5.2, the ReLU activation function is used in the model to operate with nonlinear data. To optimize the w and b parameters, the 'Adam optimizer' is used in the developed CNN model. Similarly, as described in Equation 5.10, LSTM model parameters w_f *and* b are optimized and updated iteratively by the 'Adam' optimizer to forecast the next sampling window of the synthetic data sample. The learning rate of the ‘Adam’ optimizer and the number of learning iterations of the proposed model are evaluated by running the model with ranges of learning rates and the number of epochs. Mean squared error (MSE) is used to evaluate the model’s performance.

$$mse = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widetilde{y_i})^2 \tag{5-14}$$

The number of samples in a window is denoted by ‘n’. The real data sample and forecasted data samples are denoted by y_i and $\widetilde{y_i}$.

In the algorithm's phase II, OC-SVM is trained in offline conditions using the 300-pre-processed data samples with 100 data samples in a window. To train OC-SVM based on the decision function as mentioned in Equation 5.13, $w\ and\ \rho$ parameters are tuned and determine the decision score during the algorithm training. In the training process, each sample decision score is obtained using the model, and if the decision score is -1, the sample is identified as an anomaly; the total number of anomalies is counted in each window. Also, the fault margin is evaluated using the calculated number of anomalies from the OC-SVM.

$$Fault\ Margin = \delta * max\ (anomaly_{count}) \tag{5-15}$$

δ is a constant to set the noise margin and the value is determined by the model test.

As described in the Online/Realtime fault forecast algorithm, real-time pre-processed data samples using 100 sampling windows are fed into the CNN-LSTM model to obtain the forecasted synthetic data. Subsequently, forecasted data are fed into the trained OC-SVM to determine the number of anomalies in each window. When the number of anomalies exceeds the fault margin, this is identified as a fault symptom and further tested with the next sampling window.

Algorithm – CNN-LSTM one-class SVM based fault forecasting

Offline train model

Input Pre-processed system real data,' n' time-dependent samples window

Phase I: CNN - LSTM training and parameter optimization to predict the synthetic feature data samples, learn from the previous data pattern.

1 Train CNN with 8 filters and 12 kernels, ReLU activation function, and train LSTM using previous data patterns and current data patterns.

2 Determine the hyperparameters of CNN and LSTM

3 **for** Iteration number N

4 | Optimize w and b (Equations 5.1 and 5.2) to tune CNN parameters

5 | Accumulate the feature samples (Equation 5.3)

    Optimize the w and b in LSTM gates.
    Optimize the Dense layers w and b (in Equation 5.10) based on LSTM outputs (Equations 5.8 and 5.9).
    Accumulate the forecasted synthetic samples.
end
Optimize the learning rate and Number of iterations using MSE (Equation 5.14) and develop the model.
Use the developed model to forecast the ‘n’ sampling window from the past data.
Output #Features x n sample forecasted synthetic data samples.

Phase II – Unsupervised fault learning from phase I forecasted data.

Initialize the one-class SVM parameters
for (1: (#samples))
    define f_x, using Equation 5.13 based on optimized $w\ and\ \rho$ from the trained model.
    if f_x = -1, determine the sample as an anomaly.
    Accumulate the #of anomalies in the window.
end
Run the validated real nonfault pre-process data to validate the decision function.
Set the #of anomalies margin, using validation data, and fault margin, using Equation 5.15.
Output unsupervised anomaly detection model.

Online/Real-time fault forecast

Input Forecasted n sample window data from phase I algorithm
for (1: length_window)
    Compute the decision score for each sample: g(n)
    if (g(n) == -1) then
        Anomaly_count = Anomaly_count+1
    else continue the process
end
Output Set of decision score/anomaly count for each windowed sample.

5.4 Application of the Proposed Model

The application of the CNN LSTM-OC SVM is tested on the benchmark Tennessee Eastman process data. The proposed model's fault prognosis performance is compared with the recently developed Hidden Markov Model by Galagedarage [29].

Downs and Vogel (1993) proposed the TE process that provides realistic industrial process control and a monitoring model [30]. As shown in Figure 5.7, the TE process consists of the reactor, condenser, compressor, separator, and stripper. The gaseous reactants A, C, D, and E, and the inert B are fed to the reactor where the liquid products G and H are formed. The species F is a by-product of the process.

Out of 53 total measured variables in the TE process, 22 variables are continuous process measurement, 19 variables are composition measurement, and the remaining 12 variables are manipulated variables. Only the continuous variables are used for training and testing the proposed models.

There are 21 different process faults that can be simulated in the TE process, as summarized in Table 5.1. Faults IDV 1 to 8 are related to step-change in the related variables; Faults IDV 9 to 12 are related to random variation of some variables. IDV 13 involves slow drift in reaction kinetics, and Faults IDV 14, 15, and 21 are related to sticky valves. Faults 16 to 20 are unknown faults; the causes of the faults are unknown.

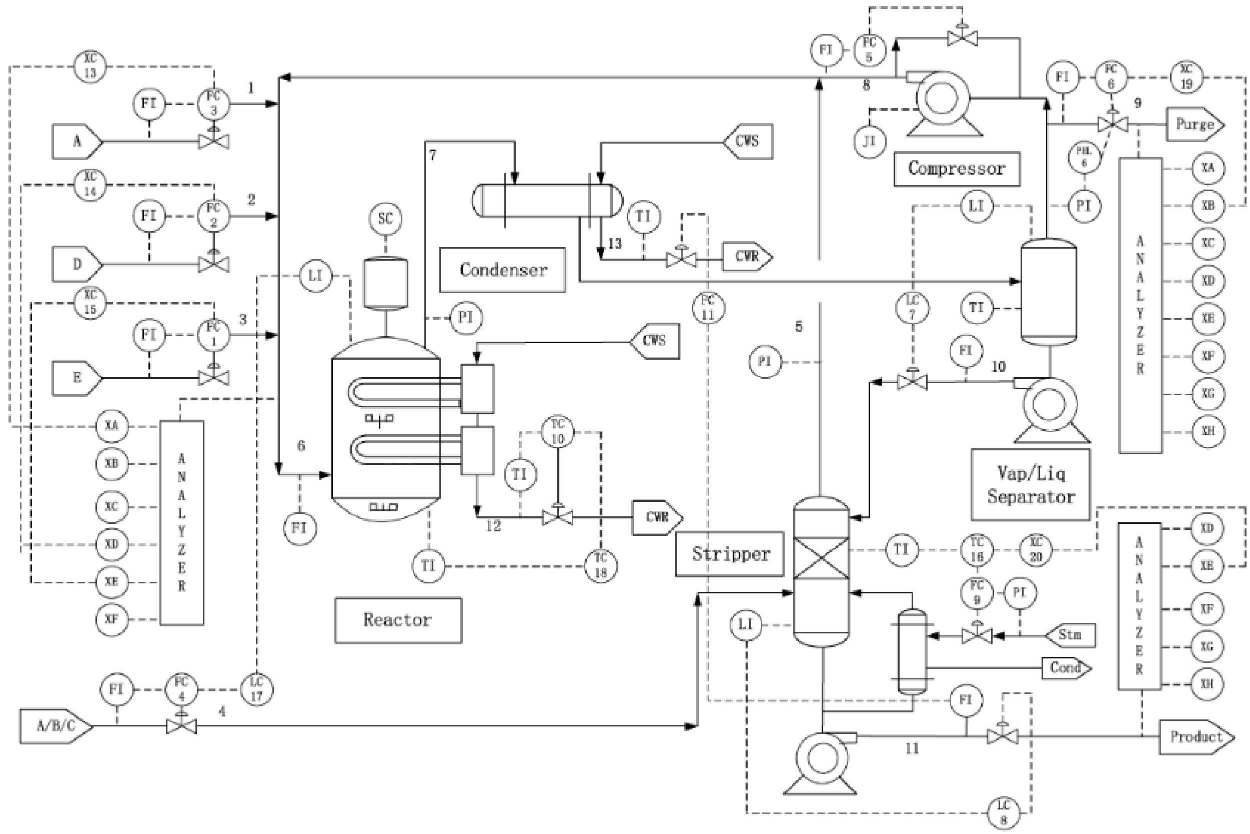


Figure 5.7: TE process flow (Downs and Vogel, 1993) *[30]*

Table 5-1 Tennessee Eastman process faults. (Downs and Vogel, 1993) [30].

Fault ID	**Description**	**Type**
IDV1	A/C feed ratio, B composition constant (Stream 4)[6]	Step
IDV2	B composition, A/C ratio constant (Stream 4)	Step
IDV3	D feed temperature (Stream 2)	Step
IDV4	Reactor cooling water inlet temperature[6]	Step
IDV5	Condenser cooling water inlet temperature[6]	Step
IDV6	A feed loss (Stream 1) [6]	Step
IDV7	C header pressure loss - reduced availability (Stream 4)	Step
IDV8	A, B, C feed composition (Stream 4)	Random variation
IDV9	D feed temperature (Stream 2)	Random variation

[6] Fault condition that are used to test the proposed algorithm.

IDV10	C feed temperature (Stream 4)	Random variation
IDV11	Reactor cooling water inlet temperature[6]	Random variation
IDV12	Condenser cooling water inlet temperature[6]	Random variation
IDV13	Reaction kinetics	Slow drift
IDV14	Reactor cooling water valve[6]	Sticking
IDV15	Condenser cooling water valve[6]	Sticking
IDV16	Unknown	
IDV17	Unknown	
IDV18	Unknown	
IDV19	Unknown	
IDV20	Unknown	
IDV21	The valve for Stream 4 was fixed in the steady-state position	Constant position

5.4.1 Data Sampling and Preprocessing

For each fault condition, 1500 samples are generated with 1000 samples under normal operation, the remaining 500 being data with the faulty condition. As shown in Figure 5.8, 300 Nonfaulty samples are standardized to zero median and unit variance in offline conditions. In the online process, moving window samples are standardized based on the same median and standard deviation value calculated in offline conditions.

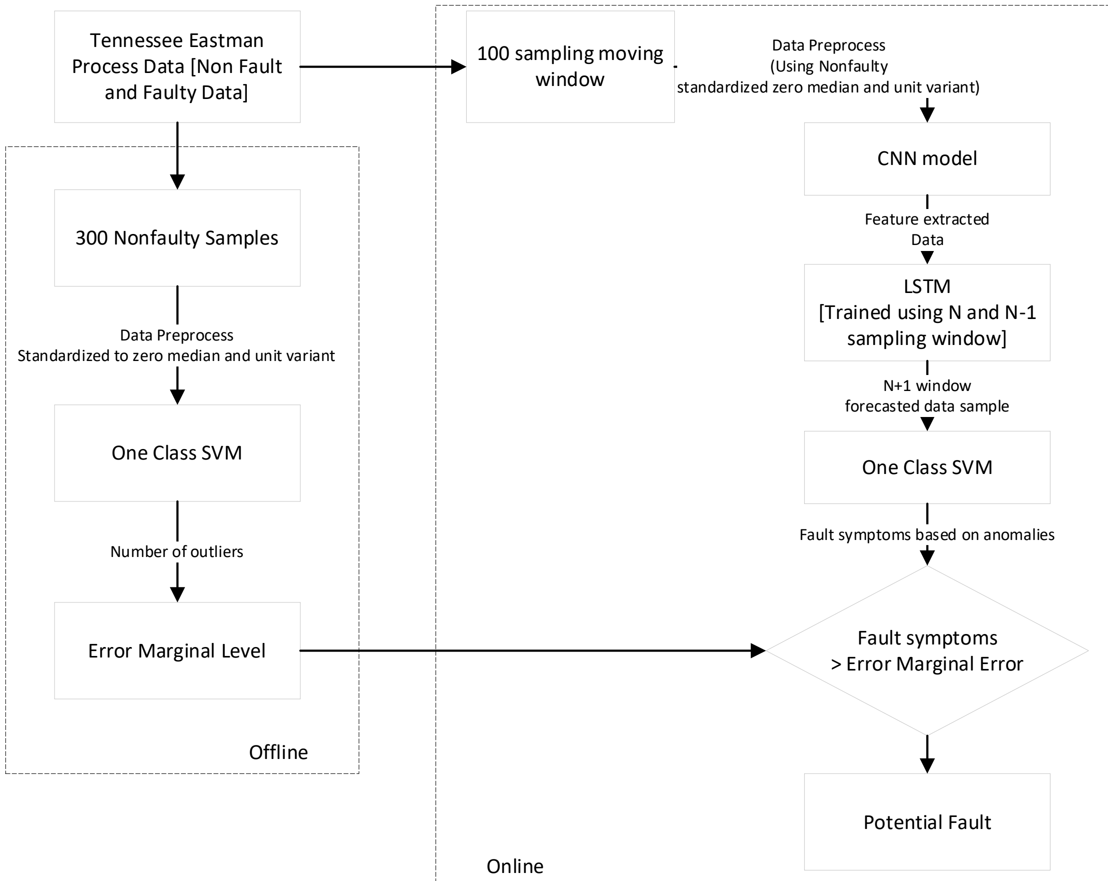


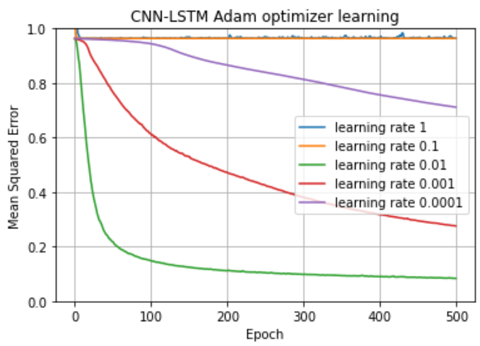
Figure 5.8: Model testing using TE process data.

5.4.2 Proposed Model Development and Testing

The step-by-step algorithm development and testing procedure are illustrated in Figure 5.8. The raw process data simulated from the system are initially scaled to zero mean and unit variance.

Step 1: Initially, a moving window of 100 samples is used to train the algorithm in phase I in offline conditions to forecast the data from past data. Then the online CNN-LSTM model is updated using real-time data samples; the n^{th} sampling window and $(n-1)^{th}$ sampling window are used to train the model in online conditions to forecast the n+1 sampling window data. Therefore, initially, after 200 samples, the CNN-LSTM model starts to forecast the synthetic features of the data.

When training the CNN-LSTM model, it is important to develop the model to minimize the cost function. Therefore, for the Adam optimizer, the learning rate and the number of iterations are tested over the range of learning rates and the number of epochs. Tests have been done with and without the CNN approach with LSTM.



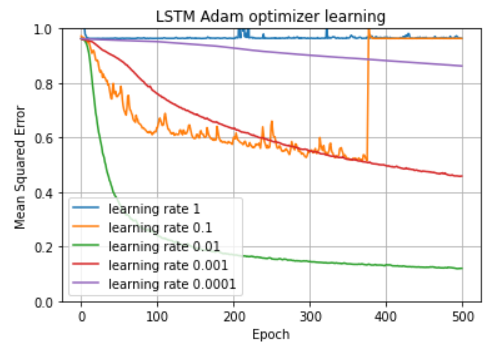


Figure 5.9: Model optimization using Adam optimizer.

Figure 5.9 shows the data forecast model learning rate and the number of iterations to optimize the model parameter by minimizing the mean squared error. To get a better performance, the proposed model is optimized using the Adam optimizer with a 0.01 learning rate and 400 iterations.

The comparison between CNN-LSTM and LSTM to forecast the data is summarized in Figure 5.9 and Table 5.2.

Table 5-2: CNN-LSTM and LSTM model comparison.

	CNN-LSTM	**LSTM**
Mean squared error (MSE)	0.091	0.187
Number of iterations	400	400
Learning rate (‘Adam’) optimizer	0.01	0.01

Step 2: As illustrated in Figure 5.8 and the algorithm in phase II, pre-processed 300 data samples using a window of 100 samples are fed into the OC SVM to train and determine the number of anomalies in each sampling window. Also, to account for noise in the data in the proposed model, the fault margin is set based on the experimental result:

$$Margin\ level = 2 \text{ x } \max(anomaly_{count}) \qquad (5\text{-}16)$$

Step 3: As described in the Online/real-time fault forecast algorithm, the pre-processed window of data is fed into the trained CNN-LSTM model to forecast the next window of 100 samples. Figure 5.10 shows the TE process's real and predicted data for the IDV1 fault condition.

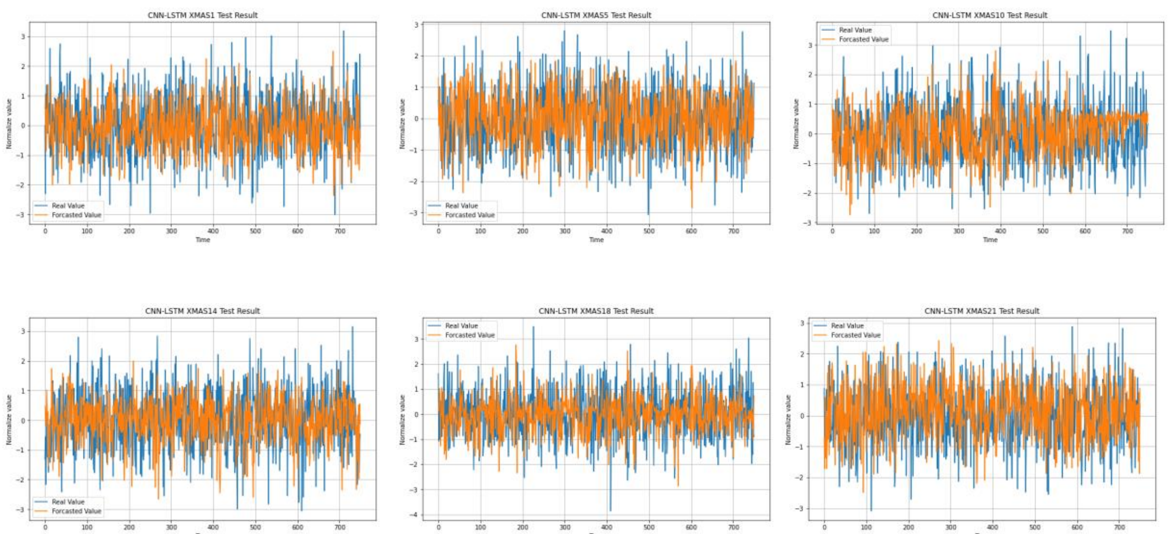


Figure 5.10: TE process IDV1 fault condition forecasted data.

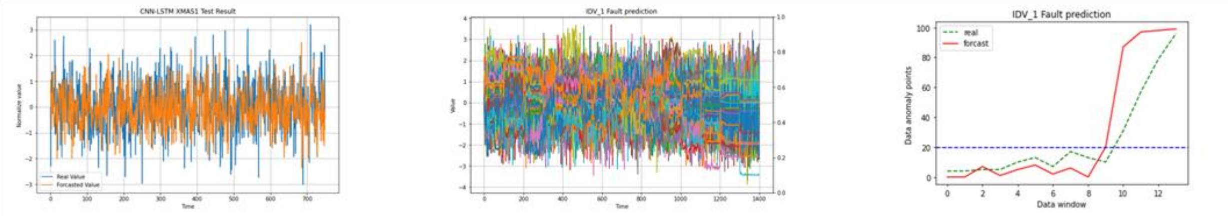
Step 4: Trained OC-SVM is used to find the outliers from CNN-LSTM forecasted data. Based on the detected outliers, system fault symptoms are evaluated by comparing the fault margin levels. When a fault symptom is excessive, and the condition continues for the next sampling window, the algorithm defines it as a potential fault. The obtained result is shown in Figure 5.12 and discussed in Table 5.3.

5.5 Results and Discussion

Of 20 different fault types, eight selected faults have been used to test the developed algorithm. IDV 1, 4, 5, and 6 faults are selected from step-type fault conditions. IDV 11 and 12 are selected to test random variation type fault conditions, and IDV 14 and 15 are selected to test the sticky type of fault condition in the Tennessee Eastman process.

5.5.1 Train and Test model

The proposed method is tested for each fault condition separately. The IDV 1 test results obtained from CNN LSTM and OC-SVM are shown in Figure 5.11. The IDV 1 single variable prediction and multivariate forecasted synthetic data are shown in Figures 5.11a and 5.11b, respectively. Anomaly points in each window sample are shown in Figure 5.11c. The fault margin obtained from the offline OC-SVM model with a non-faulty data sample test is shown in Figure 5.11c to obtain the fault condition. For IDV 1 offline, the one-class SVM fault margin is defined as 20. This margin is obtained from the first 300 non-faulty data and used online to evaluate the fault condition in each sampling window. As shown in Figure 5.11c, in the 9th sampling window, a fault condition is detected early using the forecasted early symptoms. Similarly, to all other selected fault conditions tested using the proposed methodology and shown in Figure 5.12, faults IDV 6 and 14 detect the fault exactly at the actual fault condition, and all other faults are detected at least 1 sampling window ahead.



a. Single feature forecast by CNN LSTM

b. Multi feature forecast by CNN LSTM

c. OC SVM Fault prediction result.

Figure 5.11 IDV1 Fault forecast testing using proposed models.

As illustrated in Figure 5.11c, 901 – 1000 samples predict the abnormal condition, where the fault is observed in the 1000th sample.

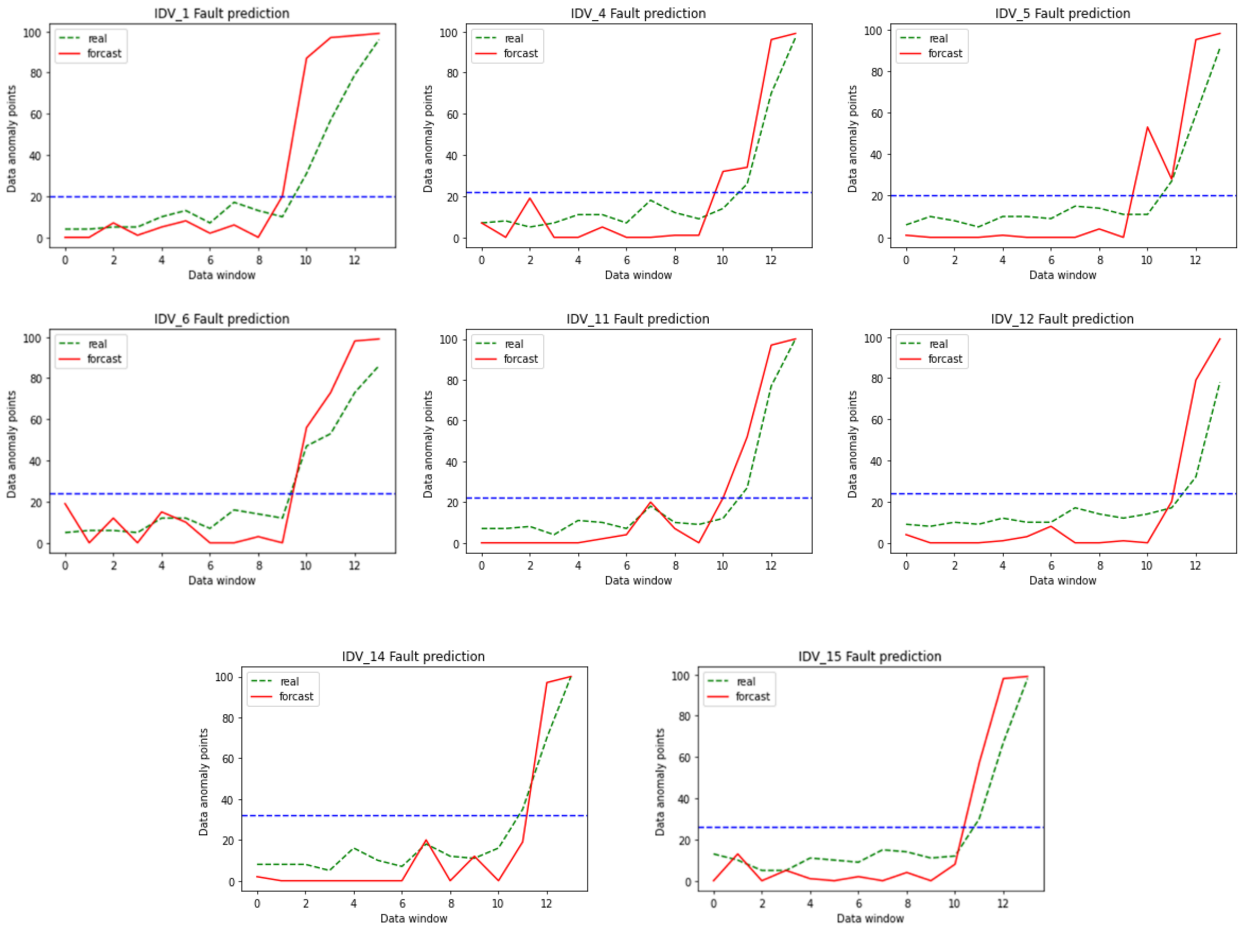


Figure 5.12: Tennessee Eastman Early Fault Detection Test Result.

5.5.2 Comparison of the Proposed Model With Recent Studies

The proposed fault detection method is compared with the recently proposed fault detection and diagnosis methods for the benchmark Tennessee Eastman model [31], [32]. The results are summarized in Table 5.3.

Table 5-3 : Fault detection model comparison.

TE Fault	Mahadevan and Shah (2009) [31]					Onel et al. (2019) [32]	Proposed model
	PCA-T^2	**PCA-Q**	**DPCA-T^2**	**DPCA-Q**	**OC SVM**	**Two class SVM**	**CNN-LSTM OC SVM**
IDV 1	99.2	99.8	99.4	99.5	99.8	99.9	100
IDV 4	4.4	96.2	6.1	100	99.6	100	100
IDV 5	22.5	25.4	24.2	25.2	100	100	100
IDV 6	98.9	100	98.7	100	100	100	100
IDV 11	20.6	64.4	19.9	80.7	69.8	100	100
IDV 12	97.1	97.5	99.0	97.6	99.9	100	100
IDV 14	84.2	100	93.9	100	100	100	100
IDV 15	-						100

The proposed fault symptoms forecasting model results are compared with those from unsupervised OC SVM and the recently proposed OC-NN; the results are summarized in Table 5.4. OC SVM and OC NN obtain the fault symptoms during the window frame after the actual fault has occurred. However, the objective of the proposed method is to forecast the fault symptoms using forecasted synthetic data, and therefore expected to detect the fault condition during the fault condition and one sampling window earlier, compared to the other two windowing methods algorithms. As shown in Figures 5.11 and 5.12, the IDV1 fault condition occurred at the 1000th data sample; therefore, OC SVM and OC NN detect the fault condition in the 1001-1100 sampling window frame. However, with the proposed model based on the fault symptoms captured by the CNN-LSTM model, combined with OC SVM, the fault symptoms are detected between 901 to 1000 samples. Therefore, as illustrated in Table 5.4, the proposed model predicts the fault condition earlier by one sampling window. The TE process data show that experimental results IDV 6 and IDV 14 did not promptly detect the fault conditions. However, from the model test, out of 8 fault conditions, 6 faults promptly capture the fault symptoms by one sampling window frame ahead.

Table 5-4: TE process fault forecast result comparison

Fault type	**One-class Neural Network [Arunthavanathan et al. (2020)] [33]** **Between 100 samples sampling window**	**One-class SVM** **[shown in Figure 5.9]** **Between 100 samples sampling window**	**CNN-LSTM One-class SVM [shown in Figure 5.9] Between 100 samples sampling window**	**An actual fault occurred /Observed (sample)**
IDV 1	1001 – 1100	1001 – 1100	901 – 1000	1001
IDV 4	1101 – 1200	1101 – 1200	1001 – 1100	1101
IDV 5	1001 – 1100	1001 – 1100	901 – 1000	1001
IDV 6	1001 - 1100	1001 – 1100	1001 – 1100	1001
IDV 11	1101 – 1200	1101 – 1200	1001 – 1100	1101
IDV 12	1201 - 1300	1201 – 1300	1101 – 1200	1201
IDV 14	1101 – 1200	1101 – 1200	1101 – 1200	1101
IDV 15	1201 - 1300	1201 – 1300	1101 - 1200	1201

To our knowledge, there are not any DNN based data forecasting or early fault detection methodologies applied to TE process data for comparison of the monitoring system, although, in recent years, DNN based approaches are being applied for known fault classification. Therefore, the proposed result is compared with the HMM-based prognosis model proposed by Galagedarage [29]. Comparison result illustrated in Table 5.5.

Table 5-5: Result comparison with recent model.

Fault type	HMM model [Galagedarage (2019)] [29]	CNN-LSTM one-class SVM (proposed model)	Real fault observed
IDV 1	1005	901 – 1000	1001
IDV 4	1005	1001 – 1100	1101
IDV 5	1005	901 – 1000	1001
IDV 6	**1020**	**1001 – 1100**	**1001**
IDV 11	1005	1001 – 1100	1101
IDV 12	1220	1101 – 1200	1201
IDV 14	**1105**	**1101 – 1200**	**1101**
IDV 15	1205	1101 - 1200	1201

To test the model, a core i7, 3.0 GHz processor equipped computer with 16GB RAM with GPU enabled is used to run the algorithm in python TensorFlow. On average, to run the 100-sample window, the algorithm process time is calculated at approximately 120 sec. However, with the floating-point neural processor, the processing time may be reduced. Therefore, when using large windows, it is recommended to use neural processors such as TensorFlow processors.

5.6 Conclusions

This paper proposed a robust approach to process fault prognosis. The proposed model can forecast multivariant time-dependent complex process data features and detect each sampling window's abnormality condition. This framework is proposed by interconnecting CNN, LSTM, and OC-SVM. A semi-supervised learning approach is used in the CNN LSTM model to forecast the next sampling window by learning the previous pattern. Historical data are used to train the CNN LSTM model. Finally, OC-SVM uses an unsupervised learning approach to define an abnormal condition of the forecasted synthetic data.

The benchmark Tennessee Eastman process is used to test the efficacy of the developed framework. The result confirms the capability of the algorithm to forecast the fault state as early as possible if the symptoms are observable from the previous data sampling window. The results also confirm that with limited knowledge of faults from the forecasted synthetic data, the fault state can be forecasted. This model can be further developed to classify the fault condition and diagnose the fault by defining the root cause for each fault condition, and also the model can be applied to predict the remaining useful life of the process systems and use predictive maintenance to prevent the system from being in a failure condition.

References

[1] M. A. Djeziri, S. Benmoussa, and M. E. Benbouzid, "Data-driven approach augmented in simulation for robust fault prognosis," *Engineering Applications of Artificial Intelligence*, vol. 86, pp. 154–164, Nov. 2019, doi: 10.1016/j.engappai.2019.09.002.

[2] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques-part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, 2015, doi: 10.1109/TIE.2015.2417501.

[3] T. Biagetti and E. Sciubba, "Automatic diagnostics and prognostics of energy conversion processes via knowledge-based systems," in *Energy*, Oct. 2004, vol. 29, no. 12-15 SPEC. ISS., pp. 2553–2572, doi: 10.1016/j.energy.2004.03.031.

[4] X. Li, Q. Ding, and J.-Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering & System Safety*, vol. 172, pp. 1–11, Apr. 2018, doi: 10.1016/J.RESS.2017.11.021.

[5] K. Javed, R. Gouriveau, and N. Zerhouni, "State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels," *Mechanical Systems and Signal Processing*, vol. 94. Academic Press, pp. 214–236, Sep. 15, 2017, doi: 10.1016/j.ymssp.2017.01.050.

[6] R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, "An analysis of process fault diagnosis methods from safety perspectives," *Computers and Chemical Engineering*, vol. 145. Elsevier Ltd, p. 107197, Feb. 01, 2021, doi: 10.1016/j.compchemeng.2020.107197.

[7] Y. Cheng, H. Zhu, J. Wu, and X. Shao, "Machine Health Monitoring Using Adaptive Kernel Spectral Clustering and Deep Long Short-Term Memory Recurrent Neural Networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 987–997, 2019, doi: 10.1109/TII.2018.2866549.

[8] J. Liu, C. Pan, F. Lei, D. Hu, and H. Zuo, "Fault prediction of bearings based on LSTM and statistical process analysis," *Reliability Engineering and System Safety*, vol. 214, no. April 2020, p. 107646, 2021, doi: 10.1016/j.ress.2021.107646.

[9] D. T. Hoang and H. J. Kang, "A survey on Deep Learning based bearing fault diagnosis," *Neurocomputing*, 2019, doi: 10.1016/j.neucom.2018.06.078.

[10] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2306–2318, 2017, doi: 10.1109/TNNLS.2016.2582798.

[11] Y. Wang, Z. Pan, X. Yuan, C. Yang, and W. Gui, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," *ISA Transactions*, vol. 96, pp. 457–467, Jan. 2020, doi: 10.1016/J.ISATRA.2019.07.001.

[12] P. Lin and J. Tao, "A novel bearing health indicator construction method based on ensemble stacked autoencoder," *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019*, pp. 1–9, 2019, doi: 10.1109/ICPHM.2019.8819405.

[13] D. Verstraete, E. Droguett, and M. Modarres, "A deep adversarial approach based on multi-sensor fusion for semi-supervised remaining useful life prognostics," *Sensors (Switzerland)*, vol. 20, no. 1, 2020, doi: 10.3390/s20010176.

[14] J. C. Hoskins and D. M. Himmelblau, "Artificial neural network models of knowledge representation in chemical engineering," *Computers & Chemical Engineering*, vol. 12, no. 9–10, pp. 881–890, Sep. 1988, doi: 10.1016/0098-1354(88)87015-7.

[15] T. Sorsa and H. N. Koivo, “Application of artificial neural networks in process fault diagnosis,” *Automatica*, vol. 29, no. 4, pp. 843–849, Jul. 1993, doi: 10.1016/0005-1098(93)90090-G.

[16] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[17] H. Zhao, S. Sun, and B. Jin, “Sequential Fault Diagnosis Based on LSTM Neural Network,” *IEEE Access*, vol. 6, pp. 12929–12939, 2018, doi: 10.1109/ACCESS.2018.2794765.

[18] G. Yue, G. Ping, and L. Lanxin, “An End-to-End model based on CNN-LSTM for Industrial Fault Diagnosis and Prognosis,” *IEEE Proceedings of IC-NIDC 2018*, pp. 3–7, 2018.

[19] L. Zheng *et al.*, “A fault prediction of equipment based on CNN-LSTM network,” *Proceedings - IEEE International Conference on Energy Internet, ICEI 2019*, pp. 537–541, 2019, doi: 10.1109/ICEI.2019.00101.

[20] P. Park, P. Di Marco, H. Shin, and J. Bang, “Fault detection and diagnosis using combined autoencoder and long short-term memory network,” *Sensors (Switzerland)*, vol. 19, no. 21, pp. 1–17, 2019, doi: 10.3390/s19214612.

[21] T. Xia, Y. Song, Y. Zheng, E. Pan, and L. Xi, “An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation,” *Computers in Industry*, vol. 115, p. 103182, Feb. 2020, doi: 10.1016/j.compind.2019.103182.

[22] Y. Han, N. Ding, Z. Geng, Z. Wang, and C. Chu, “An optimized long short-term memory network based fault diagnosis model for chemical processes,” *Journal of Process Control*, vol. 92, pp. 161–168, 2020, doi: 10.1016/j.jprocont.2020.06.005.

[23] T. Li, M. Hua, and X. U. Wu, “A Hybrid CNN-LSTM Model for Forecasting Particulate Matter (PM2.5),” *IEEE Access*, vol. 8, pp. 26933–26940, 2020, doi: 10.1109/ACCESS.2020.2971348.

[24] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, and G. Penn, “Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition,” *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp.

4277–4280, 2012, doi: 10.1109/ICASSP.2012.6288864.

[25] J. Zhao, X. Mao, and L. Chen, “Speech emotion recognition using deep 1D & 2D CNN LSTM networks,” *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, Jan. 2019, doi: 10.1016/j.bspc.2018.08.035.

[26] G. Yue, G. Ping, and L. Lanxin, “An End-to-End model based on CNN-LSTM for Industrial Fault Diagnosis and Prognosis,” *Proceedings of 2018 6th IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2018*, pp. 274–278, 2018, doi: 10.1109/ICNIDC.2018.8525759.

[27] H. Pan, X. He, S. Tang, and F. Meng, “An improved bearing fault diagnosis method using one-dimensional CNN and LSTM,” *Journal of Mechanical Engineering*, vol. 64, no. 7–8, pp. 443–452, 2018, doi: 10.5545/sv-jme.2018.5249.

[28] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Piatt, “Support vector method for novelty detection,” *Advances in Neural Information Processing Systems*, pp. 582–588, 2000.

[29] M. Galagedarage Don and F. Khan, “Process fault prognosis using hidden markov model-bayesian networks hybrid model,” *Industrial and Engineering Chemistry Research*, vol. 58, no. 27, pp. 12041–12053, 2019, doi: 10.1021/acs.iecr.9b00524.

[30] J. J. Downs and E. F. Vogel, “A Plant-wide Industrial Problem Process,” *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993, doi: 10.1016/0098-1354(93)80018-I.

[31] S. Mahadevan and S. L. Shah, “Fault detection and diagnosis in process data using one-class support vector machines,” *Journal of Process Control*, vol. 19, no. 10, pp. 1627–1639, Dec. 2009, doi: 10.1016/J.JPROCONT.2009.07.011.

[32] M. Onel, C. A. Kieslich, and E. N. Pistikopoulos, “A nonlinear support vector machine-based feature selection approach for fault detection and diagnosis: Application to the Tennessee Eastman process,” *AIChE Journal*, vol. 65, no. 3, pp. 992–1005, Mar. 2019, doi: 10.1002/AIC.16497.

[33] R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, “Fault detection and

diagnosis in process system using artificial intelligence-based cognitive technique," *Computers and Chemical Engineering*, vol. 134, p. 106697, Mar. 2020, doi: 10.1016/j.compchemeng.2019.106697.

Chapter 6 Remaining Useful Life Estimation Using Fault to Failure Transformation

Preface

A version of this manuscript has been submitted to IEEE Systems Journal. In this chapter, the RUL estimation algorithm has been developed using the degradation approach. In the developed methodology, the fault to failure transmission time is calculated online by detecting the fault condition and analyzing the root cause for the detected fault condition. This chapter contributes the targeted 5th and final objective of the thesis defined in chapter 1.

Abstract

The remaining useful life (RUL) plays a significant role in the predictive maintenance of process systems. In recent years, data-driven approaches have been used to estimate the RUL. However, without a deeper understanding of process systems and their failure mechanisms, it is challenging to estimate the RUL for process systems. This demands an approach that integrates data with mechanistic understanding. This work proposes a model to estimate the remaining useful life of process systems in real-time by monitoring the system's fault condition. The fault is modeled using real-time observed data, while the progression of fault to failure is modeled using mechanistic understanding. These steps are carried out using a novel mixed model combining machine learning-based autonomous fault diagnosis with root cause analysis and a statistical degradation model. The autonomous RUL estimation model is then developed as a self-learning model. A fault dictionary is developed to store the fault pattern and failure margin for different failure conditions. The proposed approach is benchmarked using the Tennessee Eastman process.

6.1 Introduction

In process industries, maintenance is an utmost necessary, however, a challenging part of the overall process operation to maximize functionality and minimize breakdowns. Reactive and proactive maintenance approaches are commonly used in process systems. Reactive maintenance involves repairing or substituting a machine component only once it fails and can no longer operate [1]. A proactive maintenance strategy aims to identify and fix the cause of the system failure before it occurs [2]. The process industry is more likely to use a proactive approach to prevent the process plant before they lead to failure. Proactive maintenance is classified into preventive maintenance, condition-based maintenance (CBM), and predictive maintenance (PM) [3], [4]. Preventive maintenance is based on the scheduled maintenance at a regular interval. Like reactive maintenance that is based on the repair of an asset when it breaks down, preventive maintenance that is based on periodic inspection and replacement, is also disappearing from industrial practice [5].

Condition-based monitoring and predictive maintenance approaches have been widely targeted in the research and industry in recent years. Condition-based monitoring and maintenance and predictive maintenance are highly dependent on the system's condition information [6]. Therefore, these two approaches depend on the sensor data and fault detection approaches. However, the main difference between these two methods is that the latter combines sensor measurements with accurate algorithms to predict the exact moment of the maintenance schedule. CBM takes maintenance action when a fault is detected, and PM extends the algorithm to predict the fault to failure transmission time to schedule the maintenance [7]. Moreover, fault detection and diagnosis algorithms provide the information require for CBM, however, PM should forecast the fault to failure transmission time to schedule the maintenance.

By way of introducing fault to failure transformation in the article, first, address the definition of these terms used in process monitoring. The term fault is an unpermitted deviation of a system parameter from the acceptable condition, which defines the fault as a process abnormality[8][9]. Failure is defined as a permanent interruption of a system's ability to perform a required function under a specified operating condition[9]. This defines the underlying cause of the abnormality that permanently terminates the system's required performance. Therefore, system fault is considered a state of deviation in the process systems, and failure is considered an event[10]. Therefore, one of the important aspects of predictive maintenance is the estimation of fault to failure transformation time to determine the abnormal state deviation to system failure event by defining it as a remaining useful life (RUL) estimation [11].

In the process industry, RUL estimation techniques mainly focus on predicting the failure of machines or components in process plants. It is imperative to predict the process systems failure condition as early as possible and define the RUL, not only for reducing maintenance cost and increasing machines' life cycle but also to reduce health, safety and environmental risks [12]–[14]. However, accurately predicting the RUL of an equipment cannot only guarantee failure prevention but should also allow scheduled maintenance of the equipment within a safer timeframe [6].

The existing, RUL prediction methods are classified as physics based models and data-driven models [15], [16]. Physics based models are related to determining the mathematical relationship, such as the state-space models and analytical models, and include an observer-based method and parity space to estimate the current fault condition and predict the future [17]. However, these models are not sufficient to monitor the system condition until a failure occurs or the system condition becomes observable. Data-driven models, such as traditional statistical analysis and machine learning models, are initiated to define the degradation process as a functional relationship between system conditions and monitoring data [15].

Depending on the data availability, data-driven models are further classified as RUL estimation using survival models, similarity models, and degradation models [18]. To estimate the RUL using survival model system lifetime data, proportional hazard and probabilistic distribution of the component failure time methods are utilized [19]. RUL estimation using similarity models require failure data from similar components or distinct components with similar behavior [19], [20]. Also, these models record the degradation profile and compare them with process data. RUL estimations using degradation models predict the system health condition and analyze when the predictor crosses a predefined threshold [20], [21].

In process industries, process systems are constructed using several components, and each plant uses them for different purposes and unique operations. Therefore, collecting similar behavior failure data and lifetime data with failure conditions will be a challenge or impossible. Hence, degradation models are a better choice to estimate the RUL in process systems since most process plants define the failure threshold for each component and parameter. Since process plant systems are complex and combine several components, process system data are multivariate and highly correlated [22]. When applying degradation methods to estimate the RUL, the main challenge is to identify the condition indicator for the system condition deviation [21].

Also, process systems undergo modifications and changes of operating conditions with time, and the definition of normal and abnormal states can change in the course of time [23]. Therefore, it is important to monitor the normal to fault transition behavior, and fault to failure estimation during the systems run time. Also, in line with the Industry 4.0 concepts, online autonomous failure prediction and estimating the RUL pose new challenges [24], [25].

Problem Statement

With the digitization of process systems and the introduction of Industry 4.0, it is necessary to develop predictive maintenance strategies integrated with automated process safety tools [20], [22]. In the process industries, applying these techniques leads to higher equipment efficiencies, less downtime of the system, and enhanced process safety [23]. To adopt predictive maintenance in process plant systems, RUL estimation plays an important role. In the recent literature, to estimate the RUL, data-driven approaches are used with the data set containing many runs to failure sequences [10], [24]–[26]. In the initial stage of the process, plant run to failure sequence data are not available and can be captured after the fault or failure event occurs. Also, in the process plant industry, it is impossible or rare to evaluate similar models because of plant diversity [27]. This leads to the problem of estimating RUL in a process plant.

The proposed frameworks address this issue by presenting a novel approach: detecting the fault condition and identifying the root cause using an online assessment, and further estimating RUL based on the defined root cause failure threshold. In this work, Hi and Low Alarm limits are used as a failure threshold.

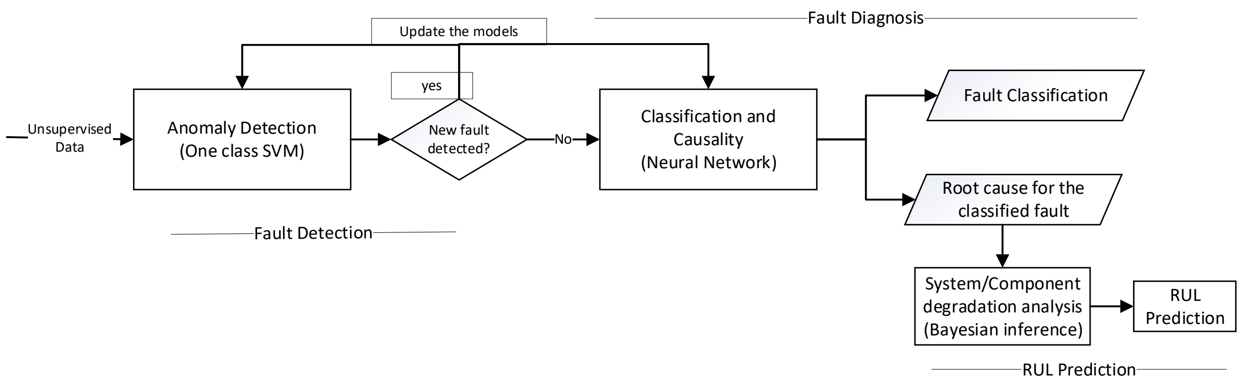


Figure 13 : Integrated model for FDD and RUL estimation.

As illustrated in figure 1, machine learning models have been proposed to monitor the real-time system operation, and to estimate the failure threshold from the deviation point, a statistical approach is proposed. In order to detect the normal to abnormal behavior using unsupervised data, OC- SVM is used. To classify the fault condition and identify the root cause of the abnormal behavior, NN permutation algorithm has been proposed. Finally, to estimate the RUL from the deviation point, statistical degradation analysis is proposed.

The remaining part of the chapter is structured as follows. Section 6.2 reviews the background information of the proposed model. Section 6.3 describes the overall proposed methodology. Section 6.4 provides the model experiment and test results. Finally, conclusions are presented in Section 6.5.

6.2 Overview of the Proposed Model

In this section, the problem statement of autonomous RUL prediction in real-time is formulated, and the theoretical background of One-class SVM, incremental permutation NN, and degradation models are introduced. A novel methodology is constructed by integrating self-learning models with the degradation model to overcome the challenges in real-time RUL estimation. Finally, algorithm integration and RUL prediction procedures are presented.

6.2.1 Problem Statement

With the digitization of process systems and the introduction of Industry 4.0, it is necessary to develop predictive maintenance strategies integrated with automated process safety tools [23], [25]. In the process industries, applying these techniques leads to higher equipment efficiencies, less downtime of the system, and enhanced process safety [26]. To adopt predictive maintenance in process plant systems, RUL estimation plays an important role. In the recent literature, to estimate the RUL, data-driven approaches are used with the data set containing many runs to failure sequences. [13], [27]–[29]. In the initial stage of the process, plant run to failure sequence data are not available and can be captured after the fault or failure event occurs. Also, in the process plant industry, it is impossible or rare to evaluate similar models because of plant diversity [30]. This leads to the problem of estimating RUL in a process plant.

The proposed frameworks address this issue by presenting a novel approach: detecting the fault condition and identifying the root cause using an online assessment, and further estimating RUL based on the defined root cause failure threshold. In this work, Hi and Low Alarm limits are used as a failure threshold.

6.2.2 Incremental One Class SVM

In this approach, unsupervised raw sensor real-time data are used for online monitoring of the system. Incremental one-class SVM is proposed as an anomaly detection algorithm in the methodology to detect the deviation from a normal to an abnormal condition.

A one-class support vector machine (OCSVM) formulation is a specific instance of SVM [31]. In the traditional binary SVM classification, a hyperplane separates the two classes with a large possible margin, and it is supported by the support vectors [32], [33]. In the one-class classification, the hyperplane exhibits only positively labeled data during the training. In OCSVM, the origin of the coordinate system is assigned to the hyperplane corresponding to the negative class [31]. As a result, the goal of OCSVM is to discover the hyperplane that is further away from the origin and where positively labeled data occur in the positive half-space of the hyperplane. Therefore, the OCSVM model output provides the +1 value for data within the region and the -1 value for the region outlier, as shown in Equation 6.1.

$$d_n = \begin{cases} +1\ if\ data\ point\ x_i\ is\ in\ the\ region \\ -1\ if\ data\ point\ x_i\ is\ not\ in\ the\ region \end{cases} \quad (6\text{-}1)$$

where $'x_i'$ is a data point available in the data set, $x_i \subseteq$ (training or testing data). Let x be a training input sample; then, to separate the data points from the origin, OCSVM initially solves the quadric programming equation given in Equation 6.2 and optimizes the parameters.

$$\min_{w,\varepsilon,b} \frac{1}{2}|w|^2 + \frac{1}{nr}\sum_{i=1}^{n} \varepsilon_i - b \quad (6\text{-}2)$$

Subject to $\langle w, \Phi(x_i)\rangle \geq$ b -ε_i, $\varepsilon_i \geq 0$

where $'\varepsilon_i'$ is a slack variable corresponding to the ith training sample that allows it to lie on the other side of the decision boundary. $'\Phi'$ is a nonlinear mapping function that maps $'x_i'$ to kernel space. $'b'$ is a bias term, $'r'$ is the regularization parameter, and 'n' is the size of the dataset. When the optimization is complete, the condition may be used to infer query sample testing data.

$$d_n = sgn(\langle w, \Phi(x_i)\rangle - b) \quad (6\text{-}3)$$

With the help of Lagrange multipliers $\alpha_i\beta_i > 0$, Equation 6.2 can be modified as:

$$\mathcal{L}(w,\varepsilon,\alpha,\beta) = \frac{1}{2}|w|^2 + \frac{1}{nr}\sum_{i=1}^{n} \varepsilon_i - b - \sum_{i=1}^{n} \alpha_i(\langle w, \Phi(x_i)\rangle - b + \varepsilon_i) - \sum_{i=1}^{n} \beta_i\varepsilon_i \quad (6\text{-}4)$$

where the column vectors $\alpha = [\alpha_i, \dots, \alpha_n]^T$ and $\beta = [\beta_i, \dots, \beta_n]^T$. When the derivatives of primal variables are set to zero, it can be shown that

$$w = \sum_{i=1}^{n}\langle \alpha_i, \Phi(x_i)\rangle,\ \alpha_i = \frac{1}{rn} - \beta_i \leq \frac{1}{rn} \text{ and } \sum_{i=1}^{n} \alpha_i = 1. \quad (6\text{-}5)$$

By substituting these values in Equation 6.2, the dual optimization problem is derived as follows:

$$\min_{\alpha} \frac{1}{2}\sum_i \sum_j \alpha_i\alpha_j K(x_i, x_j) \quad (6\text{-}6)$$

Subject to $0 \leq \alpha_i \leq \frac{1}{nr}, \sum_{i=1}^{n} \alpha_i = 1$.

Furthermore, it can be shown that when $0 \leq \alpha_i \leq \frac{1}{rn}$ is satisfied, the bias term can also be expressed as:

$$b = \langle w, \Phi(x_i) \rangle = \sum_j \alpha_j K(x_i, x_j) \tag{6-7}$$

The best value of parameters in the problem described in Equation 6.2 can be obtained using the kernel function 'K', without explicitly specifying the mapping function $'$

Φ' using the dual form of the problem, as shown in Equation 6.5. Therefore, the decision function for any test data (x_{test}) can also be expressed in terms of the kernel function using the dual variables and vectorized training samples as follows.

$$d_n = sgn(\sum_i^n \alpha_i K(x_i, x_{test}) - b) \tag{6-8}$$

To develop the incremental OC- SVM as proposed in this methodology, training data sample x_i is updated continually over time based on the newly detected fault condition [31]–[34]. Therefore, the hyperplane and the optimizing parameters, as shown in Equations 6.2 and 6.5, are up to date with the change of process data deviation. Therefore, incremental one-class SVM is trained by using x_i and some of the x_{test} concatenated data.

$$x_{incremental} \subset \forall(x_i).\, x_{test} \tag{6-9}$$

6.2.3 Incremental Output Permutational Neural Network

To develop an autonomous fault diagnosis and root cause analysis method, a dynamic output permutation neural network is proposed in the methodology. In the proposed methodology, the number of nodes in the output layer change over time based on the newly detected fault. Also, in the proposed methodology, one feature of the neural network data is shuffled and permutated to identify the root cause of the classified fault condition.

6.2.3.1 Incremental Output Layer

Figure 6.1 shows the proposed NN with incremental output nodes. Activation nodes in the hidden layer and output layer are formulated as follows:

$$a_i^l = g\left(x_i.\theta_i^l + b^l\right) \tag{6-10}$$

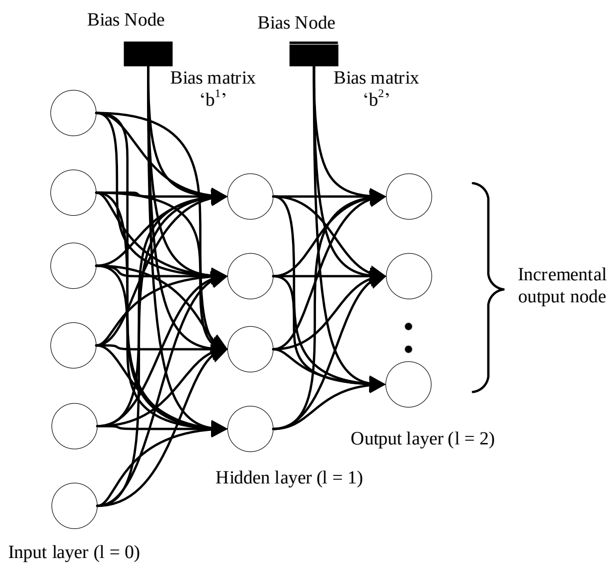


Figure 6.14 Incremental NN when training with the new fault data [32].

where, $'a_i^l{}'$ is an activation parameter of node $'i'$ in layer $'l'$. Proposed NN architecture has an input layer, output layer, and a single hidden layer, l = 1, 2 define the hidden and output layer respectively. $'\theta_i^l{}'$ is the weight matrix from layer; l to l+1. $'b^l{}'$ is a bias matrix in each layer. $'i'$ defines the number of nodes in each layer separately. $'g'$ is a nonlinear activation function to determine the appropriate input and output relationship.

In a traditional NN architecture, the number of output layer nodes is fixed based on the number of classes. The proposed methodology is similar to that proposed by Aruthvanathan et al. [35], and the number of nodes in the output layer developed as an incremental node depends on the number of faults detected over time. Therefore, in $'a_i^2'$, $'i'$ nodes change dynamically and NN parameters θ and b are optimized whenever the number of nodes increases. Similarly the proposed NN model will be continuously updated to classify all detected fault conditions.

$$\forall(outputnodes) \subseteq (\#ofdetectedfaults) \tag{6-11}$$

In the proposed model, 68 nodes were used in the hidden layer, and the ReLu function was used as a node activation function. Also in the output layer, the SoftMax activation function was used to activate each node; hence it provides one hot encoding classification from each output node. The number of output nodes are defined based on Equation 6.10. An Adam optimizer with an initial hyperparameter learning rate of 1e-3 and 300 epochs is used to optimize the parameters using the backpropagation algorithm. In the optimization category, cross-entropy is used as a loss function. When updating the neural network model, online hyperparameters are optimized by running it with a range of values.

6.2.3.2 Permutation Input Layer

To evaluate the root cause for the detected fault condition, the permutation algorithm is integrated with the proposed NN model. However, permutation importance or root cause for the diagnosis of a fault can be done after the NN model has been well fitted. Hence, this approach is based on comparing the NN model accuracy, and it is important to maintain the model accuracy over 95%.

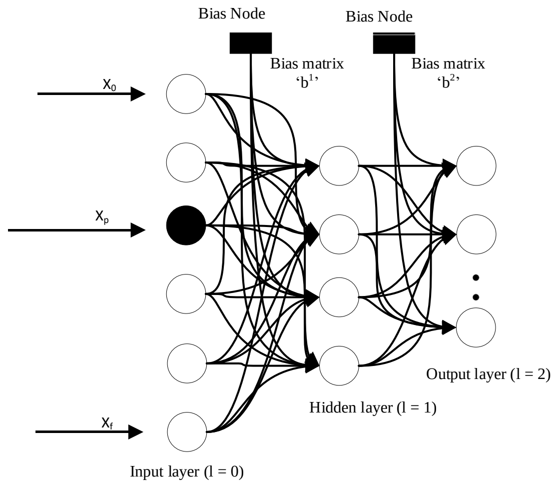


Figure 6.15 Permutation NN to define the root cause.[35]

As shown in Figure 6.2, this method is developed using a simple permutation principle, when a single feature data (x_p) is randomly shuffled in the input layer, leaving the model parameters, hidden, and output layers all in place. Feature importance is defined by the mean drop in the accuracy of the training model when each feature is shuffled [36]–[38]. Feature importance will be selected by the shuffled feature, which lowers the model accuracy. In the fault diagnosis, the identified feature importance determines the root cause for the classified fault condition [37].

6.2.4 Degradation Model

As mentioned in section 6.1, one of the main challenges for RUL prediction is collecting the system failure data. For this purpose, a degradation model to forecast the time to reach the threshold level is proposed in this section. One of the limitations in this approach is that the failure threshold must be known for the whole process and measured variables. In the process industry, the variable failure threshold and consequence threshold vary depending on the nature of the system, and the threshold information is available in the process plant manual or in the distributed control system (DCS), defined as an alarm threshold.

In the process industry, the alarm configurations are developed during the plant's engineering stage based on safety and technological factors. According to process system ANSI/ISA 18.2 standard alarm management, the consequence threshold is defined as the point at which consequences start to occur, and the alarm threshold is calculated by subtracting the consequences threshold from the process dead time, acknowledgment time, and operation time for the related process measurement [39]. Therefore, high and low alarm thresholds are usually defined for each variable in the process systems and considered in the RUL design process.

Regression methods are commonly used as degradation models to estimate the remaining time to reach the failure threshold [29]. Linear, power, and exponential regressions commonly use the simplest degradation models to estimate the condition monitoring path [40]. Due to the uncertainty in monitoring data, Bayesian regression is proposed as a degradation model to estimate the RUL.

In the Bayesian regression approach, a linear regression problem is formulated using probability distribution [41]–[44]. The estimated output response is not a deterministic single value but is assumed to be drawn from a probability distribution such as spherical Gaussian distribution [41]. Hence, the model for the Bayesian linear regression is formulated as:

$$y|X,\beta \sim N(\beta^T X, \sigma^2 I) \tag{6-12}$$

where y is a response output with a Gaussian distribution characterized by a mean and a variance. The mean for the regression is obtained by multiplying the transpose of the weight matrix (β^T), by the predictor matrix (X). $'\sigma^2'$ denotes the variance, and 'I' is the homoscedastic variance defined by the identity matrix.

Bayesian ridge regression is one of the most effective types of Bayesian regression [43], [44]. Ridge regression is a commonly used regularization method that looks for β, which minimizes the sum of the residual sum of squares and a penalty term. Ridge regression is formulated as:

$$\hat{\beta} = {}^{argmin}{}_{\beta}(y - X\beta)^T(y - X\beta) + \lambda|\beta|_2^2 \quad (6\text{-}13)$$

where $\|\beta\|^2 = \beta_1^2 + \cdots + \beta_p^2$ and $\lambda \geq 0$ are hyperparameters of the regression.

The Bayesian view of ridge regression is obtained by noting the minimizer of Equation 6.13, and this can be considered as the posterior mean of a model where $\beta \sim N(0, \tau^2 I)$. For some constant τ, this allows computing the posterior distribution of $\hat{\beta}$ formulated as:

$$\hat{\beta} = {}^{argmin}{}_{\beta}\left[(y - X\beta)^T(y - X\beta) + \frac{\sigma^2}{\tau^2}|\beta|_2^2\right] \quad (6\text{-}14)$$

where the Bayesian regression ridge estimator parameter $\lambda = \frac{\sigma^2}{\tau^2}$. In the gamma distributions, the conjugate prior for Gaussian precision is selected as prior over $\beta \ and \ \lambda$.

6.3 Prediction of RUL Using the Proposed Hybrid Method

This section elaborates on the proposed hybrid machine learning and mixed statistical model to predict the remaining useful life of process systems. The proposed methodology includes three stages: fault detection using OCSVM anomaly detection, fault diagnosis, root cause analysis using incremental output permutation NN, and RUL prediction using the failure threshold and degradation model. The basic principles and models' alteration of each model are discussed in sections 6-2.

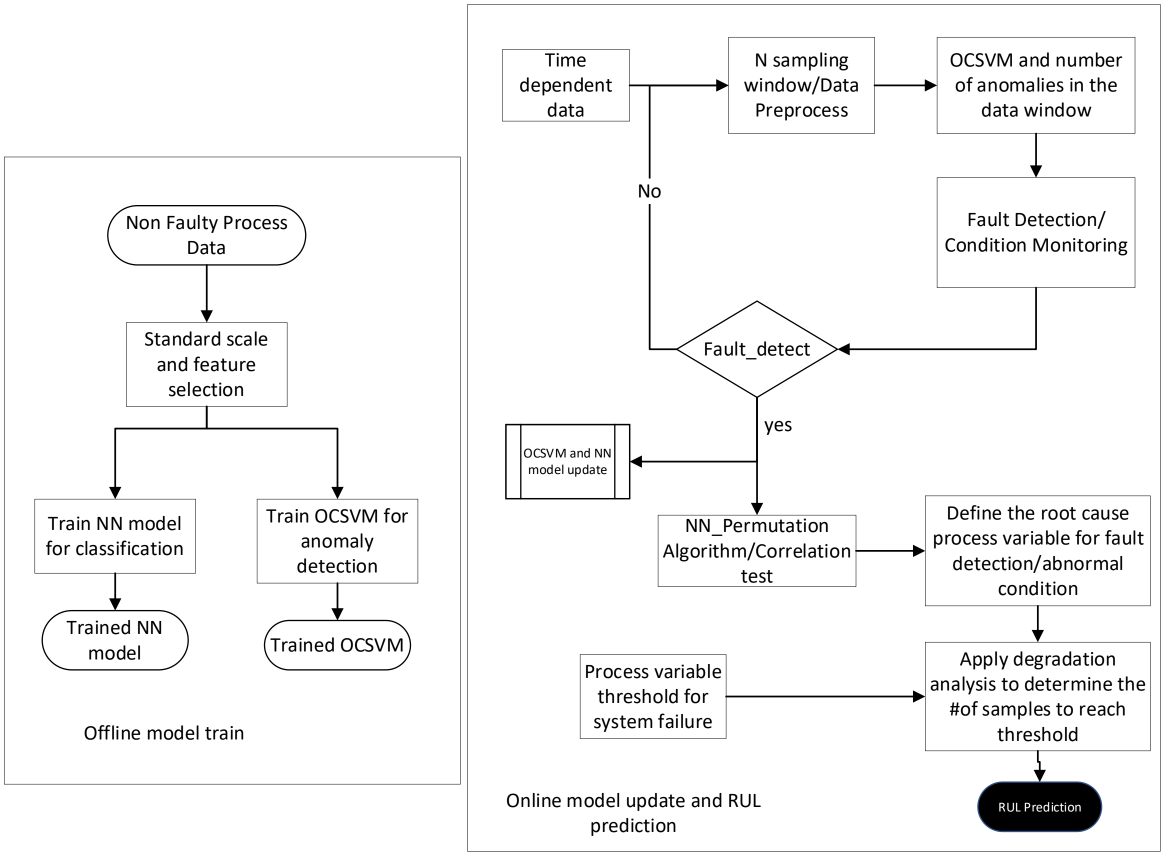


Figure 6.16 Overview of the proposed methodology flowchart.

6.3.1 Offline Training

The complete methodology and workflow of the model are shown in Figure 6.3. As shown in the proposed methodology workflow, OCSVM and NN models are initially trained by using nonfaulty data. Raw sensor data are mean-centered and scaled to standard deviation. The standardization score of the variable is formulated as:

$$z_i = \frac{X_i - \mu}{s} \tag{6-15}$$

where $'X_i'$ is an ith testing sample data, $'\mu'$ is a mean of the training sample, and s is the training samples' standard deviation.

The Spearman correlation test has been proposed in the feature selection by eliminating one of the highly correlated variables from the raw sensor data. In the proposed methodology, to evaluate the most appropriate root cause for the detected fault, highly correlated variables and control variables are eliminated from the highly correlated feature.

For nonfaulty data samples, considering two features, F and V, and corresponding ranks for the features are F_r and V_r then, the Spearman rank correlation coefficient r_s:

$$r_s = \rho(F_r V_r) = \frac{COV(F_r, V_r)}{S(F_r).S(Vr)} = \frac{n \sum_{f_r \in F_r, v_r \in V_r} (f_r v_r) - \sum_{f_r \in F_r} (f_r) . \sum_{v_r \in V_r} (v_r)}{\sqrt{\left(n \sum_{f_r \in F_r} f_r^2\right) - \left(\sum_{f_r \in F_r} f_r\right)^2} . \sqrt{\left(n \sum_{v_r \in V_r} v_r^2\right) - \left(\sum_{v_r \in V_r} v_r\right)^2}} \quad (6\text{-}16)$$

For multivariate features, the test is done within each variable combination separately. The $'r_s'$ values closer to -1 and +1 indicate highly negative and positive correlation, respectively. When $'r_s'$ decreases and gets closer to zero it indicates a non-correlation. In the proposed method, +/-0.7 has been used as a cut-off threshold, and variables outside the limit are from the training dataset.

Algorithm 1 shows the complete OCSVM and NN training. To train the one-class SVM, $w \text{ and } b$ parameters from Equation 6.2 are optimized using 'n' samples of nonfaulty data. Radial base functional kernel and gamma function are used as hyperparameters in the optimization. Similarly, to train the NN model, θ and b parameters are optimized using an "Adam" optimizer. NN hyperparameters' learning rate and the number of iterations are determined by the range of values tested over time. A 1e-3 Adam optimizer learning rate and 300 iterations provide the best result from the experiment.

In the offline training, OCSVM validation was done using windowed nonfaulty samples. The 'n' number of samples were gathered as data samples. Moving samples window were tested using OCSVM, and the number of anomalies within a sampling window were counted separately and defined as C_w. Using the non-fault sampling window anomaly count, the fault margin was formulated as follows:

$$f_m = \mathbb{Z}[\max(C_w) + \delta * \max(C_w)] \quad (6\text{-}17)$$

where the proposed methodology's fault margin $'f_m'$ is calculated using the maximum number of anomalies in the moving window. $'\delta'$ defines the noise margin. From the experimental result, $\delta = 0.2$ for 5% noise level in the data set and $\delta = 0.25$ for 10% noise level.

Algorithm 1 Offline Training

---

1. **Input:** Nonfaulty data samples ($\mathbb{X}^{N\times F_e}$), length of the data samples (N), features in the data sample (F_e). To train the OCSVM data samples $\left(x^{(i)}\right)|_{i=0}^{N}$ and to train the NN model $\left(x^{(i,F_e)}, y^{(i,c)}\right)|_{i=0}^{N}$ used. In supervised data, c denotes the #of fault conditions, and initially, c=1 for nonfaulty samples.
2. **Output:** OCSVM model, NN model and, Fault Margin.
3. Data Preprocessed using Equation 6.15. $\{z_i | 0 < z_i < 1\}$
4. Select feature using Spearman correlation test and eliminate one of the highly correlated features and number of selected features are defined as F_s $\forall(F_s) \subset F_e$ and preprocessed feature selected data samples are denoted as $\mathbb{D}^{NxF_s}$.
5. **for** $(i = 1{:}\,N)$
6. OCSVM trained by $\mathbb{D}^{N\times F_s}$, data set, and $w\ and\ b$ in Equation 6.2 are optimized.
7. **end**
8. OCSVM validated using $\mathbb{D}^{M\times F_s}$, where ‘M’ is the number of nonfaulty samples sampled using 'n' sample window. $\mathbb{D}^{n\times F_s} \subset \mathbb{D}^{M\times F_s}$ and $\mathbb{Z}\left(M/_n\right)$ define the number of windows in the validating samples. s_w denotes the window index and $w \in \mathbb{N}$.
9. **for** (1: w)
10. **for** $(i =\ 1{:}\,\text{n})$
11. Compute the decision score for each sample: d_i ← formulated using optimized $w\ and\ b$ from step 3 and Equation 6.3.
12. **if** $(d_i == 1)$ then
13. $C_W =\ C_W + 1$
14. **else**
15. $C_W =\ C_W$ Normal operation

16. **end**

17. **end**

18. $f_m \leftarrow$ From the calculated C_W and Equation 6.17 fault margin f_m is calculated.

NN Model Training

19. **for** (1: # of epochs)

20. **for** ($i = 1: N$)

21. NN trained by $\mathbb{D}^{N \times F_s}$ supervised data (x^i, y^i) and optimized w and θ by using Equation 6.9 and backpropagation algorithm.

22. **end**

23. **end**

24. **if (**MSE{Model} < 95%)

25. Repeat step by changing # of epochs and learning rate. [NN hyper parameters]

26. **else**

27 Complete the training process.

28. **end**

- Number of

6.3.2 Online Fault Detection and Diagnosis

In online monitoring, unsupervised data $\left(x^i\right)|_{i=1}^{N}$ windowed samples are preprocessed, and features are selected using algorithm 1, steps 1 and 2. The same features are selected in algorithm 1, and step 2 is selected as a feature in online testing. Therefore, the data dimension for the window remains the same as $\mathbb{D}^{n \times F_s}$.

$\mathbb{D}^{n \times F_s}$ samples data window is fed into the trained OCSVM model obtained in algorithm 1 to test the data deviation by analyzing the anomalies in the window. Algorithm 2 details the complete process of fault detection using the OCSVM.

Algorithm 2 (a) Fault detection using trained OCSVM

Input: $\mathbb{D}^{nxF_s}$, $\left(x^i\right)|_{i=1}^{n}$ windowed preprocessed sample. Process steps 1 and 2 in algorithm 1.
Output: Set decision score, #anomalies in the data window, and fault condition.
for $(i = 1{:}\,n)$
  Compute the decision score: d_i
  if $(d_i == -1)$ then
    $A_i = A_i + 1$; where, $'A_i'$ define the accumulated anomaly count.
  else
    $A_i = A_i$
end
if $(A_i > f_m)$ then
  New fault detected and new label created for the supervised data: $\left(x^{(i),f_e}, y^{(i),(c+1)}\right)|_{i=1}^{n}$, $(c+1)$ denote #of fault condition increased after a newly detected fault condition.
  OCSVM update and trained with current unsupervised window concatenate with past training window:
  $(x^{(i),f_e}) \subseteq \left(x^{(i),f_e}|_{i=1}^{N}, x^{(i),f_e}|_{i=1}^{n}\right)$
  NN model update and trained with concatenated supervised data:
  $(x^{(i),f_e,}y^{(i),(c+1)}) \subseteq \left(\left(x^{(i),f_e}|_{i=1}^{N}, x^{(i),f_e}|_{i=1}^{n}\right), (y^{(i),f_e}|_{i=1}^{N}, y^{(i),(c+1)}\Big|_{i-1}^{n}\,)\right)$
else
  Run algorithm 2(b) to classify the known fault condition by the trained NN model.
end

If $\mathbb{D}^{nxF_s}$ data sample window has tested over the OCSVM, and the new fault is not detected by the algorithm, data samples are fed into the trained NN model to classify the fault condition (f). Here, classified fault $\in C$. One hot encoding method is used in the NN output. For classified fault condition (f), one hot encoding is formulated as:

$$\left\{f_c \in \{0,1\}^c : \sum_{i=1}^{c} f_c = 1\right\} \tag{6-18}$$

Algorithm 2(b) NN fault classification

1. **Input:** Known fault data, $\mathbb{D}^{nxF_s}$, $\left(x^i\right)|_{i=1}^{n}$ preprocessed windowed sample
2. **Output:** One hot encoding fault label

3. **for** $(i = 1: n)$

4. Using trained NN, for each data sample classified one hot encoding predicted.

5. **end**

6 $\max(f_c = 1)$ for C in Equation 6.18 defined as the fault classification and store the fault information.

7. Run algorithm 2(c) for the detected fault root cause analysis.

8. **end**

Once the known fault is classified using the NN algorithm, the NN model uses the permutation algorithm to define the root cause of the classified fault condition. To process the algorithm, preprocessed data and features of the selected data window $\mathbb{D}^{nxF_s}$ are fed to the permutation NN algorithm. As presented in Figure 6.2 in section 6.2, the permutation algorithm is performed, and model accuracy is calculated using mean squared error. The highest mean squared error among the feature permutations is selected as a root cause for the detected fault condition. Hence, for the pth column feature the permutation important score (PS_p) is defined as:

$$PS_p = \frac{|\mathrm{PMAE} - \mathrm{MAE}|}{MAE\ x\ 100} \tag{6-19}$$

where 'PMAE' denotes mean absolute error with pth column data shuffle, and MAE denotes mean absolute error without permutation.

Algorithm 2(c) Root cause analysis using permutation NN

1. **Input:** $\mathbb{D}^{nxF_s}$, $\left(x^i\right)|_{i=1}^{n}$ windowed sample.
2. **Output:** Root cause variable/variables for the diagnosed fault
3. **for** (p = 1: length(F_s))
4. Randomly shuffle the data points in F_p.
5. Generate the prediction target from shuffled data: $\hat{y}_p$
6. Generate permutated data PMAE: $PMAE = \frac{|y_x - \hat{y}_p|}{n}$
7. Calculate permutation score using Equation 6.19.
8. **end**

9. Compare the permutation score and select the features, over 90% of the maximum score.
10. Run algorithm 3, to determine the RUL due to the occurred fault condition.
11. **end**

6.3.3 Remaining Useful Life Estimation

To predict remaining useful life of the system, this article proposes a Bayesian degradation model for the selected root cause variable. The remaining run time to the failure is obtained by predicting the time interval to reach the failure margin. In the degradation model, raw sensor data are used in the model development. Once the fault has been detected and root cause variables isolated, the next two windows of samples are used to train the Bayesian regression model, where $'F'$ is a root cause variable for the diagnosed fault condition. Failure margins for all the features are predefined and stored in the algorithm. The proposed model is developed to predict the linearly increasing failure types. Algorithm 3 shows the complete process of the degradation model and RUL estimation.

Algorithm 3 RUL time estimation

1. **Input:** ($\mathbb{X}^{(2xn)x\mathrm{F}}$), $F \subset F_s$, $\left(x^i\right)|_{i=1}^{2xn}$ windowed sample and failure margin of each feature.
2. **Output:** RUL time estimation
3. Using $\left(x^i\right)|_{i=1}^{2xn}$ and run time/sample time interval of the data sample Bayesian regression parameters $\beta \; and \; \alpha$ optimized using Equation 6.14.
4. For the large sample time, the feature indexed value was predicted and tested using the failure margin.
5. Calculated intersection time plotted using Equation 6.20.
6. **End**

In this approach, the linear time degradation indicator provides the range of RUL time since Bayesian regression is used in the degradation model. However, minimum RUL time was selected as a RUL. Using the time to a fault condition, RUL is estimated using the following formulated equation:

$$RUL = \begin{cases} RUL = 1, \ if \ sample < TtFa \\ RUL_{degradation}, if \ TtFa < samples < TtF \\ RUL = 0, if \ samples > TtF \end{cases} \tag{6-20}$$

In the equation, $TtFa - Time\ to\ Fault, TtF - Time\ to\ Failure.$

6.4 Experiment and Test Result

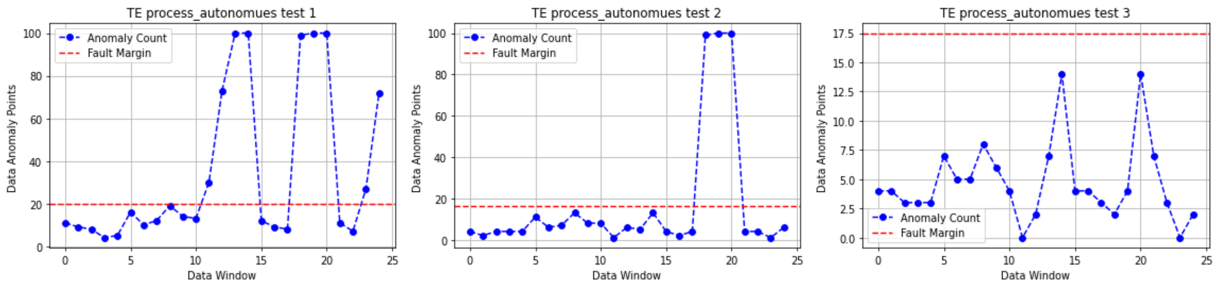
This section provides the experimental results for the case of Tennessee Eastman (TE) process data. In the TE process, the faults involved are step disturbance, random variation, slow drifting, and sticking [45]. However, step disturbance does not have a time delay for fault to failure deviation. Hence, for step disturbance, fault time and failure time will be the same. Therefore, to test the RUL prediction, random variation fault type IDV 11 and sticking fault type IDV 15 are selected from the TE fault condition. To test the proposed model, all three steps are used autonomous fault detection and diagnosis using algorithm 2(a), root cause analysis using algorithm 2(b), and RUL prediction using algorithm 3.

6.4.1 Autonomous Faults Detection and Diagnosis Test

To test autonomous FDD, TE processes nonfaulty data, IDV 15 and IDV 11 data are concatenated. $x^{(i),f_e}|_{i=1}^{1000}$ nonfaulty, $x^{(i),f_e}|_{i=1001}^{1500}$IDV 11 fault, $x^{(i),f_e}|_{i=1501}^{1800}$nonfaulty, and $x^{(i),f_e}|_{i=1801}^{2100}$IDV 15 fault, $x^{(i),f_e}|_{i=2100}^{2301}$nonfaulty, and $x^{(i),f_e}|_{i=2301}^{2500}$IDV 11 faults are concatenated to test the autonomous fault detection and diagnosis.

Initially, $x^{(i),f_e}|_{i=1}^{300}$ unsupervised samples and $x^{(i),f_e}, y^{(i),1}|_{i=1}^{300}$ supervised data samples are used to train the OCSVM and NN models, respectively, using algorithm 1. In training, the Spearman correlation test did not capture any highly correlated features among the TE process measured variables. Therefore, all sets of features are used in the experiment $\{f_e\} = \{f_s\}$.

In the online testing, concatenated data are fed to algorithms 2(a) and (2b) to test the fault condition and update the model online.



(a) (b) (c)

Figure 6.17 Self learning and autonomous update test result

Figure 6.4 shows the algorithm 2(a) online testing result. Figure 6.4(a) shows the test result with only nonfaulty data training the OCSVM, Figure 6.4(b) shows the test result after the IDV11 fault is detected at the 1000^{th} sample and the OCSVM and NN models are updated. Similarly, 6.4(c) shows the result with updated OCSVM and NN after fault IDV15 has been detected. The NN classification result and model updates are shown in Table 6.1.

Table 6-1 Incremental NN model update and prediction result

The online data sample window	**NN classification**	**Model update**
1-10	100% classified as Nonfaulty	-
10-11	-	Model update with IDV 11 fault
11-15	98% classified as fault IDV11	-
15-18	96% classified as Nonfaulty	-
18 – 19		Model update with IDV 15 fault
19 – 21	94% classified as fault IDV15	-
21-25	97% classified as Non fault	-

When updating the NN model once the new fault is detected, NN parameters θ, b are optimized from the beginning, using the Equation 6.9, and hyperparameters #iteration and learning rates are optimized by running a range of numbers.

6.4.2 Permutation Test

Once the new fault is detected, algorithm 2(b) is used to test the root cause for the data deviation and abnormalities.

As shown in Figure 6.5, the root cause for the IDV 11 fault was detected as an XMEAS 21 feature, and IDV 15 fault was detected as XMEAS 22. Results of the proposed method are compared with existing literature in Table 6.2.

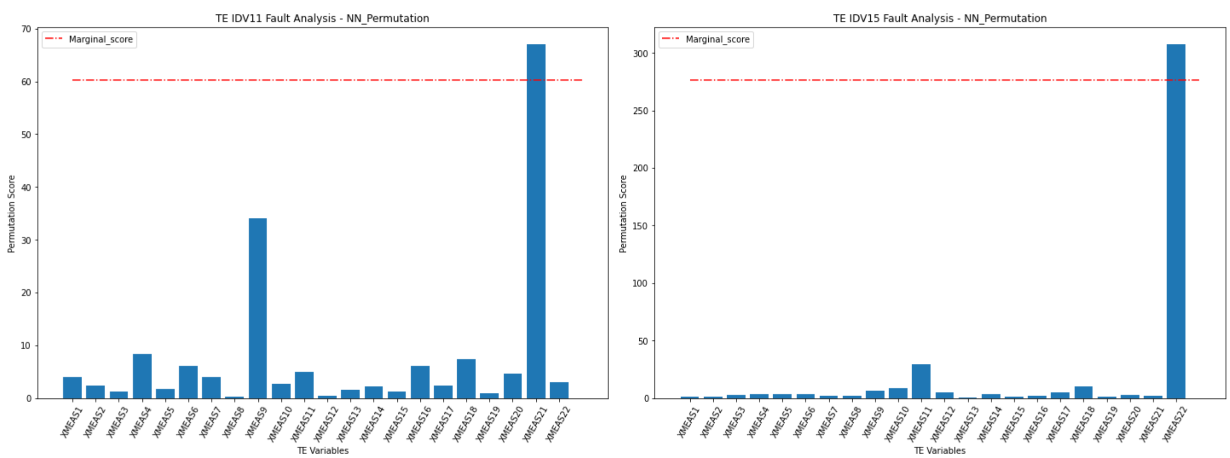


Figure 6.18 Permutation algorithm test for the root cause analysis.

Table 6-2: TE process root cause analysis with different models

Fault ID	**Amin et al. (2018)** [46]	**Rashid et al. (2018)** [47]	**Proposed method**
IDV 11	XMEAS 9	XMEAS 21, XMEAS 9	XMEAS 21
IDV 15	XMEAS 22	XMEAS 11, XMEAS 22	XMEAS 22

IDV 11 faults are described as reactor cooling water inlet temperature, and IDV 15 faults are described as condenser cooling water valves [42]. Therefore, as shown in figure 6.5 and table 6.2, the proposed method defines the root cause correctly for the tested fault condition. Also, compared to the other models, the root cause is defined autonomously without any human interactions in the proposed algorithm.

6.4.3 RUL Estimation Test

Once the root cause has been identified for the detected fault condition online, raw sample data for the root cause variable ($\mathbb{X}_{i=0}^{n,F_c}$, where $F_c \subset F_s$), in the detected window are fed into algorithm 3 to predict the remaining useful life estimation. Also, the process system operation and failure margin are accrued by industrial plant knowledge and fed into the algorithm.

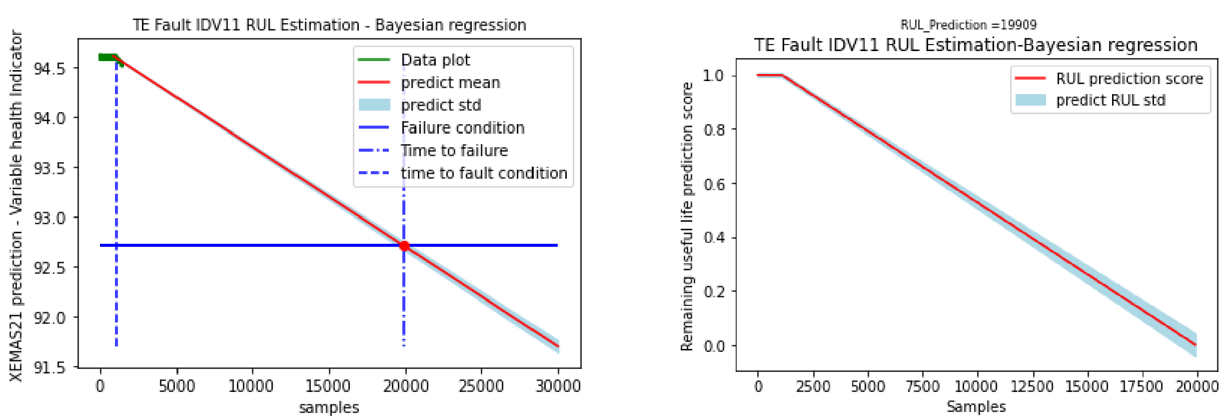
In this experiment, for the IDV11 random variation (reactor cooling water inlet temperature) fault condition, XMEAS 21 (reactor cooling water outlet temperature) has been identified as a root cause. Table 6.3 illustrates the Tennessee Eastman process variable hi (H) and low (L) alarm threshold defined by Downs and Vogel [45] and then further revised by Manca and Fay [48]. From Table 6.3, the XMEAS 21 variable failure margin has been identified as lower-level 92.71℃ and upper-level 96.49℃. Similarly, for the IDV15 fault condition (stiction in condenser cooling water valve), XMEAS 22 (condenser cooling outlet temperature) is identified as a root cause, and the failure margins for low and high level are obtained using Table 6.3, 75.75℃ and 78.84℃ respectively.

Table 6-3 : TE process variables Hi and Low failure margin (adapted from [48])

Variable	Description	L	Base	H	Unit
XMEAS1	A Feed	0.24	0.25	0.26	*kscmh*
XMEAS2	D Feed	3589	3664	3735	$kg\ h^{-1}$

XMEAS3	E Feed	4419	4509	4599	*kg h^{-1}*
XMEAS4	C Feed	9.16	9.35	9.54	*kscmh*
XMEAS5	Recycle flow	26.36	26.90	27.44	*kscmh*
XMEAS6	Reactor feed rate	41.49	42.34	43.19	*kscmh*
XMEAS7	Reactor pressure	0.0	2705	2895	*kpagague*
XMEAS8	Reactor level	50	75	100	*%*
XMEAS9	Reactor temperature	0.00	120.40	150	*^oC*
XMEAS10	Purge rate	0.33	0.34	0.34	*kscmh*
XMEAS11	Separator temperature	78.50	80.10	81.70	*^oC*
XMEAS12	Separator level	30	50	100	*%*
XMEAS13	Separator pressure	2581.03	2633.70	2686.37	*kpagague*
XMEAS14	Separator underflow	24.66	25.16	25.66	*m^3 h^{-1}*
XMEAS15	Stripper level	30.0	50.0	100.0	*%*
XMEAS16	Stripper pressure	3040.16	3102.20	3164.24	*kpagague*
XMEAS17	Stripper underflow	22.49	22.95	23.41	*m^3 h^{-1}*
XMEAS18	Stripper temperature	64.42	65.73	67.05	*^oC*
XMEAS19	Stripper steam flow	225.70	230.31	234.92	*kg h^{-1}*
XMEAS20	Compressor work	334.60	341.43	348.26	*kW*
XMEAS21	Reactor cooling water outlet temp.	92.71	94.60	96.49	*^oC*
XMEAS22	Condenser cooling outlet temp.	75.75	77.30	78.84	*^oC*

The RUL output from algorithm3 is plotted in Figure 6.6.



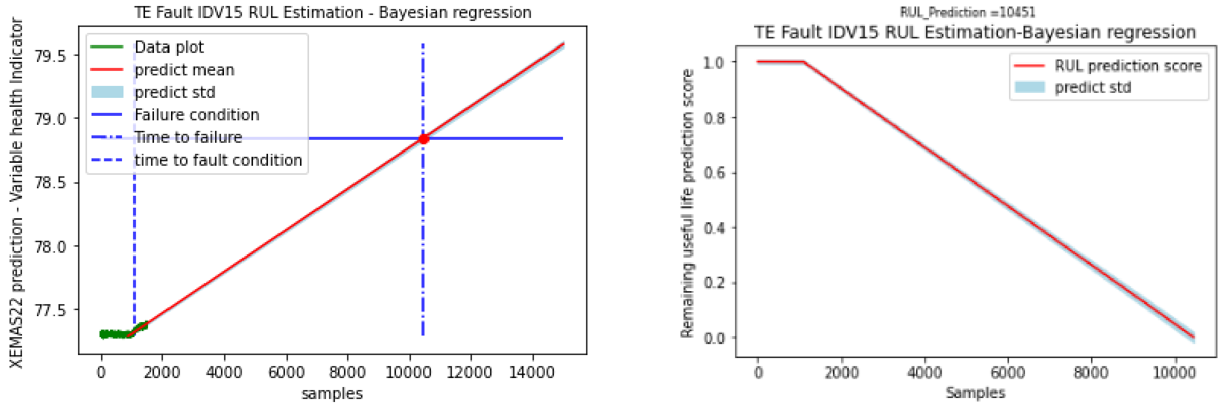


Figure 6.19 Health indicator and RUL predictor.

The remaining useful life is estimated using the fault detection point as the base point. Therefore, RUL estimation based on the fault detection is formulated as:

$$RUL_{Fault\ condition} = \tau_{Failure} - \tau_{Fault} \tag{6-21}$$

From the testing, RUL estimation for the IDV 11 and IDV 15 was estimated using Equation 6.21 and obtained 18,810 data samples and 9,451 data samples, respectively. The estimated result is compared with Ordinary least square linear regression (OLSR) and exponential regressions (ER).

Table 6-4: Result comparison (after the fault detected) in #of samples

Fault ID	**Experimental result**	**Predicted RUL (in# of samples)**		
		OLSR	**ER**	**BR (proposed method)**
IDV11	19100	19010	19470	18810
IDV15	9500	10700	10450	9451

In this approach, the Bayesian regression approach estimates the RUL with the lower value of the regression estimation. As shown in Table 6.4: in both fault conditions, BR estimates the RUL closer to the experimental result and predicted before the system failure.

6.5 Conclusion

This chapter introduces a robust methodology to estimate the RUL by online monitoring of a fault condition and estimating the time to failure margin. Self-learning OCSVM and incremental permuted NN models are used to predict the fault condition and root cause analysis online. Once the fault has been detected, the failure margin degradation model is developed and used to forecast the time to failure. Finally, using the fault time and failure predicted time, RUL is estimated. Further, these models are self-learning; therefore, this method predicts the new fault condition and root cause analysis autonomously and estimates the RUL for a detected fault.

This methodology is well fitted to the process industry, since, in a process plant, the failure margins for the measured, control, and process variables are predefined based on the process plant's operation. This article is limited with the fault to failure deviation considered as a linear relationship in the TE process failures. However, this approach can be extended for nonlinear degradation based on the nature of the failure occurring in the process plant. Also, this method can be extended with time-dependent deep learning models such as long short-term memory or autoregressive intergrade moving average.

Reference

[1] J. Lee, M. Ghaffari, and S. Elmeligy, "Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems," *Annual Reviews in Control*, vol. 35, no. 1, pp. 111–122, Apr. 2011, doi: 10.1016/J.ARCONTROL.2011.03.007.

[2] Y. Wang and M. Rausand, "Reliability analysis of safety-instrumented systems operated in high-demand mode," *Journal of Loss Prevention in the Process Industries*, vol. 32, pp.

254–264, Nov. 2014, doi: 10.1016/J.JLP.2014.09.007.

[3] A. Angelopoulos *et al.*, “Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects,” *Sensors (Switzerland)*, vol. 20, no. 1, pp. 1–34, 2020, doi: 10.3390/s20010109.

[4] Y. Peng, M. Dong, and M. J. Zuo, “Current status of machine prognostics in condition-based maintenance: A review,” *International Journal of Advanced Manufacturing Technology*, vol. 50, no. 1–4, pp. 297–313, 2010, doi: 10.1007/s00170-009-2482-0.

[5] J. Ben Ali, B. Chebel-Morello, L. Saidi, S. Malinowski, and F. Fnaiech, “Accurate bearing remaining useful life prediction based on Weibull distribution and artificial neural network,” *Mechanical Systems and Signal Processing*, vol. 56, pp. 150–172, 2015, doi: 10.1016/j.ymssp.2014.10.014.

[6] L. Biggio and I. Kastanis, “Prognostics and Health Management of Industrial Assets: Current Progress and Road Ahead,” *Frontiers in Artificial Intelligence*, vol. 3, no. November, pp. 1–24, 2020, doi: 10.3389/frai.2020.578613.

[7] O. Fink, Q. Wang, M. Svensén, P. Dersin, W. J. Lee, and M. Ducoffe, “Potential, challenges and future directions for deep learning in prognostics and health management applications,” *Engineering Applications of Artificial Intelligence*, vol. 92, p. 103678, Jun. 2020, doi: 10.1016/J.ENGAPPAI.2020.103678.

[8] J. C. Hoskins and D. M. Himmelblau, “Artificial neural network models of knowledge representation in chemical engineering,” *Computers & Chemical Engineering*, vol. 12, no. 9–10, pp. 881–890, Sep. 1988, doi: 10.1016/0098-1354(88)87015-7.

[9] R. Isermann and P. Ballé, “Trends in the application of model-based fault detection and

diagnosis of technical processes," *Control Engineering Practice*, vol. 5, no. 5, pp. 709–719, May 1997, doi: 10.1016/S0967-0661(97)00053-1.

[10] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 293–311, Mar. 2003, doi: 10.1016/S0098-1354(02)00160-6.

[11] C. Zhang, P. Lim, A. K. Qin, and K. C. Tan, "Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2306–2318, 2017, doi: 10.1109/TNNLS.2016.2582798.

[12] Y. Zhang, Y. Yang, X. Xiu, H. Li, and R. Liu, "A Remaining Useful Life Prediction Method in the Early Stage of Stochastic Degradation Process," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 6, pp. 2027–2031, 2021, doi: 10.1109/TCSII.2020.3034393.

[13] X. Li, Q. Ding, and J. Q. Sun, "Remaining useful life estimation in prognostics using deep convolution neural networks," *Reliability Engineering and System Safety*, 2018, doi: 10.1016/j.ress.2017.11.021.

[14] Z. Wang, Y. Chen, Z. Cai, Y. Gao, and L. Wang, "Methods for predicting the remaining useful life of equipment in consideration of the random failure threshold," *Journal of Systems Engineering and Electronics*, vol. 31, no. 2, pp. 415–431, 2020, doi: 10.23919/JSEE.2020.000018.

[15] L. Ren, J. Dong, X. Wang, Z. Meng, L. Zhao, and M. J. Deen, "A Data-Driven Auto-

CNN-LSTM Prediction Model for Lithium-Ion Battery Remaining Useful Life," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3478–3487, 2021, doi: 10.1109/TII.2020.3008223.

[16] S. Sayyad, S. Kumar, A. Bongale, A. M. Bongale, and S. Patil, "Estimating Remaining Useful Life in Machines Using Artificial Intelligence: A Scoping Review," *Library Philosophy and Practice*, vol. 2021, no. January, pp. 1–26, 2021.

[17] T. Gao, Y. Li, X. Huang, and C. Wang, "Data-driven method for predicting remaining useful life of bearing based on bayesian theory," *Sensors (Switzerland)*, vol. 21, no. 1, pp. 1–17, 2021, doi: 10.3390/s21010182.

[18] Y. Zheng, "Predicting Remaining Useful Life Based on Hilbert-Huang Entropy with Degradation Model," *Journal of Electrical and Computer Engineering*, 2019, doi: 10.1155/2019/3203959.

[19] X. S. Si, "An adaptive prognostic approach via nonlinear degradation modeling: Application to battery data," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5082–5096, Aug. 2015, doi: 10.1109/TIE.2015.2393840.

[20] X. S. Si, W. Wang, C. H. Hu, and D. H. Zhou, "Remaining useful life estimation – A review on the statistical data driven approaches," *European Journal of Operational Research*, vol. 213, no. 1, pp. 1–14, Aug. 2011, doi: 10.1016/J.EJOR.2010.11.018.

[21] X. Wang, C. Hu, X. Si, Z. Pang, and Z. Ren, "An Adaptive Remaining Useful Life Estimation Approach for Newly Developed System Based on Nonlinear Degradation Model," *IEEE Access*, vol. 7, pp. 82162–82173, 2019, doi: 10.1109/ACCESS.2019.2924148.

[22] H. Wu and J. Zhao, “Fault detection and diagnosis based on transfer learning for multimode chemical processes,” *Computers & Chemical Engineering*, vol. 135, p. 106731, Apr. 2020, doi: 10.1016/J.COMPCHEMENG.2020.106731.

[23] R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, “An analysis of process fault diagnosis methods from safety perspectives,” *Computers and Chemical Engineering*, vol. 145. Elsevier Ltd, p. 107197, Feb. 01, 2021, doi: 10.1016/j.compchemeng.2020.107197.

[24] Z. Li, Y. Wang, and K. S. Wang, “Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario,” *Advances in Manufacturing*, vol. 5, no. 4, pp. 377–387, 2017, doi: 10.1007/s40436-017-0203-8.

[25] S. Sajid, A. Haleem, S. Bahl, M. Javaid, T. Goyal, and M. Mittal, “Data science applications for predictive maintenance and materials science in context to Industry 4.0,” *Materials Today: Proceedings*, vol. 45, pp. 4898–4905, 2021, doi: 10.1016/J.MATPR.2021.01.357.

[26] J. Yang, S. Yin, Y. Chang, T. Gao, and C. Yang, “An Efficient Method for Monitoring Degradation and Predicting the Remaining Useful Life of Mechanical Rotating Components,” *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021, doi: 10.1109/TIM.2020.3032218.

[27] C. Cheng *et al.*, “A Deep Learning-Based Remaining Useful Life Prediction Approach for Bearings,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1243–1254, 2020, doi: 10.1109/TMECH.2020.2971503.

[28] S. Khan and T. Yairi, “A review on the application of deep learning in system health management,” *Mechanical Systems and Signal Processing*, vol. 107, pp. 241–265, Jul.

2018, doi: 10.1016/J.YMSSP.2017.11.024.

[29] Y. Wang, Y. Zhao, and S. Addepalli, “Remaining useful life prediction using deep learning approaches: A review,” *Procedia Manufacturing*, vol. 49, no. 2019, pp. 81–88, 2020, doi: 10.1016/j.promfg.2020.06.015.

[30] L. Liu, Q. Guo, D. Liu, and Y. Peng, “Data-Driven Remaining Useful Life Prediction Considering Sensor Anomaly Detection and Data Recovery,” *IEEE Access*, vol. 7, pp. 58336–58345, 2019, doi: 10.1109/ACCESS.2019.2914236.

[31] S. Alam, S. K. Sonbhadra, S. Agarwal, and P. Nagabhushan, “One-class support vector classifiers: A survey,” *Knowledge-Based Systems*, vol. 196, p. 105754, 2020, doi: 10.1016/j.knosys.2020.105754.

[32] B. Lamrini, A. Gjini, S. Daudin, F. Armando, P. Pratmarty, and L. Travé-Massuyès, “Anomaly detection using similarity-based one-class svm for network traffic characterization,” *CEUR Workshop Proceedings*, vol. 2289, 2018.

[33] Y. Xiao, H. Wang, W. Xu, and J. Zhou, “Robust one-class SVM for fault detection,” *Chemometrics and Intelligent Laboratory Systems*, vol. 151, pp. 15–25, 2016, doi: 10.1016/j.chemolab.2015.11.010.

[34] R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, “A deep learning model for process fault prognosis,” *Process Safety and Environmental Protection*, vol. 154, pp. 467–479, 2021, doi: 10.1016/j.psep.2021.08.022.

[35] R. Arunthavanathan, F. Khan, S. Ahmed, S. Imtiaz, and R. Rusli, “Fault detection and diagnosis in process system using artificial intelligence-based cognitive technique,” *Computers and Chemical Engineering*, vol. 134, 2020, doi:

10.1016/j.compchemeng.2019.106697.

[36] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: A corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010, doi: 10.1093/bioinformatics/btq134.

[37] N. Huang, G. Lu, and D. Xu, "A Permutation Importance-Based Feature Selection Method for Short-Term Electricity Load Forecasting Using Random Forest," *Energies 2016, Vol. 9, Page 767*, vol. 9, no. 10, p. 767, Sep. 2016, doi: 10.3390/EN9100767.

[38] Y. T. Chae, R. Horesh, Y. Hwang, and Y. M. Lee, "Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings," *Energy and Buildings*, vol. 111, pp. 184–194, Jan. 2016, doi: 10.1016/J.ENBUILD.2015.11.045.

[39] M. Lucke, M. Chioua, C. Grimholt, M. Hollender, and N. F. Thornhill, "On improving fault detection and diagnosis using alarm-range normalisation," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 1227–1232, Jan. 2018, doi: 10.1016/J.IFACOL.2018.09.695.

[40] B. Wang, Y. Lei, N. Li, and N. Li, "A Hybrid Prognostics Approach for Estimating Remaining Useful Life of Rolling Element Bearings," *IEEE Transactions on Reliability*, vol. 69, no. 1, pp. 401–412, Mar. 2020, doi: 10.1109/TR.2018.2882682.

[41] J. Jegminat, M. A. Jastrzębowska, M. V. Pachai, M. H. Herzog, and J. P. Pfister, "Bayesian regression explains how human participants handle parameter uncertainty," *PLOS Computational Biology*, vol. 16, no. 5, p. e1007886, May 2020, doi: 10.1371/JOURNAL.PCBI.1007886.

[42] D. Sun, J. Xu, H. Wen, and D. Wang, "Assessment of landslide susceptibility mapping based on Bayesian hyperparameter optimization: A comparison between logistic

regression and random forest," *Engineering Geology*, vol. 281, p. 105972, Feb. 2021, doi: 10.1016/J.ENGGEO.2020.105972.

[43] M. Kitai, T. Kobayashi, H. Fujiwara, R. Tani, M. Numao, and K. I. Fukui, "A Framework for Predicting Remaining Useful Life Curve of Rolling Bearings under Defect Progression Based on Neural Network and Bayesian Method," *IEEE Access*, vol. 9, pp. 62642–62652, 2021, doi: 10.1109/ACCESS.2021.3073945.

[44] M. Saqib, "Forecasting COVID-19 outbreak progression using hybrid polynomial-Bayesian ridge regression model," *Applied Intelligence*, vol. 51, no. 5, pp. 2703–2713, May 2021, doi: 10.1007/S10489-020-01942-7/TABLES/6.

[45] J. J. Downs and E. F. Vogel, "A Plant-wide Industrial Problem Process," *Computers & Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993, doi: 10.1016/0098-1354(93)80018-I.

[46] M. T. Amin, S. Imtiaz, and F. Khan, "Process system fault detection and diagnosis using a hybrid technique," *Chemical Engineering Science*, vol. 189, pp. 191–211, Nov. 2018, doi: 10.1016/j.ces.2018.05.045.

[47] B. Rashidi, D. S. Singh, and Q. Zhao, "Data-driven root-cause fault diagnosis for multivariate non-linear processes," *Control Engineering Practice*, vol. 70, pp. 134–147, Jan. 2018, doi: 10.1016/j.conengprac.2017.09.021.

[48] G. Manca and A. Fay, "Detection of Historical Alarm Subsequences Using Alarm Events and a Coactivation Constraint," *IEEE Access*, vol. 9, pp. 46851–46873, 2021, doi: 10.1109/ACCESS.2021.3067837.

Chapter 7 Conclusions and Future Research

This research work aimed to investigate and apply machine learning approaches to detect and diagnose a fault and identify its root cause for the digitized process system in an autonomous environment, and to estimate the remaining useful life by analyzing fault to failure transition online. We have the following conclusions and recommendations for future work.

7.1 Conclusions

The result indicates that compared to statistical analysis, machine learning approaches are better suited for developing intelligent fault detection and diagnosis techniques compatible with Industry 4.0 and the future. The following specific conclusion of the thesis, and all of the mentioned findings in the conclusion, are tested and validated using experimental process data and/or simulated process data from benchmark applications, such as the Tennessee Eastman process system and the continuous stirred tank heater model.

7.1.1. Development of Autonomous FDD Using Machine Learning Techniques

Applying anomaly detection approaches such as OC- SVM and OC- NN may help to investigate the unlabelled or unsupervised process data, and to classify the known fault condition, while most of the other machine learning models need to be trained with labelled data. Therefore, to develop autonomous FDD approaches, integrating anomaly detection approaches with classification models is necessary and can provide a desired outcome.

7.1.2. Development of Self Learning Model for Cognitive FDD Using Machine Learning Algorithms

It was found that machine learning models can be updated using appropriate training data in real-time. Therefore, appropriately trained self-learning models and online updating may be used for autonomous fault detection. In this research work, an anomaly detection algorithm was updated with a detected fault condition online to further investigate the new fault condition. A shallow neural network with incrementing output nodes is proposed to train the NN model with the number of detected fault conditions.

7.1.3. Development of Root Cause Analysis Model Using Machine Learning Approach

In diagnosing the root cause of a fault, a shallow neural network with a permutation algorithm was found to be effective. The proposed approach was developed to identify the important variable for the classified fault condition. Therefore, it is important to identify the correlation of each variable in the process data.

7.1.4. Development of Early Fault Predicting Model Using Machine Learning Approaches

This research also demonstrates that, once the fault symptoms are observable, they may be further investigated to diagnose and prognose the fault condition. In this study, an integrated convolutional neural network and long short-term memory model were used to predict the future multivariate progression of variables, and a one class anomaly detection approach was applied to estimate the fault condition using the predicted synthetic multivariate data.

7.1.5 Development of RUL Estimation Model Using Machine Learning Algorithm

Finally, this research shows that, for the slow drift fault to failure transmission, it is possible to estimate the RUL using a combination of FDD methods and Bayesian regression estimation. In this case, the root cause of the fault condition and failure margin of the variable must be known.

7.2 Contributions

To summarize, this thesis made the following key contributions in applying machine learning models to process systems to develop automated fault detection and diagnosis and prognosis methodologies.

1. The importance of applying machine learning models in fault detection and diagnosis linked to process safety is comprehensively reviewed and further direction suggested based on the needs in process plant safety perspective to the development of Industry 4.0. The review article has been published in the Computers and Chemical Engineering Journal.
2. Cognitive fault detection and a fault classification method were implemented to detect the unknown fault and classify the known fault condition. A self-updating approach was formulated to autonomously detect the fault in the future operation. The article with the developed methodology and demonstration results was published in the Computers and Chemical Engineering journal and implemented model open-sourced in GitHub platform for forthcoming use.
3. A stand-alone FDD method was developed to detect, diagnose, and identify the root cause for the detected fault condition in the autonomous environment using self-updating machine learning models. The methodology and the results were published in the Industrial and Engineering Chemistry Research Journal and the implemented model has been open sourced in the GitHub platform for further developers.
4. A Fault prognosis model was developed using a deep neural network to forecast the fault condition using prior fault symptoms. The results were published in the Process Safety and Environmental Protection Journal and the implemented deep neural network model open-sourced in the GitHub platform.

5. Online RUL estimation method was developed by analyzing the fault to failure transmission of the diagnosed root cause variable. Results were submitted to the IEEE access Journal for publication and the developed algorithm/program open-sourced in Github for further development.

7.3 Future Work Direction

There are many possible expansions of the works provided in this thesis. These future developments will be examined for more practical implications regarding the development of FDD and predictive maintenance implemented in the future plant industry.

1. **Further Develop the Model to Diagnose Multiple Faults Simultaneously**

   It was shown that the proposed machine learning approaches can detect, diagnose, and define root cause for the process system online. Further, the development of self-learning models was examined by updating the detected fault conditions over time. However, when more than one fault occurred during the data sampling time in the proposed model, it was defined as a single fault condition. This can be addressed in future by integrating a model-based algorithm, such as a Kalman filter or extended Kalman filter, with the proposed data driven models.

2. **Development of Root Cause Analysis by Integrating Traditional Statistical Algorithms**

   In the root cause analysis approach, results are highly dependent on the correlation test before the data is sent to the machine learning (NN permutation) model. However, an alternative solution can be considered, integrating statistical approaches such as the dynamic Bayesian approach with the proposed artificial neural network approaches.

3. **Implement Predictive Maintenance Approaches Before System Leads to Failure Or Shutdown**

   The last objective of the proposed work, estimation of the RUL by analyzing the fault to failure transmission using an alarm/consequence margin, can be implemented further by developing the deep learning models to predict the degradation and estimate the RUL.

   Using the estimated RUL, predictive maintenance can be planned or properly applied to prevent the system from reaching the failure condition.

4. **Integration With Cyber Physical System**

   Most of the proposed approaches in this study demand a large amount of good quality data. Therefore, the algorithms are tested using simulated and experimental sensor data. However, with the recent industrial revolution, cyber physical systems have been invented in the process industry with the integration of networks, computation, and physical processes. Therefore, FDD is not limited to the sensor actuator and process; it needs to consider communication channels and protocols.

   Also, if the data transmission is fully online, it is important to protect the process data from cyber-attack. Therefore, when applying developed models to cyber physical concepts, failures due to the communications channels and components, along with failure due to cyber-attack, need to be considered.

5. **Computational Time**

   This work used intel core i5 and i7 processors to develop and test the model. Nevertheless, when implanting these models in the plant, it is required to reduce the computational time to accurately detect the fault in online conditions. Therefore, to compute a large number of data, high performance artificial intelligent processors (such as TensorFlow processors) are suggested in the implementation.

6. **Investigating an Estimated Uncertainty Modeling**

   Machine learning fault and failure margins are used in this work, based on the experimental margin. This may vary with different applications. Therefore, a more formal and standard uncertainty modelling study for the machine learning algorithm is recommended to ensure consideration of all sources of uncertainty when applying the proposed model.