APPLICATION OF MACHINE HEALTH
MONITORING IN DESIGN OPTIMIZATION OF
MECHATRONIC SYSTEMS

by

Min Xia

B.Eng., Southeast University, 2009
M.Eng., University of Science and Technology of China, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL
STUDIES
(Mechanical Engineering)

The University of British Columbia
(Vancouver)

September 2017

© Min Xia, 2017
Abstract

Mechatronic systems are widely used in modern manufacturing. The key machinery of a manufacturing system should be reliable, flexible, intelligent, less complex, and cost effective, which indeed are distinguishing features of a mechatronic system. To achieve these goals, continuous or on-demand design improvements should be incorporated rapidly and effectively, which will address new design requirements or resolve existing weaknesses of the original design.

With the advances in sensor technologies, wireless communication, data storage, and data mining, machine health monitoring (MHM) has achieved significant capabilities to monitor the performance of an operating machine. The extensive data from the MHM system can be employed in design improvement of the monitored system. In that context, the present dissertation addresses several challenges in applying MHM in design optimization of a mechatronic system.

First, this dissertation develops a systematic framework for continuous design evolution of a mechatronic system with MHM. Possible design weaknesses of the monitored system are detected using the information from MHM. The proposed method incorporates an index to identify a possible design weakness by evaluating the performance, detecting failures and estimating the health status of the system.

Second, improved approaches of intelligent machine fault diagnosis (IMFD) that can be applied to more general machinery and faults, are presented. This dissertation develops an IMFD approach based on deep neural networks (DNN). It uses the massive unlabeled MHM data to learn representative features. Using very few items of labeled data, this approach can achieve superior diagnosis performance. The dissertation presents another IMFD approach, which uses the convolutional neural networks (CNN) and sensor fusion and has increased diagnosis
accuracy and reliability. The end-to-end learning capability of the two approaches enables diagnosis of fault types or machines for which limited prior knowledge is available.

Third, a hierarchical DNN-based method of remaining useful life (RUL) prediction is developed. It achieves high accuracy of RUL prediction by modeling the system degradation on different health stages. This method generates a better estimate of the system RUL, which provides accurate information for the evaluation of system design.
Lay Summary

Reliability, efficiency, and flexibility of a manufacturing system is important in modern industries. Mechatronic systems are widely used as key components. Continuous or on-demand design improvements should be incorporated effectively. Machine health monitoring (MHM) has achieved significant capabilities to monitor the performance of an operating machine. The extensive data that may be acquired can be employed in design improvement of the monitored system. This dissertation addresses several challenges in applying MHM in design optimization of a mechatronic system. First, a systematic and closed-loop framework is developed for continuous design evolution of a mechatronic system with MHM. Second, improved approaches of machine fault diagnosis based on deep neural networks (DNN) and sensor fusion that can be applied to more general machinery and faults, are presented. Third, an improved remaining useful life (RUL) prediction method is developed, which will provide accurate information for the evaluation of system design.
Preface

All the work presented in this dissertation was conducted by Min Xia in the Industrial Automation Laboratory (IAL) at the University of British Columbia, Vancouver campus, under the direct supervision and guidance of Dr. Clarence W.de Silva, Professor of Mechanical Engineering, The University of British Columbia. Dr. de Silva proposed and supervised the overall research project, acquired funding and resources for the project, suggested the topic of the thesis, suggested concepts and methodologies in addressing problems in the topic, provided research facilities in his IAL, and revised the thesis presentation.

Chapter 2 is based on the publications [Min Xia and Clarence W. de Silva, ”A Framework of Design Weakness Detection through Machine Health Monitoring for the Evolutionary Design Optimization of Multi-domain Systems,” IEEE International Conference in Computer Science & Education (ICCSE), Vancouver, August 22-24, pp. 205-210, 2014.] and [Min Xia, Teng Li, Yunfei Zhang, and Clarence W. de Silva, Closed-loop Design Evolution of Engineering System Using Condition Monitoring through Internet of Things and Cloud Computing, Computer Networks, vol. 101, pp. 5-18, 2016]. Min Xia was responsible for all major areas of concept formation, algorithm development, experiment validation, as well as manuscript composition. Teng Li and Yunfei Zhang was involved in the early stages of concept formation and contribution to manuscript editing. Clarence W. de Silva was the supervisory author on this work and was involved throughout the project in concept formation and manuscript composition.

A version of Chapter 3 has been published [Min Xia, Teng Li, Lizhi Liu, Lin Xu, Clarence de Silva, An Intelligent Fault Diagnosis Approach with Unsupervised Feature Learning by Stacked Denoising Autoencoder, IET Science, Measurement
Min Xia was responsible for all major areas of concept formation, algorithm development, experiment validation, as well as manuscript composition. Teng Li was involved in the early stages of concept formation and contributed to manuscript editing. Lizhi Liu and Lin Xu were involved in the early stage of concept formation. Clarence W. de Silva was the supervisory author and was involved throughout the work in concept formation, guidance, and manuscript composition.

A version of Chapter 4 has been published [Min Xia, Teng Li, Lin Xu, Lizhi Liu, Clarence W. de Silva, "Fault Diagnosis for Rotating Machinery Using Multiple Sensors and Convolutional Neural Networks", IEEE/ASME Transactions on Mechatronics, vol. PP, no. 99, pp. 1-1. doi: 10.1109/TMECH.2017.2728371]. Min Xia was responsible for all major areas of concept formation, algorithm development, experiment validation, as well as manuscript composition. Teng Li was involved in the early stages of concept formation and contributed to manuscript editing. Lizhi Liu and Lin Xu were involved in the early stage of concept formation. Clarence W. de Silva was the supervisory author on this project and was involved throughout the project in concept formation, guidance, and manuscript composition.

A version of Chapter 5, [Min Xia, Teng Li, Lizhi Liu, Lin Xu, Clarence W. de Silva, A Hierarchical DNN-based Approach for RUL Prediction], has been accepted by 2017 IEEE International Conference on Systems, Man, and Cybernetics. Min Xia was responsible for all major areas of concept formation, algorithm development, experiment validation, as well as manuscript composition. Teng Li was involved in the early stages of concept formation and contributed to manuscript editing. Lizhi Liu and Lin Xu were involved in the early stage of concept formation and algorithm development. Clarence W. de Silva was the supervisory author and was involved throughout the work in concept formation, guidance, and manuscript composition.
Table of Contents

Abstract .......................................................... ii
Lay Summary ...................................................... iv
Preface ............................................................... v
Table of Contents ................................................ vii
List of Tables ....................................................... x
List of Figures ...................................................... xii
Nomenclature ...................................................... xv
List of Acronyms ................................................ xvii
Acknowledgments ................................................ xx

1 Introduction ...................................................... 1
  1.1 Motivation .................................................. 1
  1.2 Research Objectives ........................................ 4
  1.3 Related Work ............................................... 4
    1.3.1 Evaluation of Mechatronic Systems .................... 4
    1.3.2 Evolutionary Design Optimization ...................... 6
    1.3.3 Machine Health Monitoring ......................... 9
    1.3.4 Deep Neural Networks ............................. 12
1.4 Contributions and Organization of the Dissertation .......................... 13

2 Closed-loop Design Evolution of Mechatronic Systems ........................ 16
  2.1 Introduction ................................................................................. 16
  2.2 System Framework ................................................................. 17
  2.3 Condition Monitoring through IoT and CC ................................. 19
    2.3.1 Data Transmission .............................................................. 20
    2.3.2 Data Mining ........................................................................ 21
    2.3.3 Design Knowledge Base through CC ................................. 22
  2.4 Design Weakness Detection ...................................................... 22
  2.5 Mechatronic Design Quotient ................................................... 25
  2.6 Case Study ................................................................................. 28
    2.6.1 Machine Condition Monitoring ........................................... 28
    2.6.2 Conceptual Design Stage .................................................... 30
    2.6.3 Detailed Design Stage ........................................................ 37

3 Intelligent Machine Fault Diagnosis using Unsupervised Feature Learning ........................................................................ 39
  3.1 Introduction ................................................................................. 39
  3.2 Unsupervised Feature Learning .................................................. 41
    3.2.1 Autoencoder ......................................................................... 41
    3.2.2 Denoising Autoencoder ....................................................... 43
    3.2.3 Softmax Regression Classifier ............................................ 44
  3.3 Feature Learning and Fault Diagnosis using SDA ....................... 44
    3.3.1 Unsupervised Feature Learning with SDA .......................... 46
    3.3.2 Fine-tuning a DNN for Classification .................................. 47
  3.4 Experiment Studies .................................................................... 48
    3.4.1 Data Description ................................................................. 48
    3.4.2 Fault Diagnosis using the Proposed Approach .................... 50
    3.4.3 Effect of the Size of Labeled Data ...................................... 54
    3.4.4 Visualization of Learned Features ...................................... 56
4 IMFD using Convolutional Neural Networks and Sensor Fusion . . 58
  4.1 Introduction ............................................. 58
  4.2 Convolutional Neural Networks .......................... 59
    4.2.1 Convolution Layer .................................. 60
    4.2.2 Feature Pooling Layer .............................. 61
    4.2.3 Softmax Layer ..................................... 61
    4.2.4 Mini-batch Stochastic Gradient Descent .............. 62
  4.3 IMFD using CNN and Sensor Fusion ........................ 63
  4.4 Experimental Studies .................................... 65
    4.4.1 Bearing Fault Diagnosis ............................ 65
    4.4.2 Gearbox Fault Diagnosis ............................ 71

5 RUL Prediction using Hierarchical Deep Neural Network ........ 79
  5.1 Introduction ............................................. 79
  5.2 Health Condition Classification using DNN .................. 82
  5.3 ANN-based RUL Predictor and Smoothing Operator .......... 84
  5.4 Experimental Study ...................................... 87
    5.4.1 Data Description .................................... 87
    5.4.2 RUL Prediction Using DNNRULP ...................... 87

6 Conclusions and Future Work .................................. 93
  6.1 Conclusions ............................................. 93
  6.2 Possible Future Work .................................... 94

Bibliography .................................................. 96
List of Tables

Table 2.1 Subsystems of the automated fish cutting system . . . . . . . . 31
Table 2.2 Specifications of the current subsystems and the estimated de-
sign requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
Table 2.3 Available choices for each subsystem . . . . . . . . . . . . . . 33
Table 2.4 Typical interactions between criteria . . . . . . . . . . . . . . 35
Table 2.5 MDQ evaluation of design alternatives . . . . . . . . . . . . . . 36
Table 3.1 Conditions of the bearing in the dataset . . . . . . . . . . . . . 50
Table 3.2 Classification results with different levels of noise . . . . . . . 53
Table 3.3 Comparison of the proposed approach with SVM and kNN . . 54
Table 4.1 Conditions of the bearing in the dataset . . . . . . . . . . . . . 69
Table 4.2 Hyper parameters of the convolution layers . . . . . . . . . . . 69
Table 4.3 Hyper parameters of the max-pooling layers . . . . . . . . . . 69
Table 4.4 Diagnosis accuracy of bearing data using multiple sensors and
one sensor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
Table 4.5 Features selected in the time and frequency domains . . . . . . 71
Table 4.6 Comparison of bearing fault diagnosis results using different
approaches . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
Table 4.7 Gearbox dataset details . . . . . . . . . . . . . . . . . . . . . 74
Table 4.8 Diagnosis accuracy of gearbox data using multiple sensors and
one sensor . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
Table 4.9 Comparison of gearbox fault diagnosis results using different
approaches . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77
Table 5.1  The operation conditions of the experiment . . . . . . . . . . . 88
Table 5.2  Training and testing result of the DNNHSC . . . . . . . . . . . 90
Table 5.3  RUL prediction error using the HDNNRULP . . . . . . . . . . . 91
List of Figures

Figure 1.1 System framework of design evolution with machine health monitoring. ........................................... 11

Figure 2.1 Framework of evolutionary design improvement incorporating MHM. .............................................. 18

Figure 2.2 Framework for closed-loop design evolution of an engineering system. ............................................. 19

Figure 2.3 Proposed scheme of condition monitoring through the IoT and CC. .................................................. 20

Figure 2.4 Procedure of design weakness detection. ................................................................. 24

Figure 2.5 Schematic diagram of the automated fish cutting system. ......................................................... 29

Figure 2.6 Reduced conceptual design space. ............................................. 34

Figure 2.7 Procedure of evolutionary design with GP. ................................................................. 38

Figure 3.1 The structure of the autoencoder. ................................................................. 42

Figure 3.2 Schematic diagram of the procedure of denoising autoencoder. ........................................ 43

Figure 3.3 The scheme of the SDA based feature learning and fault diagnosis. ........................................ 46

Figure 3.4 The procedure of stacking denoising autoencoder. ......................................................... 47

Figure 3.5 Fine-tuning of the DNN by minimizing the error in predicting the supervised target. ........................................ 48

Figure 3.6 Experiment setup for bearing condition data acquisition. ..................................................... 49

Figure 3.7 The structure of datasets. ................................................................. 51

Figure 3.8 Classification results with the proposed approach ................................................................. 52
Figure 3.9 Effect of the size of labeled data ........................................... 55
Figure 3.10 Visualization of learned features ................................. 57
Figure 4.1 Flowchart of the proposed fault diagnosis approach. ......... 63
Figure 4.2 Architecture of the CNN-based fault diagnosis model ...... 64
Figure 4.3 Vibration signals of bearings with different conditions from
one sensor. (a) BD 0.18 mm. (b) BD 0.36 mm. (c) BD 0.53
mm. (d) OR 0.18 mm. (e) OR 0.36 mm. (f) OR 0.53 mm. (g)
IR 0.18 mm. (h) IR 0.36 mm. (i) IR 0.53 mm. ............................ 66
Figure 4.4 Experimental result of bearing dataset. (a) Convergence curve
of the training process. (b) Condition classification confusion
matrix. .......................................................................................... 68
Figure 4.5 Experimental setup for gearbox fault diagnosis. (a) Conveyor
system of a fish processing machine; (b) Three faulty gear-
boxes; (c) Accelerometers mounted on the gearbox; (d) Na-
tional Instruments PXIe DAQ system. ................................. 72
Figure 4.6 Vibration signals of gearboxes with different conditions from
one sensor. (a) Normal condition. (b) Damaged gear. (c) Dam-
aged bearing. (d) Misaligned output shaft. ............................... 74
Figure 4.7 Experimental result of gearbox dataset. (a) Convergence curve
of the training process. (b) Condition classification confusion
matrix. .......................................................................................... 76
Figure 5.1 Degradation process of a bearing. (a) Vibration signal of the
run-to-failure process. (b) RMS of the vibration signal. ....... 81
Figure 5.2 Flowchart of the DNNRULP method. ......................... 82
Figure 5.3 Five stages of the degradation process. ....................... 83
Figure 5.4 Structure of the DNN used in the DNNHSC ................. 84
Figure 5.5 Flowchart of the health stage classification. ............... 85
Figure 5.6 Formation of the training samples to each ANN RULP. 85
Figure 5.7 ANN model for RUL prediction. ............................... 86
Figure 5.8 PRONOSTIA bearing test setup. ................................. 88
Figure 5.9 Vibration signal acquisition scheme ......................... 89
Figure 5.10  RUL prediction result using the HDNNRULP. . . . . . . . . .  90
Figure 5.11  Comparison of the RUL prediction results. . . . . . . . . . .  92
Nomenclature

\( a \)  Mobius transform
\( b \)  bias vector
\( \mathbf{B} \)  bias matrix
\( c \)  constraint sanctification function
\( C_v \)  Choquet integral
\( d \)  a design alternative
\( \eta \)  learning rate
\( \bar{e} \)  average prediction error
\( E \)  RUL evaluation function
\( f \)  feed-forward process
\( f_\theta \)  deterministic mapping
\( g_i \)  fault indicating function
\( g_\theta \)  deterministic mapping
\( \mathbf{H} \)  historical run-to-failure data
\( J \)  cost function of softmax regression
\( L \)  loss function
\( \mathbf{L}_P \)  actual life percentage
\( \hat{\mathbf{L}}_P \)  estimated life percentage
\( M \)  aggregation operator
\( \mathbf{p} = [p_1, p_2, \ldots, p_r]^T \)  performance aspects
\( q_D \)  stochastic mapping
\( R \)  RUL prediction function
\( \text{RMS} \)  root mean square value
\( \text{RUL} \)  remaining useful life value
\( s \)  activation function
\( S \)  performance satisfaction function
\( t \)  labels of samples
\( t \)  age value
\[ \vec{T} = [t_1, t_2, \ldots, t_r]^T \]

- \( \vec{T} \): estimated RUL of components
- \( v \): fuzzy measure
- \( \vec{W} \): weighting vector
- \( \mathbf{W} \): weighting matrix
- \( x_i^d \): partial score of degree of satisfaction of the \( i \)th criterion
List of Acronyms

AI  artificial intelligence
ANN  artificial neural networks
ANNRULP  ANN-based RUL predictor
BD  ball defect
BGs  Bond Graphs
CC  cloud computing
CNC  Computer Numerical Control
CNN  convolutional neural networks
CWRU  Case Western Reserve University
DB  damaged bearing
DBN  Deep Belief Nets
DFP  Davidon-Fletcher-Powell
DG  damaged gear
DNN  deep neural networks
DNNHSC  DNN-based health stage classifier
DNNRULP  DNN-based RUL prediction
DPCA  dynamic principal component analysis
DWCI  Design Weakness Candidate Index
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT</td>
<td>fast Fourier transform</td>
</tr>
<tr>
<td>GA</td>
<td>genetic algorithm</td>
</tr>
<tr>
<td>GP</td>
<td>genetic programming</td>
</tr>
<tr>
<td>IAL</td>
<td>Industrial Automation Laboratory</td>
</tr>
<tr>
<td>ICA</td>
<td>independent component analysis</td>
</tr>
<tr>
<td>IMFD</td>
<td>intelligent machine fault diagnosis</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of things</td>
</tr>
<tr>
<td>IPv4</td>
<td>Internet Protocol version 4</td>
</tr>
<tr>
<td>IPv6</td>
<td>Internet Protocol version 6</td>
</tr>
<tr>
<td>IR</td>
<td>inner race defect</td>
</tr>
<tr>
<td>kNN</td>
<td>k-nearest neighbors</td>
</tr>
<tr>
<td>LGs</td>
<td>Linear Graphs</td>
</tr>
<tr>
<td>MDQ</td>
<td>mechatronic design quotient</td>
</tr>
<tr>
<td>MDI</td>
<td>mechatronic design indicator</td>
</tr>
<tr>
<td>MHM</td>
<td>machine health monitoring</td>
</tr>
<tr>
<td>MOS</td>
<td>misaligned output shaft</td>
</tr>
<tr>
<td>NLDOP</td>
<td>nonlinear dynamic optimization problem</td>
</tr>
<tr>
<td>OR</td>
<td>outer race defect</td>
</tr>
<tr>
<td>PCA</td>
<td>principal component analysis</td>
</tr>
<tr>
<td>PLS</td>
<td>partial least squares</td>
</tr>
<tr>
<td>ReLU</td>
<td>rectified linear unit</td>
</tr>
<tr>
<td>RGB</td>
<td>red-green-blue</td>
</tr>
<tr>
<td>RMS</td>
<td>root mean square</td>
</tr>
<tr>
<td>RUL</td>
<td>remaining useful life</td>
</tr>
</tbody>
</table>
SAN  Sensor Area Network
SDA  stacked denoising autoencoder
SNR  signal-to-noise ratio
SVM  support vector machine
\textbf{t-SNE}  t-distributed Stochastic Neighbor Embedding
Acknowledgments

I wish to express my sincere gratitude to my supervisor, Dr. Clarence W. de Silva for his invaluable advice, guidance, strong support, and great patience throughout my academic program and research at the University of British Columbia. I deeply appreciate him offering me the opportunity to pursue my PhD studies at UBC under his supervision. He has provided me countless opportunities and support for my academic advances as well as future career development. Without his advice, guidance, support, encouragement and supervision, I would not have been possible to achieve the academic goals throughout my PhD program.

This research has been supported by several research grants, from the Natural Sciences and Engineering Research Council of Canada, the Canada Foundation for Innovation, the British Columbia Knowledge Development Fund, Mitacs Accelerate Program, China Scholarship Council and the Tier 1 Canada Research Chair in Mechatronics and Industrial Automation held by Dr. Clarence W. de Silva.

I would like to also thank all my colleagues in Industrial Automation Laboratory at UBC, particularly, Dr. Roland Lang, Dr. Yanjun Wang, Dr. Yunfei Zhang, Dr. Muhammad Tufail Khan, Ms. Yu Du, Ms. Lili Meng, Mr. Shan Xiao, Mr. Hani Balkhair, Mr. Shujun Gao, Mr. Teng Li, Mr. Zhuo Chen, Mr. Tongxin Shu and Mr. Jiahong Chen. I thank all of them for their support, help and friendship, which makes my PhD life less difficult and more joyful.

I would like to express my deepest gratitude to my family. I thank my parents and parents-in-law for their love and encouragement throughout my PhD journey. I thank my daughter Julie, whose birth was such a great joy during my PhD study. Lastly but most importantly, I wish to thank my wife for her strong support and great love to me throughout my life.
Chapter 1

Introduction

1.1 Motivation

The term "mechatronics" is considered to be first used by engineer Tetsuro Mori in 1969 who combined the word "mechanical" and "electronic" to describe the electronic control systems that were developed for mechanical factory equipment at Yaskawa Electric Corp. Since then, the meaning of the term has broadened to include more technical areas and design methods. Some literature defines mechatronics as the synergistic application of mechanics, electronics, control engineering, and computer science in the development of electromechanical products and systems through integrated design [1]. Some other literature regards the identity of mechatronics as the "interdisciplinary design methodology which solves primarily mechanically oriented product functions through the synergistic spatial and functional integration of mechanical, electronic, and information processing sub-systems" [2]. Due to the wide application areas in both industry and consumer market, mechatronic systems have played a significant role in these areas over the past few decades. Most mechatronic systems are multi-domain (or multi-physics) mechatronic systems [3], which consist of different domains such as mechanical, electrical, hydraulic, pneumatic, thermal, control, and so on, examples of which include automated packaging lines [4], industrial robots [5], Computer Numerical Control (CNC) machines, smart home appliances, and aircraft.

A typical mechatronic system consists of a mechanical skeleton, actuators, sen-
ors, controllers, signal conditioning/modification devices, computer/digital hardware and software, interface devices, and power sources [6]. With the rapid development of modern technological applications, mechatronic systems have become increasingly complex and demanding. Due to its complex structure and possible dynamic coupling between different domains, the design of mechatronic systems is a rather complicated and challenging task. Furthermore, the flexibility of a manufacturing system is quite important and advantageous in modern industries, which function in a competitive environment where market deviation and the need for customized product are growing. The machinery of a manufacturing system should be reliable, flexible, intelligent, less complex, and cost effective and should be developed using mechatronic methodologies. Specifically, to achieve these goals, the design methodologies for these systems should be revisited and improved in the context of Mechatronics. In particular, continuous or on-demand design improvements may have to be incorporated rapidly and effectively in order to address new design requirements or resolve existing weaknesses of the original design.

First, an integrated and concurrent approach should be considered in the design process, particularly in the conceptual and detailed design phases. This particularly means that dynamic interactions between domains should be considered concurrently throughout the design process. Recently, in the context of multi-domain design, attention has been given to such subjects as multi-domain modeling, multi-criteria decision making, and evolutionary computing. Appreciating that an effective method of system representation (i.e., modeling) is important for a systematic method of design, dynamic modeling approaches such as Bond Graphs (BGs) [7] and Linear Graphs (LGs) [8, 9] have been investigated for the design of multi-domain system. These modeling methods can describe relationships and interactions between different components and domains of a dynamic system. Along with modeling methods for multi-domain dynamic systems, several strategies for design optimization have been proposed. In the process of design optimization, a rather challenging task is to satisfy multiple design objectives. Fuzzy integrals have been investigated as a systematic aggregation of different and sometimes contradictory design objectives. Evolutionary optimization strategies can then be adopted to evolve the design. Particularly, genetic programming (GP) [10], in recent years, has gained much interest as it can be extended to evolutionary optimization that
satisfies a variety of design specifications. Through GP, a design model of a multi-domain system can be evolved to search for an optimal solution in the design space, in an automated manner. However, for complex mechatronic system design, the design space can be extensive. Thus, it is crucial to detect components with weak designs or parameters among the numerous subsystems and components so that evolutionary design algorithm can be applied efficiently.

Second, continuous and rapid design improvement should be considered based on the running condition of the engineered system to achieve enhanced requirements in reliability, flexibility, cost, intelligence, and so on. By evaluating the performance of the running system and detecting, and diagnosing system malfunctions and failures, design weaknesses of the system can be identified for further improvement. With the development of monitoring techniques, machine health monitoring (MHM) has been conventionally utilized in mechatronic systems as an efficient tool for the diagnosis of system malfunctions and the prognosis of the impending failures. Methodologies can be developed to decrease the occurrence of faults and to enhance the reliability of the system. Researchers have realized as well the potential in facilitating design evolution using MHM system.

Recently, with the emerging developments of the Internet of things (IoT), especially with the advances in industrial IoT, the data collection from operating machinery is growing exponentially as more and more subsystems and components are communicating with each other or are being monitored. The big data collection offers new opportunities not only for smart and flexible manufacturing but also for the potential of design improvement of existing machinery. It is conceivable that, through the information acquired by an MHM system, some design weaknesses of the monitored mechatronic system can be detected and identified. Then, with the assistance of multi-domain dynamic system modelling and evolutionary optimization, improvements in the design of the system can be prescribed. However, to evaluate a current design and thereby detect the possible design weakness through the massive amounts of acquired machine condition data, a systematic approach has to be established. Moreover, the traditional machine health diagnosis and prognosis algorithms can only deal with limited categories of machinery. They are not capable of capturing some valuable information in massive amounts of machine condition data. Thus, more general and intelligent machine health diagnosis
and prognosis algorithms should be developed with enhanced capability of acquiring useful information from big data so that more subsystems and components can be evaluated.

1.2 Research Objectives

To achieve continuous and rapid design improvement in mechatronic systems, the performance of the existing systems should be evaluated and thereby the existing design weaknesses should be identified. Then, integrated design optimization methods can be carried out more efficiently based on the detected design weaknesses. MHM data of running system can form valuable feedback for the design improvement process. The main challenge is how to establish systematic methods to determine the design weaknesses by using massive MHM data. The first main goal of the present research is to develop a systematic framework for the continuous design evolution of a mechatronic system through MHM. In this dissertation, an MHM scheme based on IoT and cloud computing (CC) is developed. The second objective of this dissertation is to develop a formal and systematic approach for the detection of existing weaknesses of a monitored mechatronic systems. The method proposed in the present work develops an index to identify a possible design weakness by evaluating the performance of a system and thereby detecting system failures and quantifying the health status of the system. Also, with the collected big data, improved algorithms for fault diagnosis and remaining useful life (RUL) prediction are developed by using deep neural networks (DNN), which is a cutting edge learning method that can learn wide knowledge from large amounts of data.

1.3 Related Work

1.3.1 Evaluation of Mechatronic Systems

Using a quantitative evaluation of the performance of a mechatronic system, the competing design solutions can be objectively compared. Different performance specifications and evaluation criteria have been proposed and applied to the design of mechatronic systems. The design methodologies of mechatronic systems have
evolved from the sequential approach to the concurrent and integrated approach where the interaction and coupling between different domains are considered. Accordingly, various new methods have been proposed to evaluate a mechatronic system in the process of design evolution.

Moulianitis et al. [11] proposed an approach for the evaluation process of a mechatronic system in the stage of conceptual design. The candidate solutions were evaluated through an evaluation index, which was based on fuzzy set theory. The evaluation index included three elements: complexity, intelligence, and flexibility where each element contained several criteria. Weight factors were used to indicate the significance of each criterion. Different t-norms and averaging operators were compared and discussed to calculate the final evaluation score. The method was applied to the conceptual design of robot grippers for handing fabrics, which showed its effectiveness. However, the interactions between criteria were not discussed. Furthermore, only discrete design space was considered.

De Silva [12] argued that optimal design of the subsystems in a sequential design approach does not guarantee the optimum performance of the overall mechatronic system. He proposed to specify separate performance indices for the subsystems of the mechatronic system. Then a performance measure called mechatronic design quotient (MDQ) was developed, which is employed to optimize the design by making the indices with subsystem interactions to reach the optimal values of the indices in a sequential design. Then the goal of the mechatronic design is to find a design solution that maximizes the MDQ.

Based on the concept of MDQ, Behbahani and de Silva [13] developed an integrated and multi-criteria methodology for the conceptual design of mechatronic systems. The controller design issues and parameters were considered concurrently with other issues and parameters. The interactions between criteria were modeled by fuzzy logic criteria. A nonlinear fuzzy integral called Choquet integral was used for the aggregation of criteria. The methodology was implemented in the design of the manipulator of an industrial machine, which demonstrated the effectiveness of the MDQ approach in the conceptual design stage. In the detailed design stage, MDQ may be employed in a similar manner.

Janschek [2] developed an approach for mechatronic system evaluation using performances metrics based on the deviation of the measured value to a reference
value. System performance based on both deterministic and stochastic models was discussed. The author used $p$-norms of the performance over an observation time to evaluate the performance under a deterministic model and the 3 standard deviation method to evaluate the stochastic performance under a stochastic model. Then, the performance metrics were combined by nonlinear summation, for example, quadratic summation or max-summation. The correlation between metrics was represented by the covariance values.

Hammadi et al. [14] proposed a multicriteria performance indicator, called mechatronic design indicator (MDI) for evaluating the performance of a mechatronic system in the preliminary design stage. The individual performance metrics were defined and evaluated using model-based simulation. Then a neural network of Gaussian radial basis functions was adopted to calculate the MDI by aggregating the individual performance metrics. The MDI approach was applied to the design of a mechatronic system with subsystems of an electric motor, a PID controller, and a crack-slider mechanism. The result showed that MDI could provide accurate information for decision making. This approach is essentially similar to the previous MDQ approach even though the authors failed to mention that.

Villarreal-Cervantes et al. [15] proposed a concurrent design method for a planar five revolute two degree-of-freedom parallel robot. The design problem was modeled as a nonlinear dynamic optimization problem (NLDOP) where the kinematic and dynamic behaviors were considered simultaneously. They used two optimization techniques, differential evolution, and sequential quadratic programming to solve the NLDOP. The optimum structure parameters and the optimum PID control gains were obtained by optimizing the cost function considering the positioning accuracy and a manipulability measure.

1.3.2 Evolutionary Design Optimization

In the past decade, evolutionary algorithms have been increasingly used in design optimization due to their explorative capabilities and flexible representations. Considerable literature incorporates evolutionary algorithms, in particular, genetic algorithm (GA) and genetic programming (GP), to explore the parameters of the design. GA is inspired by the evolutionary process of species and operates on
the Darwinian principle of survival of the fittest. The solution is represented by a chromosome structure that consists of a number of genes. A population of trial solutions is randomly generated and evaluated to see how satisfactory each solution is. Then reproduction operators are applied to generate new design solutions from a pair of parent solutions selected from the existing population. The reproduction process continues until the termination criterion has been satisfied.

Grimbleby [16] proposed a new approach to circuit synthesis based on a hybrid GA algorithm. The circuit topology was represented by a chromosome that consists of a number of genes. Each gene contained the component type and their terminal nodes. The length of the chromosome was flexible by introducing the empty component type. When the circuit topology was generated, its fitness was evaluated after numerically optimizing its component values using a quasi-Newton algorithm based on the Davidon-Fletcher-Powell (DFP) method. Circuits generated by the proposed method were efficient and fully met the design specifications.

Affi et al. [17] applied GA for multi-objective optimization of a four-bar mechatronic system where the geometry and the dynamic behavior were considered simultaneously. The authors compared the concurrent approach with the sequential approach where they first optimized the geometry of the mechanism for a given path and optimized the mass distribution to minimize several objective functions including the maximum current used by the motor, its maximum variation, and its fluctuation. The results showed that the GA-based multi-objective optimization approach could realize better design solutions.

GP is an extension of genetic algorithms where the encoding of the solution is a tree of computer programs of varying sizes and shapes instead of a chromosome structure in a conventional GA. Then the genetic algorithm operates on a population of computer programs to find the optimal solution. GP has been successfully applied in design optimization due to its representation flexibility and the exploration capability.

Koza et al. [18] applied GP to achieve an automatic synthesis of analog electrical circuits. They established a mapping between the tree structure solution in GP with the electrical circuits. A set of circuit-constructing functions were defined including component-creating functions, topology-modifying functions, and development-controlling functions. The genetic operators were then applied on a
population of embryo solutions with the circuit-constructing functions. Both topol-
ogy and sizing of the electrical circuits were optimized to automatically create an
analog electrical circuit from a high-level statement of the circuits desired behav-
ior. Their approach was applied in the automatic synthesis of eight prototypical
analog circuits.

Inspired by Kozas work, Seo et al. [19] proposed a unified and automated
procedure capable of designing mechatronic systems to meet given performance
specifications, subject to various constraints. They used BG to model a multi-
domain mechatronic system and used GP to explore the design space in achieving
an optimal design. BG is an efficient tool for modeling multi-domain systems in
view of its domain independent, free and open-ended composition, and efficiency
in classification and analysis of models. Thus, various types of acceptable and
feasible candidate designs can be determined. The authors established a mapping
between BG model and a GP tree structure representation. They defined various GP
construction functions and terminals for constructing BG models. Their method
achieved automatically synthesizing designs for a multi-domain dynamic system.
However, their approach did not consider controller parameters.

Wang et al. [20] extended the basic BG/GP approach to achieve concep-
tual mechatronic system design where controller schemes were also incorporated.
Their work focused on the improvement of the BG-GP approach through knowl-
edge interaction with GP by extracting and incorporating the modular design knowl-
edge. Different controller schemes, for example, proportional, proportional plus
derivative, proportional plus integral, proportional-integral-derivative, and lead and
lag compensators, were represented by combinations of bond graph elements. Com-
plex module functions were defined and used as building blocks to reduce the GP
search space. The incorporated design knowledge enhanced the search efficiency
and realizability of the generated designs.

Behbahani and de Silva [21] proposed a two-loop evolutionary approach with
a hybrid of GA and GP for the design optimization of a mechatronic system. A
multi-domain mechatronic system was represented by a BG. GP was used in the
outer loop for topology optimization and GA was adopted to find an elite solution
for each topology generated by the GP. To avoid the creation of repeated topologies,
a memory feature was added to the GP process. Their approach provided a con-
current, integrated, and autonomous tool for topology synthesis of a mechatronic system. It was applied to the design of analog filters and the controller design of an industrial fish processing machine. Later, the authors improved this approach by introducing a niching scheme [22]. Several elite solutions were generated for different preferences. The designer would be able to make the final decision by evaluating these elites by further domain knowledge or other criteria what could not be easily incorporated in the fitness measure. This approach was more consistent with a real design practice. It offered increased flexibility to designers and more chances to achieve a feasible design.

1.3.3 Machine Health Monitoring

The growing demand for high quality, low cost and highly customized products requires increasing reliability of machinery in production systems. Machine health monitoring (MHM or, machine condition monitoring MCM) has been traditionally used to detect, diagnose and correct system faults [23]. In the past several decades, condition monitoring and fault diagnosis have been widely investigated and applied in a variety of areas such as industrial automation, process control, and anomaly detection and maintenance in aerospace, automobile, wind turbine, railway and manufacturing sectors [24–27]. Both model-based [28] and data-driven [29] approaches have been developed in the past several decades. The characteristics and their application scenarios of different fault detection and diagnosis approaches in complex systems have been reviewed and analyzed in [30] from the perspective of data processing. The high complexity and cost of model-based approaches limit their implementation in real applications [31]. However, with advances in sensor techniques and signal processing, data-driven approaches based on various signals such as vibration, acoustic emission, temperature, pressure, and current, have received increased attention in recent years [29]. Traditional data-driven fault diagnosis algorithms primarily involve the following three steps: data acquisition, feature extraction and condition classification [32].

Features of the signals in the time domain, frequency domain, and time-frequency domain have been widely investigated and applied for fault diagnosis. Feature selection techniques such as principal component analysis (PCA), dynamic principal
component analysis (DPCA), independent component analysis (ICA), and partial least squares (PLS) are usually applied to decrease the dimension of the feature space while retaining most of the information from the data [33]. Classification algorithms, typically artificial intelligence (AI) approaches e.g. support vector machine (SVM), artificial neural networks (ANN) and k-nearest neighbors (kNN) [34], are used to classify the conditions. Classifier ensembles have been investigated as well to increase the classification accuracy and robustness [35].

Another significant application area of machine condition monitoring is prognostics and condition-based maintenance. Through the information from condition monitoring of a dynamic system, the remaining useful life (RUL) of the system can be predicted and an appropriate maintenance plan can be established to achieve good performance of the system at a minimum maintenance cost [36]. Data-driven approaches with the integration of AI techniques, e.g. ANN and SVM, have been widely studied in recent years for RUL prediction.

Wu et al. [37] developed an ANN-based decision support system for predictive maintenance of rotating equipment. A three layer ANN was used for the prediction of the RUL. The root mean square (RMS) value was calculated from the vibration signal of the degradation process. The current and previous RMS values together with the current time value were constructed as inputs to the ANN. Ten point moving average was used to smooth the RMS value.

Tian [38] proposed an improved ANN-based method for RUL prediction. Another hidden layer was added to a typical three layer ANN to better capture the nonlinearity. The author introduced more measurements as inputs to the ANN. The condition monitoring measurement series for a failure history was fitted by a function generalized from the Weibull failure rate function. The fitted measurement values were used as inputs to the ANN to reduce the effects of the noise factors. This approach achieved better RUL prediction than that from Wu's method.

With the advances in wireless sensor networks, cloud computing, Internet of Things and smart factory, the amount of data acquired in condition monitoring has grown exponentially. Such big data present new challenges in data storage, transmission, and processing as well as new opportunities in discovering more useful underlying information in a reliable and effective manner [39]. Machine condition monitoring has shown much potential in identifying weaknesses in engineering...
Figure 1.1: System framework of design evolution with machine health monitoring.

systems that can be related to poor design. de Silva [1] and then Gamage and de Silva [40] proposed a framework that can specifically integrate an MHM system and an expert system to carry out design evolution of a multi-domain dynamic system, as shown in Fig. 1.1. The utilization of information from condition monitoring of an engineering system was proved to be useful in design improvement of the system by detecting malfunctions of the system.

Besides fault detection, such valuable information as system performance and remaining useful life can be evaluated from the condition monitoring data to detect possible system weaknesses. Xia and de Silva [41] presented a methodology for design weakness detection of an engineering system through MHM. Using the sensed condition data, system performance, fault diagnosis, and remaining useful life estimation were performed to identify the weaknesses of the current design.

Still, there are many important and unresolved issues of integrating machine condition monitoring and engineering system design. For instance: (1) The traditional monitoring systems face many challenges related to communication and storage of huge amounts of sensed data; (2) Computing capability of a local computer can limit the analysis and interpretation of the collected big data; (3) How
to analyze the data and evaluate the current and future status of the monitored system; (4) How to translate the results into knowledge. (5) How to manage and share knowledge for assisting the design process, especially collaborative design. (6) How to keep it cost effective due to the large amounts of sensors and data acquisition devices that are needed.

1.3.4 Deep Neural Networks

Most of the traditional fault diagnosis approaches rely on manual feature extraction, which requires significant prior knowledge of signal processing and diagnostic expertise. Also, the existing algorithms are meant for specific issues and therefore are case sensitive and not meant for general application. In recent years, deep neural networks (DNN) have been investigated and have shown the promising capability of capturing representative features from raw data through multiple nonlinear transformations across their deep structures. DNN have been implemented in many applications such as computer vision, natural language processing, speech recognition, and bioinformatics with outstanding performance compared to the approaches that use traditional manually designed features.

Krizhevsky et al. [42] trained a deep neural network to classify 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into 1000 classes. Their DNN consisted of convolutional layers, max pooling layers, fully connected layers and a softmax layer. The dropout method was used to reduce overfitting in the fully-connected layers. They achieved considerably better classification results compared with the state-of-the-art.

Ruhi et al. applied Deep Belief Nets (DBN) to a natural language callrouting task. The performance of the DBN-based approach was compared with SVM, boosting and maximum entropy. With pre-training of the DBN by unsupervised learning using the additional unlabeled data, the proposed approach achieved better performance than other methods.

Graves et al. [43] proposed a deep recurrent neural network approach with a combination of multiple levels of representation and the flexible use of long-range context, which achieved a lower error rate on the TIMIT phoneme recognition benchmark. Di Lena et al. [44] utilized a DNN architecture for protein contact map
prediction, which increased the classification accuracy by more than 10% over the existing approaches.

Due to the superior capability of DNN in feature learning and classification, they have recently attracted the attention of researchers in machine fault diagnosis. Jia et al. [45] presented a five-layer DNN model for diagnosing the faults of rotating machinery. The model was first pre-trained by an unsupervised autoencoder and fine-tuned with the backpropagation algorithm for classification. It achieved excellent diagnosis performance. Sun et al. [46] proposed a sparse auto-encoder-based DNN approach for induction motor fault classification. Representative features were learned automatically through a sparse autoencoder and used for fault classification.

1.4 Contributions and Organization of the Dissertation

The main contributions of the dissertation are listed below.

1. A novel closed-loop design evolution framework for mechatronic systems is presented. Compared with other design evolution methodologies, it can achieve continuous design improvement for mechatronic systems through conceptual design, detailed design, implementation, condition monitoring and design weakness detection. New design requirements or existing design weaknesses can be addressed by the proposed approach. IoT and CC are introduced to address the limitation of the traditional MHM approach in sensing, data transmission, data storage and data processing. A systematic evaluation approach is developed to detect possible design weaknesses, which will guide the redesign by reducing the search space.

2. An intelligent fault diagnosis approach based on stacked denoising autoencoder (SDA) and DNN is developed to enhance the capability of an MHM system for use with more general and wider categories of mechatronic systems. The typical drawbacks of the traditional fault diagnosis methods are: (1) features are extracted with prior knowledge, (2) a large amount of labeled data is needed, and (3) diagnosis model has to be rebuilt for diagnosing new conditions. The developed approach overcomes these drawbacks. It can learn representative features automatically from a massive quantity of unlabeled condition data. Only a few items
of labeled data are needed to fine-tune the DNN to perform fault diagnosis. In addition, the proposed approach can utilize the trained model to diagnose new conditions by further fine-tuning the DNN with a few items of labeled data of the new conditions. The effectiveness of the developed approach is verified by using a standard dataset of bearing faults. The robustness of the method to noise is evaluated by corrupting the original signal with different levels of noise.

3. A convolutional neural networks (CNN)-based approach with multiple sensor fusion is developed for fault diagnosis of rotating machinery. Signals from multiple sensors are fused at the data level to form the input to the model. Representative features can be learned directly from raw signals by the CNN model where no hand-crafted features are needed. Dropout is integrated to overcome the problem of overfitting when the size of training data is small. The effectiveness of the approach is verified by experimental studies in gearbox and bearing fault diagnosis. The comparison between the proposed method and the traditional approaches show the superior performance of the proposed method. More importantly, this end to end feature learning capability of the proposed approach would enable its wide application in fault diagnosis of machinery with limited prior knowledge and limited representative handcrafted features.

4. A hierarchical DNN-based RUL prediction method is proposed with enhanced prediction accuracy. This method models the degradation process under several health stages. Several ANN are used to model the degradation in each stage. A DNN-based classifier is trained to achieve the health stage classification using raw monitoring data. The degradation in each stage is then modeled by an ANN using the calculated features. A smoothing operator is applied on the health stage classification result and the RUL predictions of each stage. The experiment result shows that the RUL prediction accuracy using the proposed method is superior to that with only one single model.

The organization of the dissertation is as follows:

Chapter 2 presents the proposed closed-loop design evolution approach for continuous and on-demand design improvement of a mechatronic system. An MHM scheme based on IoT and CC is proposed to employ condition monitoring in the design improvement process by evaluating system performance, detecting the system failure and estimating system future status. A design weakness index
is proposed to detect possible design weakness and reduce the search space for effective use of the evolutionary design optimization.

Chapter 3 presents an improved DNN-based fault diagnosis approach that has the potential for fault classification of more general mechatronic components and systems. A SDA-based DNN is incorporated. Representative features are learned by applying the denoising autoencoder to the unlabeled data in an unsupervised manner. A DNN is then constructed and fine-tuned with a small amount of labeled data. New conditions can be classified by simply fine-tuning the trained DNN model using a small amount of labeled data under the new conditions.

Chapter 4 presents the CNN-based approach for fault diagnosis with data fused from multiple sensors. Data level sensor fusion is achieved with the CNN structure where temporal and spatial features can be learned. No hand-crafted features are used because the CNN model can extract representative features directly from raw signals. Fault diagnosis of two widely used machine components, gearboxes and ball bearings, are tested using the CNN-based approach. The results show the superior diagnosis performance of the proposed method compared to the traditional approaches. The developed approach can be extended to fault classification of other machinery with various types of sensors, due to its end-to-end feature learning structure.

Chapter 5 presents the hierarchical DNN-based RUL prediction method. The overall degradation process is segmented into different health stages. A DNN-based classification model is built to achieve the health stage classification. For each stage, an ANN-based RUL predictor is trained using data samples in each stage. A smoothing operator functions on the output of the health stage classifier and the output of each RUL predictor are used to obtain the final RUL prediction.

Chapter 6 concludes the dissertation by summarizing its main contributions and discussing the direction of possible future research.
Chapter 2

Closed-loop Design Evolution of Mechatronic Systems

2.1 Introduction

Flexibility of a manufacturing system is quite important and advantageous in modern industries, which function in a competitive environment where market deviations, production of small batches, and the need for customized products are common. The key machinery of a manufacturing system should be reliable, flexible, intelligent, less complex, and cost effective. To achieve these goals, the design methodologies for engineering systems should be revisited, improved, and innovated. In particular, continuous or on-demand design improvements have to be incorporated rapidly and effectively in order to address new design requirements or resolve existing weaknesses of the original design. The design of an engineering system, which is typically a multi-domain system, can become complicated due to its complex structure and possible dynamic coupling between physical domains. Hence, an integrated and concurrent approach should be considered in the design process, in particular both conceptual and detailed design phases. In the context of multi-domain design, attention has been given recently to such subjects as multi-criteria decision making, multi-domain modeling, evolutionary computing, and genetic programming. More recently, machine health monitoring (MHM) has been considered for integration into a scheme of design evolution even though
many challenges exist for this to become a reality. The challenges include lack of systematic approaches and the existence of technical barriers in the acquisition, transmission, storage, and mining of very large amounts of condition data.

de Silva [6] proposed a framework of evolutionary design improvement incorporating MHM, as shown in Fig. 2.1. Evolutionary design improvement can be facilitated through online monitoring, in conjunction with a model of the existing system whose design needs to be improved and evolutionary computing techniques, e.g. GP. The MHM system is expected to identify the regions (sites) of possible design weaknesses in the system. This provides ”modifiable sites” for the existing system or model for further optimization.

Today, the IoT and CC are being developed quickly, which offer new opportunities for evolutionary design in such tasks as data acquisition, storage and processing. A framework for the closed-loop design evolution of engineering system is developed to achieve continuous design improvement for engineering system using an MHM system assisted by IoT and CC. New design requirements or the detection of design weaknesses of an existing engineering system can be addressed through the proposed framework. A design knowledge base that is constructed by integrating design expertise from domain experts, on-line process information from condition monitoring and other design information from various sources is proposed to realize and supervise the design process to achieve increased efficiency, design speed, and effectiveness.

2.2 System Framework

In this dissertation, a framework is proposed for design evolution of an engineering system through condition monitoring using IoT and CC, as shown in Fig. 2.2. With the help of IoT, condition data are collected at different sites of the engineering system. The status of the operational system and of the subsystem modules is analyzed through data mining, which involves evaluation of the performance, diagnosis of faults, and estimation of the RUL of the system. Then design weaknesses are detected for the monitored system and provided for consideration in a future stage of design improvement. New design methodologies, available technologies, developed functional modules and their parameters can be collected from domain
experts, handbooks, Internet and other information sources. This information, together with the information from condition monitoring, can be utilized in forming a design knowledge base, which is continuously updated. It can assist the designers in generating an innovative and efficient design solution both in the conceptual design stage and the detailed design stage. In this manner, a closed-loop design improvement process is achieved to accommodate new design requirements or to correct the system design weaknesses.
2.3 Condition Monitoring through IoT and CC

A key task of condition monitoring is the acquisition of condition data of an engineering system such as throughput, capacity, speed, torque, power consumption, size, weight, vibration, current, voltage, and other response variables and parameters that can be valuable for evaluating the status of the system. The proposed scheme of condition monitoring through IoT and CC is shown in Fig. 2.3. With the help of IoT, the condition data from various modules of an engineering system, not just at one site but at different and geographically separated sites can be collected. When a system consists of multiple sites (or sensor nodes), fusion of the condition data from different sites can provide a more reliable estimation of the system performance. This process includes three main steps: data acquisition and
preprocessing, data transmission, and data mining.

2.3.1 Dada Transmission

A network layer is utilized to transmit the sensed data. Short-range wireless networks such as WiFi, Bluetooth, Zigbee and Sensor Area Network (SAN) are common technologies that support the connection of sensors, devices and users, for
data transmission. Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) are common standards for the transport networks. Data on the system condition (condition data), after preprocessing, are transmitted through the network layer to the cloud database.

2.3.2 Data Mining

From the condition data, three main types of analysis are conducted: 1. System performance evaluation, 2. Machine fault detection, and 3. Prediction of the RUL.

Acceptable system performance and low cost are two key objectives in the mechatronic design of a system. Factors related to performance of a mechatronic system include: capacity, efficiency, reliability, stability, accuracy, and so on. Requirements of system performance can be extended to integrate other requirements such as controllability and low cost. Here, performance variables are defined by the designers according to the design specifications.

Machine fault detection may be treated as a procedure of mapping the information obtained in the measurement space and/or features in the feature space to machine faults in the fault space [23]. This mapping process may be considered as a procedure of pattern recognition. It is achieved by automatic classification of the signals based on the features extracted from them. AI techniques have been increasingly applied to machine fault diagnosis and have shown to improve the performance over conventional approaches. Some common AI techniques include expert systems, fuzzy logic, ANN, GA and SVM. The efficiency of AI techniques has been found to be satisfactory in many case studies.

Prediction of the RUL means the prediction of the time left before a failure would occur given the current and past profile of the machine condition. To perform prognosis, one must have knowledge (or information) of the failure mechanism as well as knowledge (or information) of the fault propagation process. Similar to fault diagnosis, there are three main approaches of prognosis: 1. Statistical approaches, 2. AI approaches, and 3. Model-based approaches. In particular, statistical approaches such as Hidden Markov Model and Particle Filter are widely used in the RUL prediction.
2.3.3 Design Knowledge Base through CC

In the proposed framework, the design knowledge base is constructed on a cloud computing platform using the design expertise of domain experts (design methodologies, regularized design, and so on.), technical solutions, available function modules with their specifications, design handbooks, data tables, catalogues, the Internet, information from condition monitoring system (system performance, detected fault, maintenance history, remaining useful life), and so on. The knowledge base is updated in an evolutionary manner through mining of the continuous condition monitoring data, experts input, new technology approaches, newly developed modules and so on, supported by IoT. Collaborative design can be promoted by sharing the knowledge base between the designers from different sites through ubiquitous access to the knowledge base. Through an inference engine, the design knowledge base is utilized to supervise the searching of the design space both in the conceptual and detailed phases by reducing the search space and offering design guidance.

2.4 Design Weakness Detection

As shown in Fig. 2.2, closed-loop design improvement of an engineering system is achieved through design weakness detection using condition monitoring and subsequent conceptual design, detailed design, and implementation of the design improvements. The process of design improvement is carried out under the guidance of a design knowledge base, which is continuously updated.

In the stage of conceptual design, innovative ideas and multi-criteria evaluation are crucial. With a cloud-based design knowledge base and a collaborative design scheme, creative design ideas can be investigated by the design team more easily and efficiently. With the assistance of the design knowledge base, the design space for conceptual design can be structured to form possible conceptual alternatives.

In the detailed design stage, after establishing a dynamic model of the multi-domain system, algorithms that can explore the design space should be applied to achieve the detailed design leading to a desired optimal behavior in the system. Specifically, the model will be modified in some manner so that its behavior approaches the desired behavior, in an optimal manner, as represented by a cost
function. For detailed design optimization, GP has been employed to realize an optimal design in an evolutionary manner, so as to satisfy a set of specified design objectives.

Production of the engineering system may be carried out using the outcome of detailed design. The resulting new generation of engineering systems will be implemented in manufacturing systems to achieve the required production performance. Condition monitoring is conducted once the designed system is running. Fig. 2.4 shows the procedure of detecting a design weakness in the current design.

System performance evaluation, fault diagnosis, and prognosis are carried out using the monitored condition data. Design weakness candidates can be identified using the Design Weakness Candidate Index (DWCI), which is defined as:

\[
\text{DWCI} = (\hat{W})^T \begin{bmatrix} S(\hat{P}) \\ E(\hat{T}) \end{bmatrix} \prod_{i=1}^{f} g_i(\hat{F})
\]

(2.1)

where \(\hat{W}\) is an \(r+k\) element column vector, and \(\sum_{i=1}^{r+k} w_i = 1\). \(S(\hat{P})\) is an \(r\) element column vector. Each element \(s_i(p_i)\) is a function that shows the degree of satisfaction of the \(i\)th performance aspect. \(\hat{P} = [p_1, p_2, ..., p_r]^T\) is a vector consisting of all performance aspects, and \(E(\hat{T})\) is a \(k\) element column vector. Each element \(e_i(t_i)\) is a function that indicates whether the estimated RUL of the \(i\)th component is close to its designed lifetime, and \(\hat{T} = [t_1, t_2, ..., t_r]^T\) is a vector consisting of all estimated RUL values of the components. Furthermore, \(g_i(\hat{F})\) is a function indicating whether a fault has occurred. It is equal to 0 if a fault has occurred and 1 otherwise. Both \(S(\hat{P})\) and \(E(\hat{T})\) should take into consideration the situations of over performance and under performance. This means if the performance or RUL of the system considerably exceeds the designed specification, it is a wasteful situation, and those functions should be able to add a punishment into the DWCI.

According to the definition of DWCI, once a failure is detected, its value will drop to zero. If a fault does not occur and the system is running smoothly, DWCI should also be as smooth as \(S(\hat{P})\) and \(E(\hat{T})\) would change slightly. If a significant decrease in DWCI is observed in a comparatively short time, there must be some problem during operation. Therefore, once DWCI drops to zero or decreases significantly, the design weakness candidates will be isolated by checking the com-
Figure 2.4: Procedure of design weakness detection.
ponents of the index. Then the design weakness candidates will be evaluated first to see if the behavior is related to non-design issues such as inappropriate installation, non-standard operation or poor maintenance. If so, these issues will be corrected and condition monitoring of the system will be continued. Otherwise, the design weakness will be imported to the design process for design improvement. After the design improvement is made, the redesigned engineering system will be fabricated and put into operation. Condition monitoring of the system is continued to detect further design weaknesses. The process of design evolution is in a loop and it can offer design improvements continuously.

2.5 Mechatronic Design Quotient

The design of an engineering system is carried out broadly at two levels: the conceptual design where the type and function of the subsystems are identified and some high-level decisions about the operation of the system are made [47], and the detailed design where the topology and parameters of the subsystem are specified or tuned [48].

In the conceptual phase, high-level decisions of the system structure and feasible conceptual choices are made according to the design expectation. Conceptual design is rather important in a design process. The design space can be extensive as there can be a variety of possible configurations, and it is not feasible to achieve the best design in one step. In conceptual design, the designer divides the complex design space into several subspaces and evaluates all these subspaces properly and narrows the design down to one or two subspaces. This offers a less complex searching space for the subsequent phase of detailed design. A rather challenging task in design optimization is to concurrently satisfy multiple design objectives. Behbahani and de Silva [13] presented a systematic approach for concurrent and integrated design of a mechatronic system by using the concepts of MDQ. Their approach used MDQ in the evaluation model to facilitate decision making to achieve an optimal conceptual design of a 2-D manipulator. MDQ is an effective tool in multi-criteria design evaluation of mechatronic systems. It can be utilized to evaluate possible conceptual alternatives in the conceptual design phase. In a design problem of \( n \) design criteria and \( r \) constraints, MDQ can be written in
In the following form:

\[
MDQ(d) = M \left[ x_1^d, x_2^d, \ldots, x_n^d \right] \prod_{i=1}^{r} c_i(d)
\] (2.2)

where \( d \) represents a design alternative, \( M \) is an aggregation operator, \( x_i^d \) is the partial score that shows the degree of satisfaction of the \( i \) th criterion, and \( c_i(d) \) is a function indicating whether a constraint has been met. In particular, \( c_i(d) \) is equal to one if the \( i \) th constraint is met. Otherwise, it is equal to zero.

The evolution criteria in the MDQ formulation may include "Meeting task requirement", "Complexity", "Reliability", "Matching", "Flexibility", "Control friendliness", "Efficiency", and "Cost" [49]. In practice, the designer may decide to utilize other criteria if they are important for the design problem, or drop some of the above criteria if they are not important in the particular design problem. Some of the criteria may take an analytical form while some others may be qualitative and fuzzy and may involve human perception [50]. A key step is the aggregation of criteria. Interactions can exist between criteria. The traditional aggregation method of weighted average cannot deal with the interaction between criteria and it is only suitable when the criteria are independent. Fuzzy measures are used to model the interactions between criteria in many situations [51]. In the discrete case, a fuzzy measure on \( N \) is a set function \( v : 2^N \to [0, 1] \) satisfying

\[
v(\emptyset) = 0 \quad \text{(2.3)}
\]
\[
v(N) = 1 \quad \text{(2.4)}
\]
\[
S \subseteq T \Rightarrow v(S) \leq v(T) \quad \text{(2.5)}
\]

For any \( S \subseteq N \), \( v(S) \) can be interpreted as the weight of the degree of importance of the combination \( S \) of criteria [52]. Several fuzzy integrals have been developed in aggregating the criteria in multi-criteria decision making [53]. Choquet integral has been developed and utilized in many applications of multi-criteria evaluation [54, 55], and can be used for the aggregation of criteria in MDQ.

Specifically, the Choquet integral can be utilized to aggregate the criteria to

26
compute the global score of each alternative using the following equation:

$$C_v(x) := \sum_{i=1}^{n} x(i) \left[ v(A(i)) - v(A(i+1)) \right]$$  \hspace{1cm} (2.6)

where $(\cdot)$ indicates a permutation of $N$ such that $x(1) \leq \cdots \leq x(n)$, $A(i) = \{(i), \ldots, (n)\}$ and $A(n+1) = \emptyset$ [51]. The Choquet integral may be written as well in the form:

$$C_v(x) = \sum_{T \subseteq N} a(T) \land_{i \in T} x_i$$  \hspace{1cm} (2.7)

where $\land$ denotes the minimum operator and the set function $a : 2^N \rightarrow R$ is the Mobius transform of fuzzy measure $v$ as given by

$$a(S) = \sum_{T \subseteq S} (-1)^{s-t} v(T)$$  \hspace{1cm} (2.8)

where $s = |S|$ and $t = |T|$.

A key problem in using the Choquet integral is that $2^n$ coefficients in $[0, 1]$ need to be specified to define the fuzzy measure on every subset of $n$ criteria. This is challenging for the designer and is not practical in real applications. Grabisch [56] suggested to consider the 2nd order Choquet integral, which seems to be more practical in real applications. It allows modeling of interaction among criteria while remaining simple and operational. The 2nd order Choquet integral is given by

$$C_v(x) = \sum_{i \in N} a(i)x_i + \sum_{i,j \in N} a(ij)(x_i \land x_j)$$  \hspace{1cm} (2.9)

where $a(i) = v(i)$ and $a(ij) = v(ij) - v(i) - v(j)$ defined by Equation 2.8.

After the specification of $v(i)$ and $v(ij)$, all $a(i)$ and $a(ij)$ can be calculated. Therefore, instead of $2^n$ coefficients, only $n + C_n^2 = n(n+1)/2$ coefficients are needed.

In the phase of detailed design, first the best topology is determined, for example, system components and their interconnections. Then component details are specified to achieve the best satisfaction of the design requirement. Methods of evolutionary computing, GP in particular, have received much attention in recent years for autonomous topology generation. Evolutionary algorithms are proved to
be effective in assisting designers to search the detailed design space and achieve an optimal design. The loop of GP operation is iterated until a termination condition is reached or a predefined number of iterations is carried out. The design outcome is then further evaluated for practical implementation.

However, many issues are still to be addressed before this approach can be applied in practice for complex mechatronic systems. For example, arbitrary evolution of a design model of complex system can result in vast computation [19] as well as infeasible outcomes that cannot be implemented in reality [40]. Therefore, it is important to narrow down the search space by detecting possible weaknesses where redesign or design improvement should be conducted.

2.6 Case Study

Now the automated evolution of the reconfiguration and the design of an automated industrial fish cutting system is investigated as a case study. This automated fish cutting system is designed by the Industrial Automation Laboratory of the University of British Columbia and is used in industry to cut the fish head automatically with minimized wastage of fish meat [1, 9]. The conventional machines used in the industry caused about 10%-15% wastage of useful meat, each unit percentage of wastage costing about $5 million annually in the province of British Columbia, Canada [40]. The automated fish cutting system is a multi-domain manufacturing system that consists of mechanical, electrical, hydraulic and pneumatic subsystems [57]. A schematic diagram of the system is shown in Fig.2.5. The present case study employs the proposed closed-loop design improvement architecture for the reconfiguration and design evolution of this engineering system, based on a new set of production requirements.

2.6.1 Machine Condition Monitoring

Within the framework proposed in Section 2.2, both real-time condition data of the system (production speed, capacity, power consumption, size, fish weight, waste percentage, malfunction record, etc.) and condition data from similar machines at other fish-processing plants can be acquired by a variety of sensors and then be transmitted to the cloud database through the network layer. The sensed condition
data provides a rather precise and up to date status of the current manufacturing system. Also, the system model, information of available subsystem alternatives (technologies, devices, parameters, cost, etc.) and design expertise are acquired and transmitted to the cloud platform. Then the design knowledge base is formed with this information to assist the design process.

Two key production requirements of the automated fish cutting system are production speed and percent wastage. The performance limit of the original system is 1.5 fish per second with 3.5% wastage of fish meat according to the sensed condition data. In the present case study, new production requirements are given as: 2.0 fish per second with 2% wastage for the design task. Since the current system is not capable for achieving the new production requirements, the performance of
the system is unsatisfactory. A conceptual redesign of the machine is needed as the new technical solutions for the subsystems are available (e.g. a robotic arm that is better capable and more cost-effective may replace the human operator).

2.6.2 Conceptual Design Stage

First a conceptual design improvement is carried out. Then the outcome of the improved conceptual design is presented to the detailed design process for exploration of the topologies and parameters of each component. This case study illustrates the closed-loop design improvement framework, with a focus on the detailed steps of the multi-criteria decision making of the conceptual design phase. MDQ is utilized here for the multi-criteria evaluation of the design alternatives. The current automated fish cutting system contains five main subsystems as listed in Table 2.1.

Given the new production requirements, the following steps are followed to realize a conceptual design of the automated fish cutting system to satisfy the production requirements and various constraints.

Step 1: Review the design specifications.
The new production requirements are 2.0 fish per second with a 2% wastage limit for fish meat.

Step 2: Determine the system configuration.
According to the expertise in the design knowledge base, the current configuration (feeding, conveying, vision, positioning table and cutter blade) can achieve the task of removing the fish head.

Step 3: Design specification estimation.
Condition monitoring data provides the current system behavior. Based on the current values, the designer can estimate the performance for each subsystem according to the new production requirements. Table 2.2 lists the specifications of the current subsystems and the estimated required specifications corresponding to the new design requirements.

Step 4: Construct the conceptual design space.
For each subsystem, search the knowledge base for available and feasible design alternatives.

Table 2.3 lists the available technology choices for each subsystem to achieve
### Table 2.1: Subsystems of the automated fish cutting system

<table>
<thead>
<tr>
<th>Subsystems</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeding</td>
<td>A human operator will place raw fish in the feeding area of the conveyor, ideally at the same pace of movement of the conveyor to achieve maximum productivity.</td>
</tr>
<tr>
<td>Conveying</td>
<td>An electromechanical conveying subsystem delivers the raw fish from the feeding area to the cutting area and then off the fish cutting system after removing the head. The conveying subsystem is driven by an AC induction motor.</td>
</tr>
<tr>
<td>Vision</td>
<td>There are two primary tasks for the vision subsystem: identifying the optimal cutting location and evaluating the cutting quality, particularly related to the wastage of fish meat and the smoothness of the cut. One image is taken for each fish before it enters the cutting area. Image processing is then performed at a local computer to identify the best reference location for the cut. The corresponding coordinates are sent to the controller of the positioning table control system. After the cut, another image is taken to check the quality of the cut and the percent wastage.</td>
</tr>
<tr>
<td>Positioning Table</td>
<td>This subsystem moves the positioning table that carries the cutting blade accurately and rapidly to the desired cutting location as identified by the vision subsystem. Motion in the horizontal plane, perpendicular to the cutter blade, is controlled to achieve this task. The positioning table is powered by two hydraulic actuators and has two degree of freedom.</td>
</tr>
<tr>
<td>Cutter Blade</td>
<td>This subsystem is assembled on the positioning table. The cutter blade is pushed down by a pneumatic cylinder to cut the fish head when a raw fish is brought to the cutting area by the conveyor and the positioning table with the cutter is moved to the correct location.</td>
</tr>
</tbody>
</table>
### Table 2.2: Specifications of the current subsystems and the estimated design requirements

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Parameter</th>
<th>Current Performance Limit</th>
<th>Required Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeding</td>
<td>Feed speed</td>
<td>1.50</td>
<td>1.50</td>
</tr>
<tr>
<td>Conveying</td>
<td>Output speed</td>
<td>2,400</td>
<td>6,000</td>
</tr>
<tr>
<td></td>
<td>AC Motor #0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>with desired output torque (rpm)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positioning Table</td>
<td>Motion time (s)</td>
<td>0.48</td>
<td>0.45</td>
</tr>
<tr>
<td>Hydraulic solution</td>
<td>Motion accuracy (mm)</td>
<td>5.00</td>
<td>5.00</td>
</tr>
<tr>
<td>Cutter Blade</td>
<td>Cutting time (s)</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>Pneumatic solution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vision Camera</td>
<td>Processing time (s)</td>
<td>0.55</td>
<td>0.55</td>
</tr>
<tr>
<td>kit type A</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Step 5: Determine the criteria for evaluation.

The criteria of "meeting task requirements", "reliability", "matching", "efficiency", "intelligence," and "cost" are chosen as the MDQ attributes for this problem.

Step 6: Reduce the design search space by veto effect criteria.

Among the six criteria, "meeting task requirements" has veto effect. Use this criterion to eliminate any design alternative that cannot meet the task requirements. From Table 2.3, for feeding and cutter blade, their two alternatives can meet the requirements. For the conveying module, the output speed of AC motor #0002 is below the required value. For the positioning table, only the electrical solution can achieve the requirement of motion accuracy. Camera kit type C can fulfill the re-
Table 2.3: Available choices for each subsystem

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Available choices</th>
<th>Estimated cost</th>
<th>Parameter</th>
<th>Performance limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeding</td>
<td>Two operators</td>
<td>$85,000</td>
<td>Feed speed</td>
<td>3.0 (Fish/s)</td>
</tr>
<tr>
<td></td>
<td>Robotic arm</td>
<td>$120,000</td>
<td></td>
<td>2.5 (Fish/s)</td>
</tr>
<tr>
<td>Conveying</td>
<td>AC Motor #0001</td>
<td>$8,000</td>
<td>Output speed</td>
<td>6000 (rpm)</td>
</tr>
<tr>
<td></td>
<td>AC Motor #0002</td>
<td>$5,000</td>
<td>with desired</td>
<td>3800 (rpm)</td>
</tr>
<tr>
<td></td>
<td>AC Motor #0003</td>
<td>$6,000</td>
<td>output torque</td>
<td>5000 (rpm)</td>
</tr>
<tr>
<td>Positioning</td>
<td>Hydraulic solution</td>
<td>$15,000</td>
<td>Motion time</td>
<td>0.45 (s)</td>
</tr>
<tr>
<td>table</td>
<td></td>
<td></td>
<td>Motion accuracy</td>
<td>5.0 (mm)</td>
</tr>
<tr>
<td></td>
<td>Electrical solution</td>
<td>$18,000</td>
<td>Motion time</td>
<td>0.30 (s)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Motion accuracy</td>
<td>2.0 (mm)</td>
</tr>
<tr>
<td>Cutter blade</td>
<td>Hydraulic solution</td>
<td>$9,000</td>
<td>Cutting time</td>
<td>0.15 (s)</td>
</tr>
<tr>
<td></td>
<td>Pneumatic solution</td>
<td>$7,000</td>
<td></td>
<td>0.11 (s)</td>
</tr>
<tr>
<td>Vision</td>
<td>Camera kit type A</td>
<td>$5,000</td>
<td>Processing time</td>
<td>0.65 (s)</td>
</tr>
<tr>
<td></td>
<td>Camera kit type B</td>
<td>$3,000</td>
<td></td>
<td>0.80 (s)</td>
</tr>
<tr>
<td></td>
<td>Camera kit type C</td>
<td>$7,500</td>
<td></td>
<td>0.35 (s)</td>
</tr>
</tbody>
</table>

required image processing within the time limit. The cropped tree structure of the conceptual design space is shown in Fig. 2.6. Eight design alternatives need to be evaluated through multi-criteria evaluation.

Step 7: Assign a fuzzy measure to each subset of the criteria. The remaining five criteria, “reliability”, “matching”, “efficiency”, “intelligence”, and “cost” form a number of $2^5 = 32$ subsets of criteria. It needs 32 fuzzy measures using the ordinary Choquet integral (two of them are self-evident: $v(\emptyset) = 0$ and $v(N) = 1$). Here, the 2-additive Choquet integral is adopted [52]. Thus, only $n(n + 1)/2 = 15$ fuzzy measures need to be specified. The common type of interactions and their fuzzy measure representation are summarized in Table 2.4.

The fuzzy measures are typically assigned by expert designers. The fuzzy measures used in this case study are obtained from [13] since the engineering system is the same and the criteria are similar. The fuzzy measures are as follows: $v_1 = 0.25, v_2 = 0.35, v_3 = 0.22, v_4 = 0.18, v_5 = 0.15, v_{12} = 0.52, v_{13} = 0.45, v_{14} = 0.50, v_{15} = 0.52, v_{23} = 0.50, v_{24} = 0.48, v_{25} = 0.60, v_{34} = 0.45, v_{35} = 0.50, v_{45} = 0.42$. These values reflect a negative correlation between cost and other...
criteria and a small positive correlation between any two criteria except cost.

Step 8: Multi-criteria evaluation of each design alternative. Evaluate each design alternative according to the chosen criteria except the ones with veto effect and assign a score to each design alternative. Evaluation guidelines are found in [13]. Aggregate the partial scores by using 2-additive Choquet integral to determine the global score of each design alternative. Choose the design alternative with the highest global score. The results are shown in Table 2.5.

Figure 2.6: Reduced conceptual design space.
Table 2.4: Typical interactions between criteria

<table>
<thead>
<tr>
<th>Interaction Type</th>
<th>Explanation</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Correlation</td>
<td>High score in criterion $i$ implies a high score in criterion $j$, and vice versa.</td>
<td>$v(ij) &lt; v(i) + v(j)$</td>
</tr>
<tr>
<td>Negative Correlation</td>
<td>High score in criterion $i$ implies a low score in criterion $j$, and vice versa.</td>
<td>$v(ij) &gt; v(i) + v(j)$</td>
</tr>
<tr>
<td>Substitutiveness</td>
<td>Satisfaction of only one criterion produces almost the same effect than the satisfaction of both</td>
<td>$v(T) &lt; \left{ \begin{array}{c} v(T \cup i) \ v(T \cup j) \end{array} \right}$ $\approx v(T \cup ij)$, $T \subseteq N \setminus ij$</td>
</tr>
<tr>
<td>Complementarity</td>
<td>Satisfaction of only one criterion produces a very weak effect compared with the satisfaction of both</td>
<td>$v(T) \approx \left{ \begin{array}{c} v(T \cup i) \ v(T \cup j) \end{array} \right} &lt; v(T \cup ij)$, $T \subseteq N \setminus ij$</td>
</tr>
</tbody>
</table>
Table 2.5: MDQ evaluation of design alternatives

<table>
<thead>
<tr>
<th></th>
<th># 1</th>
<th># 2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feeding module</td>
<td>Human operator</td>
<td>Human operator</td>
<td>Human operator</td>
<td>Human operator</td>
<td>Robotic arm</td>
<td>Robotic arm</td>
<td>Robotic arm</td>
<td>Robotic arm</td>
</tr>
<tr>
<td>Conveyor Motor</td>
<td>#0001</td>
<td>#0001</td>
<td>#0003</td>
<td>#0003</td>
<td>#0001</td>
<td>#0001</td>
<td>#0003</td>
<td>#0003</td>
</tr>
<tr>
<td>Positioning table</td>
<td>Electrical solution</td>
<td>Electrical solution</td>
<td>Electrical solution</td>
<td>Electrical solution</td>
<td>Electrical solution</td>
<td>Electrical solution</td>
<td>Electrical solution</td>
<td>Electrical solution</td>
</tr>
<tr>
<td>Cutter blade</td>
<td>Hydraulic solution</td>
<td>Pneumatic solution</td>
<td>Hydraulic solution</td>
<td>Pneumatic solution</td>
<td>Hydraulic solution</td>
<td>Pneumatic solution</td>
<td>Hydraulic solution</td>
<td>Pneumatic solution</td>
</tr>
<tr>
<td>Vision</td>
<td>Type C</td>
<td>Type C</td>
<td>Type C</td>
<td>Type C</td>
<td>Type C</td>
<td>Type C</td>
<td>Type C</td>
<td>Type C</td>
</tr>
<tr>
<td>Matching</td>
<td>0.70</td>
<td>0.60</td>
<td>0.70</td>
<td>0.60</td>
<td>0.80</td>
<td>0.70</td>
<td>0.80</td>
<td>0.70</td>
</tr>
<tr>
<td>Reliability</td>
<td>0.50</td>
<td>0.40</td>
<td>0.50</td>
<td>0.40</td>
<td>0.50</td>
<td>0.60</td>
<td>0.50</td>
<td>0.60</td>
</tr>
<tr>
<td>Intelligence</td>
<td>0.70</td>
<td>0.60</td>
<td>0.70</td>
<td>0.60</td>
<td>0.90</td>
<td>0.80</td>
<td>0.90</td>
<td>0.80</td>
</tr>
<tr>
<td>Efficiency</td>
<td>0.50</td>
<td>0.60</td>
<td>0.60</td>
<td>0.70</td>
<td>0.60</td>
<td>0.70</td>
<td>0.70</td>
<td>0.80</td>
</tr>
<tr>
<td>Cost</td>
<td>0.85</td>
<td>0.90</td>
<td>0.90</td>
<td>0.95</td>
<td>0.70</td>
<td>0.75</td>
<td>0.75</td>
<td>0.80</td>
</tr>
<tr>
<td>Global score</td>
<td>0.621</td>
<td>0.589</td>
<td>0.651</td>
<td>0.624</td>
<td>0.657</td>
<td>0.694</td>
<td>0.693</td>
<td>0.729</td>
</tr>
</tbody>
</table>
The best conceptual design of the automated fish cutting system is the No.8 design alternative, which corresponds to a conceptual design solution of Robotic arm (feeding module), Motor #003 (conveying), Electrical solution (positioning table), Hydraulic solution (cutter blade), Type C camera kit (vision module). In the case where two or more alternatives have very close global scores, designers can include more features and evaluate the alternatives further.

2.6.3 Detailed Design Stage

After the conceptual design process, detailed design is conducted to specify the topology and tune the parameters to achieve the desired design requirements. GA and GP can be utilized to explore the detailed design space and find the optimal design in the design space. The procedure of evolutionary design with GP is shown in Fig. 2.7.

GP evolution for complex engineering system with numerous subsystems and components can be computationally expensive. A CC platform is utilized here as well to execute the GP computation and find the optimal detailed design with the specific topologies and parameters for each subsystem of the fish cutting machine. The design solution is further evaluated and then physically realized. The redesigned fish cutting machine is then put into production.
Figure 2.7: Procedure of evolutionary design with GP.
Chapter 3

Intelligent Machine Fault Diagnosis using Unsupervised Feature Learning

3.1 Introduction

As introduced in Chapter 2, machine fault diagnosis is a key aspect in the evaluation of the design of an existing system. Due to the advantages of data-driven approaches and the availability of large amounts of condition monitoring data, data-driven approaches of fault diagnosis are considered in this study. Most of the existing data-driven methods have two main deficiencies: (1) Feature extraction needs the specification of representative and robust features by domain experts. This process requires prior knowledge of signal processing and a comprehensive knowledge of the specific problem. For instance, in the fault diagnosis of commonly found rotating machinery, numerous methods have been developed in different domains (e.g., time domain, frequency domain, and time-frequency domain) [58–60]. However, they are only effective for specific types of machinery and hence are case sensitive. (2) Most data-driven algorithms use supervised machine learning techniques, which require large amounts of labeled training data. However, it is difficult to collect labeled data that ensure training of a good model.
In real applications, a large portion of the data collected from a condition monitoring system is unlabeled [61].

Clearly, it is desirable to develop rather intelligent fault diagnosis methodologies that can utilize massive amounts of unlabeled condition monitoring data to learn good features without requiring extensive prior or domain knowledge. Recently, DNN have gained much attention and have achieved great progress in many areas; e.g., computer vision, natural language processing, speech recognition and bioinformatics [62]. With its deep and comprehensive structure, a DNN with unsupervised learning is able to learn useful features from unlabeled raw data [63]. With further fine-tuning using labeled data, a DNN has been shown to achieve remarkable performance in many classification tasks. In the field of fault diagnosis, DNN has been investigated only recently. Tao et al. [64] applied stacked autoencoder and softmax regression in bearing fault diagnosis, leading to high diagnosis accuracy. Feng et al. developed a deeper DNN together with autoencoder and softmax regression to achieve improved classification results [45]. In both approaches, the fine-tuning process used all the labeled training data to achieve satisfactory classification accuracy. In reality, however, it is difficult to acquire the needed amount of labeled data for fine-tuning the model. In addition, the basic autoencoder has the disadvantage that it cannot guarantee the extraction of useful features [65]. DNN has shown the promising capability in generalized learning that it can classify new classes by slightly adjusting the trained model without training from scratch since the hidden layers contain representative features already extracted from the training data. This is beneficial in many real applications since training of a DNN model from scratch is computationally expensive and time-consuming.

Inspired by prior research, this section presents a novel approach of intelligent machine fault diagnosis (IMFD) that addresses the deficiencies of the existing methods. Representative features are extracted automatically from unlabeled condition monitoring data by SDA in an unsupervised manner. By stacking the trained denoising autoencoder, a DNN is constructed to perform intelligent fault diagnosis after fine-tuning the model with a few items of available labeled data. In this approach, a massive amount of easily accessible unlabeled condition monitoring data is utilized to learn useful and robust features. Only a few items of labeled data are needed, which is an advantage in a practical application. In addition, after fur-
ther fine-tuning of the trained DNN, newly occurring conditions can be correctly classified by the proposed method. Rotating machinery is used as an example to demonstrate the proposed method. Due to the end to end learning capability, this method can be applied to fault diagnosis of other systems using corresponding sensed data.

3.2 Unsupervised Feature Learning

A DNN has multiple layers, each having a number of nodes with nonlinear transformations. With its deep architecture, DNN is capable of extracting the discriminative features from the input data through a large number of linear and nonlinear transformations [63]. Unsupervised learning as in the autoencoder can learn representative features from unlabeled raw data [66]. Unsupervised learning is effective in intelligent fault diagnosis since it does not necessarily use labeled condition monitoring data, which are hard to acquire. Stacked denoising autoencoder (SDA), an improved version of the autoencoder, is utilized in the approach proposed in the present work to learn features from the condition signals of the monitored system.

3.2.1 Autoencoder

The autoencoder is a three-layer neural network that reconstructs the input data in the output layer [67]. It seeks to reconstruct the original input by minimizing the reconstruction error in an unsupervised manner. Fig. 3.1 shows the structure of the autoencoder.

The autoencoder uses a deterministic mapping \( f_\theta \) with parameters \( \theta = \{W, b\} \) to map the input vector \( x \in \mathbb{R}^d \) to a hidden representation \( y \in \mathbb{R}^{d'} \) according to:

\[
y = f_\theta(x) = s(Wx + b)
\]  

(3.1)

where \( s \) is the activation function, \( W \) is a \( d' \times d \) weighting matrix, and \( b \) is a bias vector. Then the representation \( y \) is mapped back to a reconstructed vector \( z \in \mathbb{R}^d \) by:

\[
z = g_{\theta'}(y) = s(W'y + b')
\]  

(3.2)

where \( \theta' = \{W', b'\} \). The autoencoder is said to have tied weight if the weight
matrix $W'$ of the reverse mapping is chosen to be $W' = W^T$. The parameters of the model are optimized by minimizing the average reconstruction error as given by:

$$\theta^*, \theta'^* = \arg\min_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^{n} L(x^{(i)}, z^{(i)}) = \arg\min_{\theta, \theta'} \frac{1}{n} \sum_{i=1}^{n} L(x^{(i)}, g_{\theta'}(f_\theta(x^{(i)})))$$ (3.3)

where $L$ is a loss function such as the squared error $L(x, z) = ||x - z||^2$.

Autoencoder is introduced to initialize a DNN using the representation of the $k$th layer as the input for the $(k + 1)$th layer [68]. After each layer has been initialized, these layers are stacked to form the DNN. Supervised training is then conducted to fine-tune the network [69]. The layer-wise initialization has shown significant improvement in achieving better local minima than random initialization of deep networks. However, the autoencoder may not guarantee the extraction of useful features as it can lead to the obvious solution that simply copies the input or uninteresting ones that trivially maximize mutual information.
3.2.2 Denoising Autoencoder

The denoising autoencoder is an extension of a basic autoencoder, and aims at learning more suitable and robust representations to initialize a deep network [70]. A denoising autoencoder is trained to rebuild a repaired clean input from a corrupted version of it. First, the initial input \( x \) is corrupted into \( \tilde{x} \) through the stochastic mapping \( \tilde{x} \sim q_D(\tilde{x}|x) \). The basic autoencoder is then used to map the corrupted input \( \tilde{x} \) to a hidden representation by \( y = f_{\theta}(\tilde{x}) = s(W\tilde{x} + b) \) (3.4)

Subsequently, the reconstruction is done by the reverse mapping \( z = g'_{\theta}(y) \). The loss function of squared error \( L_H(x,z) = ||x - z||^2 \) is minimized by updating the parameters. A schematic representation of the procedure of denoising autoencoder is shown in Fig. 3.2.

![Figure 3.2: Schematic diagram of the procedure of denoising autoencoder.](image)

The parameters \( \theta \) and \( \theta' \) are updated through a training set by minimizing the average reconstruction error between and the uncorrupt input \( x \), which is the same as the basic autoencoder. The difference between the denoising autoencoder and the basic autoencoder is that \( z \) here is a deterministic function of \( \tilde{x} \) rather than \( x \). It thus encourages the learning of a more clever mapping than the identity: one that extracts features useful for denoising [71]. The layer-wise procedure is the same as that of the basic autoencoder. The input corruption is used only for the training of each layer to learn useful representations. After the mapping \( f_{\theta} \) is learned, uncorrupt inputs are used to produce a representation that will serve as the clean input to the following layer. Different types of corruption processes may
be considered such as additive isotropic Gaussian noise, salt-and-pepper noise and masking noise [70]. Masking noise is used in this work where a fraction of the elements of \( x \) (chosen at random for each sample) is forced to zero.

### 3.2.3 Softmax Regression Classifier

The softmax regression is a generalized form of logistic regression, which is often used for multiclass classification [72]. Given a \( k \) class input training set with \( n \) samples \( \{ x^{(i)} \}_{i=1}^{n} \) where \( x^{(i)} \in \mathbb{R}^{m} \) and its label set \( \{ t^{(i)} \}_{i=1}^{n} \) where \( t^{(i)} \in \{1, 2, \ldots, k\} \), softmax regression estimates the probability of each input sample belonging to each class. The probability is given by

\[
P(t^{(i)} = j | x^{(i)}; \theta) = \frac{1}{\sum_{l=1}^{k} e^{\theta_{l}^{T}x^{(i)}}} \begin{bmatrix} e^{\theta_{1}^{T}x^{(i)}} \\ e^{\theta_{2}^{T}x^{(i)}} \\ \vdots \\ e^{\theta_{k}^{T}x^{(i)}} \end{bmatrix} \tag{3.5}
\]

where \( j = 1, 2, \ldots, k \). \( \theta = [\theta_{1}, \theta_{2}, \ldots, \theta_{k}] \) are the parameters of the softmax regression model and \( 1/\sum_{l=1}^{k} e^{\theta_{l}^{T}x^{(i)}} \) normalizes the distribution so that the summation of the probability is one.

The cost function of softmax regression is defined as:

\[
J(\theta) = -\frac{1}{n} \left[ \sum_{i=1}^{n} \sum_{j=1}^{k} 1 \{ t^{(i)} = j \} \log \frac{e^{\theta_{j}^{T}x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_{l}^{T}x^{(i)}}} \right] \tag{3.6}
\]

where \( 1 \cdot \) is the indicator function, which returns 1 if the condition is true and 0 otherwise. The parameters are updated by minimizing the cost function \( J(\theta) \) over the training dataset. In this work, softmax regression is used to generate the classes of input data.

### 3.3 Feature Learning and Fault Diagnosis using SDA

This dissertation proposes an intelligent fault diagnosis approach based on SDA and DNN. It learns representations from a massive set of unlabeled condition data; e.g., vibration signals of a bearing, gearbox or induction motor with different con-
ditions by unsupervised learning using SDA. Spectrum of the vibration signals are used as the input data in this work. This method can automatically achieve machine condition classification after fine-tuning the model with very few labeled condition data. In addition, new conditions can also be classified after further fine-tuning of the trained model with a few labeled observations of the new conditions. The scheme of the proposed approach is shown in Fig. 3.3 with the following main steps.

Step 1: Data acquisition and pre-processing
Condition monitoring data can be acquired through various sensors such as accelerometer, pressure sensor, thermometer, and cameras depending on the characteristics of the monitored system. In this work, vibration signals are used and the spectrum of the time signal is obtained by applying fast Fourier transform (FFT).

Step 2: Initialization of the DNN
A DNN with $N$ hidden layers is initialized with random parameters. The number of layers can be increased until the model starts to overfit the training data as suggested in [73]. The number of nodes in the first layer can be set equal to or slightly larger than the dimension of the input data. The number of nodes in the subsequent layers is usually decreased gradually.

Step 3: Unsupervised feature learning
Denoising autoencoder is applied here to learn representative features from the unlabeled data as shown in Section 3.2.2. Layer wise training is carried out before the layers are stacked.

Step 4: Supervised fine-tuning of the model
After Step 3, a softmax layer is added on top of the DNN. Labeled condition data is used to fine-tune the parameters of the DNN by stochastic gradient descent.

Step 5: Fault diagnosis with the trained DNN
With the trained DNN, fault diagnosis can be carried out on the acquired condition data.

Step 6: Further fine-tuning and fault diagnosis of new conditions
As new conditions occur, the trained model can be further fine-tuned with a few labeled condition data of the new conditions. The further fine-tuned model can diagnose the machine conditions including the new ones.
Figure 3.3: The scheme of the SDA based feature learning and fault diagnosis.

### 3.3.1 Unsupervised Feature Learning with SDA

First, a denoising autoencoder is used to learn representations from massive unlabeled condition data. A DNN with $N$ hidden layers can be pre-trained by stacking $N$ denoising autoencoders. Consider an unlabeled dataset $\{x^{(i)}\}_{i=1}^n$ where $x^{(i)} \in \mathbb{R}^m$. Each $x^{(i)}$ can be the Fourier transform of a vibration signal of a running machine, e.g., bearing, gearbox and motor under different conditions including normal and various faulty conditions. The parameter of the first autoencoder $\theta_1 = \{W_1, b_1\}$ is randomly initialized and then is trained following the procedure shown in Fig. 3.2 by minimizing the reconstruction error of the input data $x^{(i)}$ and the reconstructed data $z^{(i)}$ in Equation 3.3. The sigmoid function is used here as the activation function. The trained parameters of the first encoder, $\theta_1$ is stored for initialization of the first hidden layer of the deep network. The learned encoding function $f^{(1)}_{\theta}$ is then used on the uncorrupted input. The resulting representation is used as the input to train the next level denoising autoencoder to obtain the next level encoding function $f^{(2)}_{\theta}$. The trained parameters of the next encoder are used for initialization of the next hidden layer. This procedure is repeated as shown in Fig. 3.4 until the parameters of all hidden layers of the deep network are updated.
3.3.2 Fine-tuning a DNN for Classification

In this work, after the hidden layers are trained and stacked by the denoising autoencoder, an output layer, the softmax regression layer is added on top of the stack. A small amount of labeled data is then used to fine-tune the parameters of the entire network by minimizing the error in the predicted conditions. Fig. 3.5 shows the fine-tuning procedure. Stochastic gradient descent based backpropagation algorithm is used to update the parameters of all hidden layers by minimizing the error between the predicted label and the actual label of the labeled dataset. The parameters $W$ and $b$ can be updated as:

\[
W_l = W_l - \eta \frac{\partial}{\partial W_l} J(W, b; X, t) \tag{3.7}
\]

\[
B_l = B_l - \eta \frac{\partial}{\partial B_l} J(W, B; X, t) \tag{3.8}
\]

where $W_l$ and $B_l$ are the weights and bias, respectively, of the $l$-th layer; $\eta$ is the learning rate; $(X, t)$ is a batch containing $m$ training samples; $X$ are the spectrum of vibration signals and $t$ are the corresponding labels.

After fine-tuning, the deep network is capable of classifying conditions in the current condition space. In order to diagnose new conditions, another small amount of labeled data of the corresponding new conditions is combined with the fine-tuning dataset. With the new fine-tuning dataset, the parameters of the DNN can be further fine-tuned with the same procedure as shown in Fig.3.5 to achieve fault diagnosis in the new condition space. The proposed approach enables effective feature learning and diagnosis with new conditions by simply fine-tuning the DNN with a few labeled data rather than training from scratch, which needs a large amount of training data and computational resources.

Figure 3.4: The procedure of stacking denoising autoencoder.
3.4 Experiment Studies

This section presents a case study on fault diagnosis of a motor bearing unit. It serves to illustrate the application of the methodology developed in this work.

3.4.1 Data Description

In the present case study, the dataset of motor bearing vibration signals from Case Western Reserve University (CWRU) is analyzed [74]. The experimental setup includes a 2 hp motor, a torque transducer and a dynamometer shown in Fig. 3.6. Vibration data has been collected from an accelerometer at the drive end bear-
ing under different conditions: normal condition, ball defect (BD), outer race defect (OR) and inner race defect (IR). For the conditions with defects, three levels of defect severity (0.18, 0.36 and 0.53mm) have been introduced. Thus, 10 bearing conditions in total are included in the dataset of the present case study, as shown in Table 3.1. The vibration signals have been collected under four load conditions (0, 1, 2 and 3 hp) with a sampling frequency of 12 kHz. For each bearing condition, 100 samples with 1200 data points have been collected from each load condition. Then fast Fourier transformation is applied to each signal to obtain the spectrum. Since the Fourier coefficients are symmetric, only the first half of the Fourier coefficients are used to compose each data sample. The dimension of each data sample is thus 600. There are in total 4000 samples in the dataset with 400 samples for each bearing condition.

A five layer DNN is constructed here including an input layer, three hidden layers, and a softmax output layer. The number of neurons of the input layer is equal to the dimension of the input data which is 600. The number of neurons of the three hidden layers are selected as 400, 200 and 50. The number of neurons of the softmax layer is 10 which is the number of conditions. The sigmoid function is used as the activation function in the hidden layers. The learning rate is set at 0.05
Table 3.1: Conditions of the bearing in the dataset

<table>
<thead>
<tr>
<th>Condition Type</th>
<th>Defect Severity (mm)</th>
<th>Class Label</th>
<th>Sample Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0</td>
<td>1</td>
<td>400</td>
</tr>
<tr>
<td>BD</td>
<td>0.18</td>
<td>2</td>
<td>400</td>
</tr>
<tr>
<td>BD</td>
<td>0.36</td>
<td>3</td>
<td>400</td>
</tr>
<tr>
<td>BD</td>
<td>0.54</td>
<td>4</td>
<td>400</td>
</tr>
<tr>
<td>OR</td>
<td>0.18</td>
<td>5</td>
<td>400</td>
</tr>
<tr>
<td>OR</td>
<td>0.36</td>
<td>6</td>
<td>400</td>
</tr>
<tr>
<td>OR</td>
<td>0.54</td>
<td>7</td>
<td>400</td>
</tr>
<tr>
<td>IR</td>
<td>0.18</td>
<td>8</td>
<td>400</td>
</tr>
<tr>
<td>IR</td>
<td>0.36</td>
<td>9</td>
<td>400</td>
</tr>
<tr>
<td>IR</td>
<td>0.54</td>
<td>10</td>
<td>400</td>
</tr>
</tbody>
</table>

[BD denotes ball defect; OR denotes outer race defect; IR denotes inner race defect.]

and the fraction of masked zero is 0.1.

In order to investigate the performance of the proposed approach in the diagnosis of new conditions, the dataset is divided into the two parts: dataset B with one condition (IR with 0.53 mm defect severity) as the dataset of a new condition and dataset A with all the remaining nine conditions as the original conditions. 75% of the samples in dataset A are randomly selected to form the training dataset C and the remaining 25% to form the testing dataset D. Then 95% of the dataset C is randomly chosen as the unsupervised training dataset E and the remaining 5% as the fine-tuning dataset F. Fig. 3.7 shows the dataset structure.

3.4.2 Fault Diagnosis using the Proposed Approach

In the first part, unsupervised feature learning and fine-tuning are performed on dataset A, which contains the original nine conditions. When performing SDA on dataset E, we assume that no labeled information is available for these samples, thus unsupervised feature learning is needed. Fine-tuning of the deep network is then conducted with the labeled data in dataset F. Testing dataset D is used to check the diagnosis performance of the trained DNN on the original nine classes. The experiment is repeated 20 times with a maximum epoch of 200 to reduce the randomness. Fig. 3.8(a) shows the training and testing accuracy over nine
bearing conditions. All the trials achieve more than 97% diagnosis accuracy and the average testing accuracy is 97.88%. This demonstrates the effectiveness of the proposed approach in intelligent fault diagnosis using a large amount of unlabeled data and a few items of labeled data.

In the next part, fault diagnosis of the new condition space is performed. With the proposed approach, the trained DNN in the first part only needs further fine-tuning with a few items of labeled data, rather than training from scratch. 15% of the data is randomly selected from dataset G to form dataset I. Dataset F and dataset I form the new fine-tuning dataset for further fine-tuning. The new testing dataset is composed by combining dataset D and dataset I. Also, 20 trials are run and the diagnosis result form that exercise is shown in Fig. 3.8(b). The testing accuracy of ten classes is compared with the testing accuracy of the original nine classes. All trials obtain a testing accuracy of more than 96.50% and the average testing accuracy is 97.59%. It shows that after further fine-tuning with very few items of labeled data that contains the new condition, the proposed method performs well in fault diagnosis in the new condition space.

An experiment is conducted to assess the robustness of the proposed approach to noise. The original vibration signal is corrupted using different levels of white Gaussian noise with the signal-to-noise ratio (SNR) from 30 dB to 0 dB, and a step
size of 6 dB. For each level of noise, 20 trials are carried out and the average result is listed in Table 3.2. The training accuracy of nine classes, the testing accuracy of nine classes, and the testing accuracy of ten classes is represented by $R_1$, $R_2$, and $R_3$ respectively. The result shows that the classification accuracies with an SNR of 30 dB and 24 dB are almost the same as those without noise. When the noise level
Table 3.2: Classification results with different levels of noise

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>30</th>
<th>24</th>
<th>18</th>
<th>12</th>
<th>6</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>97.29</td>
<td>97.21</td>
<td>96.19</td>
<td>96.58</td>
<td>95.05</td>
<td>82.38</td>
</tr>
<tr>
<td>$R_2$</td>
<td>97.42</td>
<td>96.67</td>
<td>96.20</td>
<td>95.86</td>
<td>94.05</td>
<td>77.89</td>
</tr>
<tr>
<td>$R_3$</td>
<td>96.26</td>
<td>96.28</td>
<td>98.17</td>
<td>96.53</td>
<td>95.34</td>
<td>81.50</td>
</tr>
</tbody>
</table>

$R_1$ is the training accuracy of nine classes; $R_2$ is the testing accuracy of nine classes; $R_3$ is the testing accuracy of ten classes.

is increased and the SNR is 18 dB and 12 dB, the accuracies decrease slightly. Even when the SNR is 6 dB, the proposed approach can still achieve a 94.05% testing accuracy of nine classes and 95.34% testing accuracy of ten classes. The performance of the approach degrades when the SNR is 0 dB where the power of the noise is equal to that of the original signal. This is the case with noise isolation and other signal processing methods should be considered when the signal is acquired or pre-processed. Overall, the DNN based approach is robust when the signal is corrupted by noise.

A comparison of the proposed approach with typical data-driven approaches is conducted now. Statistical features in time domain and frequency domain are extracted and used for the fault classification using linear SVM, quadratic SVM, kNN, and weighted kNN. All the data are treated as labeled data in using the four supervised learning methods indicated above. Ten features: absolute mean, variance, crest, clearance factor, kurtosis, crest factor, root mean square, pulse factor, skewness, and shape factor, are selected in the time domain and five features including average frequency, crest, kurtosis, mean energy, and variance are selected in the frequency domain. The testing accuracies for both nine classes and ten classes are listed in Table 3.3.

The results show that the proposed approach has achieved the highest testing accuracy in the nine classes dataset even with only 5% of the data labeled. Also, with only 15% of the data of the tenth condition labeled, it achieves better result than three of the other approaches (linear SVM, kNN, and weighted kNN) and only slightly lower than the quadratic SVM. However, those four approaches require 100% of the data to be labeled.
Table 3.3: Comparison of the proposed approach with SVM and kNN

<table>
<thead>
<tr>
<th></th>
<th>Linear SVM</th>
<th>Quadratic SVM</th>
<th>kNN</th>
<th>Weighted kNN</th>
<th>Proposed approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2 (%)</td>
<td>96.70</td>
<td>97.70</td>
<td>95.30</td>
<td>95.80</td>
<td>97.88</td>
</tr>
<tr>
<td>R3 (%)</td>
<td>97.50</td>
<td>98.20</td>
<td>96.40</td>
<td>96.70</td>
<td>97.59</td>
</tr>
</tbody>
</table>

$R_2$ is the testing accuracy of nine classes; $R_3$ is the testing accuracy of ten classes.

3.4.3 Effect of the Size of Labeled Data

The proposed approach is valuable in situations where the availability of labeled data is quite limited while there exists extensive unlabeled data. It is important to investigate the robustness of the method to different ratios of labeled data to unlabeled data. Therefore, further study is done by changing the fraction of unlabeled data that is used for unsupervised feature learning from 1% to 15% with a step size of 1%. Fig. 3.9(a) shows the result of the average testing accuracy after 20 trials for each step size. The diagnosis accuracy rises rapidly when the fraction of labeled data is increased from 1% to 4%. It is seen that even with 3% of labeled data, the diagnosis accuracy is around 95%, which shows that the features learned from unlabeled data are representative. With further increase of labeled data, the diagnosis accuracy tends to increase slightly and becomes stable. The proposed approach achieves satisfactory diagnosis accuracy even with very few items of labeled data.

In the diagnosis of new conditions, the effect of the amount of labeled new condition data that is used in further fine-tuning of the model is studied. Experiments are run using different amounts of new condition data for further fine-tuning. Fig. 3.9(b) shows the diagnosis result based on different amounts of new condition data. The horizontal axis indicates the percentage of data selected from dataset G, which forms dataset I. It is seen that the testing accuracy on the new condition improves rapidly when the percentage of new condition data is increased from 5% to 15%. This is because, with the increase of the labeled new condition data, the DNN is tuned to be more accurate in detecting the new condition. The testing accuracy of the other nine classes decreased slightly with the increase of the new condition data. With further increase of the new condition data, the testing accuracy of both
the original nine classes and the new class becomes stable. The result shows that the proposed method performs satisfactorily in detecting the new condition after further fine-tuning of the DNN.

Figure 3.9: Effect of the size of labeled data
3.4.4 Visualization of Learned Features

The features learned from the SDA as well as the features fine-tuned by the DNN are visualized by using t-distributed Stochastic Neighbor Embedding (t-SNE) which is an effective tool to visualize the characteristics of high dimensional data [75]. Before running t-SNE, principal component analysis is applied to speed up the additional computation by reducing the dimension of the features to 30. Then, t-SNE is used to map the 30-dimensional data to a two-dimensional map that can visualize the learned features from the proposed approach. Fig. 3.10(a) shows the scatter plots of the features learned from the SDA. It is seen that after unsupervised learning over unlabeled data, the features can accurately distinguish many of the conditions. After fine-tuning the DNN by using only a small amount of labeled data, the data of different conditions are separated clearly, as shown in Fig. 3.10(b). It indicates that the features learned by the proposed method are representative. Fig. 3.10(c) presents the features of the new condition space after further fine-tuning with the labeled data of the new condition. The result shows that the features work well in clustering the conditions with the new one. The proposed approach is capable of learning representative features by SDA-based unsupervised learning and fine-tuning with very few items of labeled data.

In summary, the intelligent fault diagnosis approach based on SDA and DNN is able to learn representative features automatically from a massive quantity of unlabeled condition data. Only a few items of labeled data are needed to fine-tune the DNN to perform fault diagnosis. In addition, the proposed approach is able to utilize the trained model to diagnose new conditions by further fine-tuning the DNN with a few items of labeled data of the new condition. The effectiveness of the developed approach was verified by using a standard dataset of bearing faults. The robustness of the method to noise was evaluated by corrupting the original signal with different levels of noise. With the comparison of traditional fault diagnosis methods, the proposed method was shown to overcome the drawbacks of (1) features are extracted with prior knowledge, (2) a large amount of labeled data is needed, and (3) diagnosis model has to be rebuilt for diagnosing new conditions.
(a) Features learned from the unsupervised SDA.

(b) Features of nine classes after fine-tuning.

(c) Features of ten classes after further fine-tuning.

Figure 3.10: Visualization of learned features
Chapter 4

IMFD using Convolutional Neural Networks and Sensor Fusion

4.1 Introduction

Recent research has shown that an estimator employing multiple sensors with sensor fusion techniques can provide enhanced and robust estimates [76], [77]. Sensor fusion can be classified into three categories: data level, feature level, and decision level [78]. Data level sensor fusion can achieve highest performance because it loses less information than the other two categories. The convolutional neural networks (CNN) model is designed for processing two dimensional (2D) or three dimensional (3D) input data. This property has the potential to incorporate sensor fusion to improve the diagnosis accuracy and reliability. For instance, temporal signals from different locations can be aligned into a 2D matrix as the input to a CNN model where the temporal and spatial information is integrated.

Moreover, CNN overcomes the limitation of the regular fully connected DNN in solving more complex problems. The parameters of DNN can grow exponentially when more layers are added to the model. It can lead to high computational effort or the overfitting problem. Compared with the standard DNN with all fully
connected layers, a CNN is constructed using fewer connections through shared filters. Training of a CNN is easier and it uses less computational resource and less time. Another advantage of the CNN model is that it is less likely to cause overfitting with the same available training data. With the linear and nonlinear layers in the model, CNN has shown a strong capability in learning sensitive and robust features [79].

Chen et al. [80] proposed an approach using CNN for gearbox fault diagnosis with vibration signals and achieved high classification accuracy. However, manual feature extraction was still needed to form the input for their CNN model. Janssens et al. [81] developed a three layer CNN model for bearing fault detection with vibration signals. However, it could not work on raw data and discrete Fourier transform was needed. More recently, Guo et al. [82] proposed a hierarchical adaptive deep CNN for bearing fault diagnosis from raw vibration data. However, the convergence was quite slow and one-dimensional raw vibration data was arbitrarily converted into a square matrix as the input to the CNN model.

With the above advantages and capabilities of CNN, the present research proposes a CNN-based fault diagnosis approach using signals from multiple sensors. Raw signals are directly used as the input to the model to detect different failures. Signals from multiple sensors are fused at the data level in this model to increase the accuracy and reliability of the diagnosis. With mini-batch stochastic gradient descent, the parameters of the network can be tuned efficiently to obtain a CNN-based fault prediction model. A dropout technique is adopted in this approach to decrease the likelihood of overfitting. Representative features are extracted automatically through feature learning of the CNN-based model. Fault diagnosis of rotating machinery is used in this study to illustrate the process of the proposed method. The performance of the method is evaluated through both a roller bearing dataset and a gear transmission dataset.

4.2 Convolutional Neural Networks

CNN are an important class of DNN and have been successfully applied in various classification problems due to its capability of feature extraction [83]. CNN is composed of trainable multi-stage architectures involving linear and nonlinear
operations. The input and output of each stage are sets of arrays called feature maps [84]. Typically, each stage includes two layers: a convolution layer and a feature pooling layer. A typical CNN is constructed by stacking one or a multiple of such 2-layer stages together with a classification layer, e.g., a softmax layer. The feed-forward process can be represented:

\[ f(X) = f_L(\ldots f_2(f_1(X, \theta^{(1)}), \theta^{(2)}), \ldots), \theta^{(L)}) \]  

(4.1)

Here \( X \) is the input raw data, e.g., an image, an audio sequence or vibration signals from a machine condition monitoring system; \( \theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(L)} \) are learnable parameters such as weights and biases at each of the \( L \) stages; and \( f_1, f_2, \ldots, f_L \) are operations at each stage. Outputs of these functions are intermediate feature maps. For computer vision applications, the input of the network is usually a 2D array of pixels if the image is of grey scale, or a 3D array for typical images with red-green-blue (RGB) channels.

### 4.2.1 Convolution Layer

In a convolution layer, the input is convolved with a bank of learnable filters (simply known as kernels) to generate new feature maps as the input to the next layer [26]. The operation can be expressed by:

\[ X_{k'}^{(l)} = f \left( \sum_{k=1}^{K} W_{kk'}^{(l)} * X_k^{(l-1)} + B_k^{(l)} \right) \]  

(4.2)

Here \( l \) donates the layer number of the network; \( k' = 1, 2, \ldots, K' \) is the index of the output feature maps; and \( k = 1, 2, \ldots, K \) is the index of the input feature maps, where \( K = 1 \) at the first layer if the input data is a 2D array. The \( * \) denotes the 2D discrete convolution operator applied to the \( k \)th filter \( W_{kk'}^{(l)} \) at the \( l \)th layer with the \( k \)th feature map \( X_k^{(l-1)} \) from the \( (l-1) \)th layer. A bias matrix \( B_{k'}^{(l)} \) is then added to the convolutional outcome. Finally, a nonlinear activation function \( f \) is applied point-wise on each element of the feature maps. Typical nonlinear activation functions include hyperbolic tangent function, sigmoid function, and rectified linear unit (ReLU). In the present work, ReLU is used due to its superior
performance as reported in a recent work [85]. ReLU is given by:

\[ y_{ijk} = \max(0, x_{ijk}) \]  

(4.3)

where \( x_{ijk} \) is the \((i, j)\) component of the \(k\)th feature map.

### 4.2.2 Feature Pooling Layer

Another important operation in a CNN is pooling, which achieves spatial invariance by reducing the resolution of the feature maps [86]. A pooling operator is applied to each feature map separately by fusing nearby feature values into one value through a suitable operator such as max-pooling (using the max operator) or average-pooling (using the average operator). The neighborhoods can be stepped by a stride larger than 1. The pooling window can be of different size. Max-pooling is increasingly used in recent models, given by:

\[ y_{ijk} = \max(y_{i'j'k} : i \leq i' < i + p, j \leq j' < j + q) \]  

(4.4)

where \( p \) is the length of the pooling window and \( q \) is the width. The maximum in the neighborhood is selected to be the value of that area. Thus, the pooling layer produces a feature map of lower resolution.

### 4.2.3 Softmax Layer

The softmax regression is often used for multiclass classification as a generalized form of logistic regression [72]. Given a training dataset of \(k\) classes with \(m\) samples \(\{x^{(i)}\}_{i=1}^{m}\) where \(x^{(i)} \in \mathbb{R}^n\) and its label set \(\{t^{(i)}\}_{i=1}^{m}\) where \(t^{(i)} \in \{1, 2, \ldots, k\}\), softmax regression estimates the probability of each input sample belonging to each class. The probability is calculated by:

\[
P(t^{(i)} = j | x^{(i)}; W^{(L)}) = \left( \sum_{l=1}^{k} e^{(W_l^{(L)})^T x^{(i)}} \right)^{-1} \times \left[ e^{(W_1^{(L)})^T x^{(i)}} e^{(W_2^{(L)})^T x^{(i)}} \ldots e^{(W_k^{(L)})^T x^{(i)}} \right]^T
\]

(4.5)
where \( j = 1, 2, \ldots, k \). \( W^{(L)} = [W_1^{(L)}, W_2^{(L)}, \ldots, W_k^{(L)}] \) are the parameters of the softmax regression model and \( 1/\sum_{l=1}^{k} e^{(W_l^{(L)})^T x^{(i)}} \) normalizes the distribution so that the summation of the probability is unity.

The cost function of softmax regression is defined as:

\[
J(W^{(L)}) = -\frac{1}{m} \left[ \sum_{i=1}^{m} \sum_{j=1}^{k} 1\{t^{(i)} = j\} \log \frac{e^{(W_j^{(L)})^T x^{(i)}}}{\sum_{l=1}^{k} e^{(W_l^{(L)})^T x^{(i)}}} \right]
\]  

(4.6)

where \( 1\{\cdot\} \) is the indicator function, which returns 1 if the condition is true and 0 otherwise. The parameters are updated by minimizing the cost function \( J(W^{(L)}) \) over the training sample. In the present work, softmax regression is used to generate the classes of the input data.

4.2.4 Mini-batch Stochastic Gradient Descent

Supervised training is performed using gradient descent, which can be implemented through the backpropagation algorithm [87]. The weights of all the filters of the CNN are updated through the learning procedure to minimize the loss function that captures the difference between the target output and the predicted output of the CNN. Instead of updating the weights over the entire training dataset in the standard gradient descent method, which can be computationally expensive, mini-batch stochastic gradient descent updates the weights by the average gradient over a small batch of training samples. The parameters \( W \) and \( B \) are updated according to:

\[
W^{(l)} = W^{(l)} - \eta \frac{\partial}{\partial W^{(l)}} J(W, B; X, Y)
\]  

(4.7)

\[
B^{(l)} = B^{(l)} - \eta \frac{\partial}{\partial B^{(l)}} J(W, B; X, Y)
\]  

(4.8)

where \( \eta \) is the learning rate; \((X, Y)\) is a batch containing \( m \) training samples; \( X \) are the raw signals; and \( Y \) are the corresponding labels.

With supervised training using labeled data, the banks of filters at different convolutional layers are tuned in an automatic manner. For example, in an image classification task, the learned filters are found to be edges in the first layer, object
parts in the intermediate layer, and complicated object models in later layers. The classification layer can use the learned bank of filters (or the extracted features) to achieve the classification.

### 4.3 IMFD using CNN and Sensor Fusion

The present work proposes an IMFD approach based on CNN using raw data from multiple sensors. With the appealing capability of automatic feature extraction of the approach, no hand-craft features are needed to classify different conditions. Multiple sensor fusion at data level is achieved by combining the raw data from multiple sensors into a 2D matrix at the input layer. Fig. 4.1 shows the flowchart of the proposed fault diagnosis method. Condition monitoring data of the running machinery is acquired from multiple sensors and fused together. The fused data is then used to train a CNN model. The trained CNN model can then be used to diagnose the condition of the machinery.

![Flowchart of the proposed fault diagnosis approach.](image)

**Figure 4.1:** Flowchart of the proposed fault diagnosis approach.
Machine condition monitoring data $X_i^n, (i = 1, 2, ..., m)$ from $m$ vibration sensors is collected and fused at the data level as the input $X \in \mathbb{R}^{m \times n}$ of the CNN model. The input is convolved by $K_1$ filters of size $p_1 \times q_1 \times 1$. The ReLU operation is applied on the convolved outcome to form the $K_1$ feature maps with dimension $(m - p_1 + 1) \times (n - q_1 + 1)$. A max-pooling layer is followed to subsample the feature maps by using Equation (4.4). Followed by another such stage, the convolution process aims to capture the representative features from the input data. A fully connected layer and a softmax layer are added next to generate the machine condition. Mini-batch stochastic gradient descent is used in this work to update the parameters of the model in the training process using Equation (4.6) through Equation (4.8). After training, the CNN model extracts representative features directly from the raw vibration signals from multiple sensors. Fault diagnosis can then be performed on new monitoring data.

Overfitting is a common problem in training, which leads to a poor perfor-
mance on the test data especially with limited training data. The present work uses dropout to prevent overfitting. Dropout is a technique that avoids extracting the same features repeatedly to reduce the possibility of overfitting [88]. During each iteration of training, neurons are randomly dropped out, which means temporarily removed from the network, along with all their incoming and outgoing connections with probability $p$, so that a reduced network is left for training [89]. It can be implemented by setting the selected elements of the feature maps to be zero. In the testing phase, the dropout is turned off and the probability $p$ is multiplied by each feature map element. Dropout is considered to combine exponentially many different neural network architectures in an efficient way to find the fittest model.

4.4 Experimental Studies

To evaluate the effectiveness of the proposed approach for the fault diagnosis of machinery, two practical rotating devices, bearings and gearboxes, are investigated in the present work. Vibration signals of different machine conditions are collected from multiple accelerometers. The parameters of the model are trained and updated through the training samples. Then the test samples are provided to the trained model to evaluate the fault diagnosis accuracy of the proposed approach.

4.4.1 Bearing Fault Diagnosis

Experimental setup and data description

In this case study, the publicly available roller bearing condition dataset collected from a motor drive system by CWRU is analyzed [74]. The objective is to diagnose different faults of bearing, with different levels of severity. The main components of the experimental setup are a 2 hp motor, a torque transducer and a dynamometer as shown in Fig. 3.6. Vibration signals have been collected using accelerometers mounted at three different locations: the drive end, the fan end, and the base. Bearings under different conditions are tested including normal condition, ball defect (BD), outer race defect (OR), and inner race defect (IR). Each defect type (BD, OR, and IR) has three levels of severity. A single point fault was introduced to each bearing by electro-discharge machining with fault diameters of 0.18 mm,
Figure 4.3: Vibration signals of bearings with different conditions from one sensor. (a) BD 0.18 mm. (b) BD 0.36 mm. (c) BD 0.53 mm. (d) OR 0.18 mm. (e) OR 0.36 mm. (f) OR 0.53 mm. (g) IR 0.18 mm. (h) IR 0.36 mm. (i) IR 0.53 mm.
0.36 mm, and 0.54 mm. In this case study, the nine faulty bearing conditions are included in the dataset, as shown in Table 4.1. The vibration signals have been collected under four load scenarios (0, 1, 2, and 3 hp) at the sampling frequency of 12 kHz. For each bearing condition, 400 samples with 1200 data points have been collected from the four load conditions. Fig. 4.3 plots the vibration signals of the nine conditions from one sensor.

The dimension of each data sample is $3 \times 1200$. There are in total 3600 samples in the dataset with 400 samples for each bearing condition. The hyper parameters of the convolution layers and the max-pooling layers are selected as listed in Table 4.2 and Table 4.3. In this work, random subsampling with validation is used. The dataset is split randomly into subsets of training, validation and testing. The ratio of each subset is defined as 70%, 15%, and 15%, which are commonly used. The training dataset is used to train the model. The validation data is used to stop the training when the error rate decreases slightly or even increases, to prevent overfitting. In the experiment, the training is first run for a comparatively long time. Then, based on the error curve of validation dataset, an appropriate epoch is chosen and the corresponding model is selected. The testing dataset is then used to test the fault diagnosis performance of the trained model. Ten trials are carried out and the average accuracy and the variation are calculated to evaluate the performance of the proposed method.

**Results and discussion**

The parameters of the model are updated using mini-batch stochastic gradient descent with a batch size of 50. Different learning rates are tested from 0.005 to 0.05 with an interval of 0.005. The results show that 0.015 achieves the best testing accuracy and the best convergence. The training and testing result of one trial is shown in Fig.4.4. The training process converges comparatively fast within 25 epochs. The testing accuracy is satisfactory, with only one sample misclassified. The average testing accuracy of the ten trails is 99.41% with a standard deviation of 0.37%. Further comparison and analysis are made under the following scenarios.

1. Fused data from multiple sensors vs. data from one sensor.

The training and testing accuracies of the proposed model with multiple sen-
Figure 4.4: Experimental result of bearing dataset. (a) Convergence curve of the training process. (b) Condition classification confusion matrix.
Table 4.1: Conditions of the bearing in the dataset

<table>
<thead>
<tr>
<th>Condition Type</th>
<th>Defect severity (mm)</th>
<th>Class label</th>
<th>Sample number</th>
</tr>
</thead>
<tbody>
<tr>
<td>BD</td>
<td>0.18</td>
<td>1</td>
<td>400</td>
</tr>
<tr>
<td>BD</td>
<td>0.36</td>
<td>2</td>
<td>400</td>
</tr>
<tr>
<td>BD</td>
<td>0.54</td>
<td>3</td>
<td>400</td>
</tr>
<tr>
<td>OR</td>
<td>0.18</td>
<td>4</td>
<td>400</td>
</tr>
<tr>
<td>OR</td>
<td>0.36</td>
<td>5</td>
<td>400</td>
</tr>
<tr>
<td>OR</td>
<td>0.54</td>
<td>6</td>
<td>400</td>
</tr>
<tr>
<td>IR</td>
<td>0.18</td>
<td>7</td>
<td>400</td>
</tr>
<tr>
<td>IR</td>
<td>0.36</td>
<td>8</td>
<td>400</td>
</tr>
<tr>
<td>IR</td>
<td>0.54</td>
<td>9</td>
<td>400</td>
</tr>
</tbody>
</table>

[BD denotes ball defect; OR denotes outer race defect; IR denotes inner race defect.]

Table 4.2: Hyper parameters of the convolution layers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>First Convolution Layer</th>
<th>Second Convolution Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of filters</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>Size of filter</td>
<td>[3, 17, 1]</td>
<td>[1, 8, 64]</td>
</tr>
<tr>
<td>Stride</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Dropout ratio</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 4.3: Hyper parameters of the max-pooling layers

<table>
<thead>
<tr>
<th>Parameter</th>
<th>First Convolution Layer</th>
<th>Second Convolution Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pooling size</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Stride</td>
<td>8</td>
<td>4</td>
</tr>
</tbody>
</table>

Sensors are compared with using signal from only one sensor, as given in Table 4.4. Ten trials are carried out to diagnosis the bearing conditions. The average testing accuracy is 99.41% by using signals from multiple sensors, which is higher than 98.35% by using the signal from only one accelerometer. Also, the standard de-
Table 4.4: Diagnosis accuracy of bearing data using multiple sensors and one sensor

<table>
<thead>
<tr>
<th>Tail #</th>
<th>Training Accuracy (%)</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multiple sensors</td>
<td>One sensor</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>99.37</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>99.25</td>
</tr>
<tr>
<td>3</td>
<td>99.96</td>
<td>99.21</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>99.33</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>98.73</td>
</tr>
<tr>
<td>6</td>
<td>99.96</td>
<td>99.56</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>99.48</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>98.97</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>98.77</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>98.97</td>
</tr>
<tr>
<td>Average</td>
<td><strong>99.99</strong></td>
<td><strong>99.16</strong></td>
</tr>
</tbody>
</table>

Standard Deviation 0.02 0.29 0.37 1.16

Variations of both training and testing accuracy of using the proposed approach are much lower than those using only one sensor, which shows a more reliable performance. The result shows that the proposed approach achieves greater and more robust diagnosis accuracy with the fusion of signals at the data level.

(2) Proposed approach vs. SVM and kNN based on hand-crafted statistical features.

As in the traditional data-driven fault diagnosis approaches, manual feature extraction is conducted first. Statistical features in the time and frequency domains used in [82][90] are calculated and used in this case study for the fault diagnosis using SVM and kNN. Table 4.5 shows the ten features in the time domain and five features in the frequency domain that are selected. Features are calculated using vibration signals from all the three sensors. Two SVM classifiers with linear kernel and quadratic kernel are used as well as kNN and weighted kNN. Table 4.6 displays the diagnosis results using SVM and kNN compared with the proposed approach. Only the classification accuracies of the bearing with a ball defect of 0.36mm and 0.54mm through the proposed method are slightly lower than using quadratic SVM. The proposed method performs better in other conditions and ob-
Table 4.5: Features selected in the time and frequency domains

<table>
<thead>
<tr>
<th>Domain</th>
<th>Feature</th>
</tr>
</thead>
</table>
| Time   | Absolute mean: \( \frac{1}{n} \sum_{i=1}^{n} |x_i| \)  
Variance: \( \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2 \)  
Crest: \( \max(|x_i|) \)  
Clearance factor: \( \max(|x_i|) / \left( \frac{1}{n} \sum_{i=1}^{n} \sqrt{x_i^2} \right)^2 \)  
Kurtosis: \( \frac{1}{n} \sum_{i=1}^{n} x_i^4 \)  
Crest factor: \( \max(|x_i|) / \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \)  
Root mean square: \( \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} / n \)  
Pulse factor: \( \max(|x_i|) / (\frac{1}{n} \sum_{i=1}^{n} |x_i|) \)  
Skewness: \( \frac{1}{n} \sum_{i=1}^{n} x_i^3 \)  
Shape factor: \( \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} / \left( \frac{1}{n} \sum_{i=1}^{n} |x_i| \right) \)  
Average frequency: \( \frac{\left( \sum_{i=1}^{n} \omega_i X_i \right)}{\sum_{i=1}^{n} X_i} \)  
Crest: \( \max(|X_i|) \)  
Frequency | Kurtosis: \( \frac{1}{n} \sum_{i=1}^{n} X_i^4 \)  
Mean energy: \( \frac{1}{n} \sum_{i=1}^{n} X_i \)  
Variance: \( \frac{1}{n} \sum_{i=1}^{n} (X_i - \bar{X})^2 \) |

contains the highest overall testing accuracy of 99.44%. It achieves end to end learning from raw data with no hand-crafted features.

### 4.4.2 Gearbox Fault Diagnosis

**Experimental setup and data description**

In this case study, the gearbox condition dataset is collected from the conveyor subsystem of an industrial fish processing machine as shown in Fig. 4.5 (a). The objective is to diagnose the different type of faults of the gearboxes. A SEW-Eurodrive R57DT80N4ES1S motor is used as the driving source of the conveyor system connected with a SEW-Eurodrive R57 gearbox. Four gearbox conditions
Figure 4.5: Experimental setup for gearbox fault diagnosis. (a) Conveyor system of a fish processing machine; (b) Three faulty gearboxes; (c) Accelerometers mounted on the gearbox; (d) National Instruments PXIe DAQ system.
Table 4.6: Comparison of bearing fault diagnosis results using different approaches

<table>
<thead>
<tr>
<th>Fault condition</th>
<th>Proposed method</th>
<th>Linear SVM</th>
<th>Quadratic SVM</th>
<th>kNN</th>
<th>Weighted kNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>BD 0.18</td>
<td>100</td>
<td>99.25</td>
<td>99.5</td>
<td>99</td>
<td>99.25</td>
</tr>
<tr>
<td>BD 0.36</td>
<td>96.67</td>
<td>94.25</td>
<td><strong>97.25</strong></td>
<td>93.75</td>
<td>92</td>
</tr>
<tr>
<td>BD 0.54</td>
<td>98.33</td>
<td>97.75</td>
<td><strong>98.75</strong></td>
<td>96.25</td>
<td>96</td>
</tr>
<tr>
<td>OR 0.18</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>OR 0.36</td>
<td>100</td>
<td>99</td>
<td>99.75</td>
<td>99.75</td>
<td>99.75</td>
</tr>
<tr>
<td>OR 0.54</td>
<td>100</td>
<td>100</td>
<td>99.75</td>
<td>99.75</td>
<td>99.75</td>
</tr>
<tr>
<td>IR 0.18</td>
<td>100</td>
<td>100</td>
<td>98.75</td>
<td>98.25</td>
<td>98.25</td>
</tr>
<tr>
<td>IR 0.36</td>
<td>100</td>
<td>99.75</td>
<td>99.5</td>
<td>98.75</td>
<td>98.25</td>
</tr>
<tr>
<td>IR 0.54</td>
<td>100</td>
<td>99.5</td>
<td>99.5</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Overall</td>
<td><strong>99.44</strong></td>
<td>98.83</td>
<td>99.36</td>
<td>98.61</td>
<td>98.36</td>
</tr>
</tbody>
</table>

[BD denotes ball defect; OR denotes outer race defect; IR denotes inner race defect.]

are tested including the normal condition and three faulty conditions (damaged gear (DG), damaged bearing (DB) and misaligned output shaft (MOS) as shown in Fig. 4.5 (b)). Two accelerometers (KISTLER 8702B25 and KISTLER 8704B100) are mounted on the gearboxes in vertical and horizontal directions as shown in Fig. 4.5 (c). The vibration signals are acquired by National Instruments PXIe DAQ system (shown in Fig. 4.5 (d)) with sampling frequency of 5 kHz. In total 6000 samples are collected where each gearbox provides 1500 samples with a length of 1000 data points.

Fig. 4.6 shows the plots of the vibration signals of each condition from one accelerometer. Since two sensors are used, the dimension of each sample is $2 \times 1000$. The same random subsampling with validation method is used here. Seventy percent of the samples are used for training, fifteen percent for validation, and fifteen percent for testing. The details of the dataset are shown in Table 4.7.

Results and discussion

A similar CNN structure as in the first case study is used. The filter size of the first convolution layer is adjusted to $[2,17,1]$ due to the change in the data dimension.
Figure 4.6: Vibration signals of gearboxes with different conditions from one sensor. (a) Normal condition. (b) Damaged gear. (c) Damaged bearing. (d) Misaligned output shaft.

Table 4.7: Gearbox dataset details

<table>
<thead>
<tr>
<th>Condition type</th>
<th>Class label</th>
<th>Training samples</th>
<th>Validation samples</th>
<th>Testing samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>1</td>
<td>1050</td>
<td>225</td>
<td>225</td>
</tr>
<tr>
<td>DG</td>
<td>2</td>
<td>1050</td>
<td>225</td>
<td>225</td>
</tr>
<tr>
<td>DB</td>
<td>3</td>
<td>1050</td>
<td>225</td>
<td>225</td>
</tr>
<tr>
<td>MOS</td>
<td>4</td>
<td>1050</td>
<td>225</td>
<td>225</td>
</tr>
</tbody>
</table>

[DG denotes damaged gear; DB denotes damaged bearing; MOS denotes misaligned output shaft.]

In the training stage, 1050 samples of each condition are used to train the model. Parameters are updated using mini-batch stochastic gradient descent with a batch size of 32.
Table 4.8: Diagnosis accuracy of gearbox data using multiple sensors and one sensor

<table>
<thead>
<tr>
<th>Tail #</th>
<th>Training Accuracy (%)</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multiple sensors</td>
<td>One sensor</td>
</tr>
<tr>
<td>1</td>
<td>99.95</td>
<td>98.29</td>
</tr>
<tr>
<td>2</td>
<td>99.98</td>
<td>98.07</td>
</tr>
<tr>
<td>3</td>
<td>99.81</td>
<td>98.07</td>
</tr>
<tr>
<td>4</td>
<td>99.95</td>
<td>98.52</td>
</tr>
<tr>
<td>5</td>
<td>99.93</td>
<td>98.71</td>
</tr>
<tr>
<td>6</td>
<td>99.98</td>
<td>97.31</td>
</tr>
<tr>
<td>7</td>
<td>99.98</td>
<td>98.52</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>98.62</td>
</tr>
<tr>
<td>9</td>
<td>99.88</td>
<td>98.43</td>
</tr>
<tr>
<td>10</td>
<td>99.93</td>
<td>96.60</td>
</tr>
<tr>
<td>Average</td>
<td>99.94</td>
<td>98.18</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.06</td>
<td>0.69</td>
</tr>
</tbody>
</table>

size of 100. Fig.4.7 shows the training and testing results of one trial. Fig. 4.7(a) plots the convergence curve of the training process of one trial. The proposed approach converges fast within 20 epochs. Fig. 4.7(b) shows the confusion matrix of the test dataset. The classification performance is outstanding, with only one sample on the condition of the damaged bearing being misclassified to the condition of the damaged gear. The average testing accuracy of the ten trials is 99.83% with a standard deviation of 0.13%.

To further evaluate the effectiveness of the proposed method, comparison and analysis are made under the following two scenarios.

1. Fused data from multiple sensors vs. data with one sensor.

Table 4.8 lists the training and testing accuracies of the proposed model with multiple sensors, compared with using the signal from only one sensor. Ten trials are carried out to diagnose the gearboxes with four conditions. The average testing accuracy by using signals from multiple sensors is 99.83%, which is higher than 97.58% by using the signal from one accelerometer. Result shows that the proposed approach achieves higher diagnosis accuracy with the fusion of signals at the data level. Also, similar to case study one, the standard deviation of the accuracy using
Figure 4.7: Experimental result of gearbox dataset. (a) Convergence curve of the training process. (b) Condition classification confusion matrix.
Table 4.9: Comparison of gearbox fault diagnosis results using different approaches

<table>
<thead>
<tr>
<th>Fault condition</th>
<th>Proposed method</th>
<th>Linear SVM</th>
<th>Quadratic SVM</th>
<th>kNN</th>
<th>Weighted kNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>100</td>
<td>100</td>
<td>99.56</td>
<td>100</td>
<td>99.89</td>
</tr>
<tr>
<td>DB</td>
<td>100</td>
<td>96.47</td>
<td>97.00</td>
<td>94.20</td>
<td>96.40</td>
</tr>
<tr>
<td>DG</td>
<td>99.56</td>
<td>94.53</td>
<td>95.27</td>
<td>92.93</td>
<td>94.47</td>
</tr>
<tr>
<td>MOS</td>
<td>100</td>
<td>97.20</td>
<td>97.80</td>
<td>95.40</td>
<td>96.47</td>
</tr>
<tr>
<td>Overall</td>
<td>99.89</td>
<td>97.05</td>
<td>97.80</td>
<td>95.40</td>
<td>96.47</td>
</tr>
</tbody>
</table>

[DG denotes damaged gear; DB denotes damaged bearing; MOS denotes misaligned output shaft.]

multiple sensors is lower, which indicates that the performance of the proposed method is more reliable than when using data from only one sensor.

(2) Proposed approach vs. SVM and kNN based on hand-crafted statistical features.

The same statistical features in the time and frequency domains are used in this case, as given in Table 4.5. Features are calculated from vibration signals from both sensors. The linear SVM, quadratic SVM, kNN, and weighted kNN, are compared here. Table 4.9 shows the diagnosis results using SVM and kNN compared with the proposed approach. Classification accuracies of the gearbox with normal condition are all close to 100% for all the methods. For other conditions, the proposed method achieves much better accuracy than all the other approaches. The overall performance of the proposed approach is the best.

In summary, a CNN-based approach with multiple sensor fusion was proposed for fault diagnosis of rotating machinery. Sensor fusion was achieved at the data level to increase the diagnosis accuracy and reliability by integrating the raw signals to form the input of the CNN-based model. Representative features were learned directly from raw signals by training the CNN model where no hand-crafted features were needed. Mini-batch stochastic gradient descent and dropout were utilized in the training process to increase efficiency and to prevent overfitting when the size of available data was small. Experimental studies on both roller
bearings and gearboxes verified the diagnosis performance of the proposed approach for fault diagnosis. The comparison between the proposed method and the traditional approaches showed that the proposed CNN-based method could achieve higher and more reliable diagnosis performance. Besides, the end-to-end feature learning capability of the proposed approach would enable its wide application in fault diagnosis of different types of machinery and various faults even when there was limited prior knowledge and limited representative hand-crafted features.
Chapter 5

RUL Prediction using Hierarchical Deep Neural Network

5.1 Introduction

Supporting the critical decision-making processes of MHM such as maintenance scheduling, machine health prognostics are of great importance to engineered systems composed of multiple components [91]. RUL prediction is conventionally the key task in machine health prognostics. With an accurate prediction of RUL, appropriate replacement or maintenance actions can be taken to ensure the reliability and safety of the running system. The approaches of RUL prediction can be classified into three categories: model-based methods, data-driven methods, and hybrid methods. Model-based methods predict the RUL by using a physical model that captures the possible damage progression processes. Due to the increased complexity of engineered systems, it is almost impossible to understand the physics-of-failures of all the components or subsystems. On the other hand, with significant advances in sensors, communication, data storage and processing technologies, the data-driven approaches have become more popular and more widely used. Data-driven approaches generally include sensory data acquisition, feature extraction,
pattern recognition, and regression. Based on the characteristics of the machinery, various sensors, e.g., accelerometer, pressure sensor, thermometer, and camera can be used to acquire the condition monitoring data.

From the point of view of system design, the actual lifetime of a system or subsystems will present the optimal cost effectiveness if it is equal to the designed or required life time of the system or subsystems. Shorter lifetime can lead to unsatisfactory system performance and longer lifetime can potentially increase the cost if more appropriate alternative design is available. Also, inappropriate design or assembly can reduce the lifetime of related components. Therefore, incorporating the lifetime assessment into the evaluation of the system design can facilitate the design optimization by identifying components or subsystems that may need improvement.

The main steps of most of the existing approaches of RUL prediction include feature extraction and regression modeling. For instance, in the prognostics of machines based on vibration, features such as RMS and kurtosis are first calculated from the vibration measurements. Then a regression model is fitted using the features. The existing approaches usually build RUL prediction models based on the entire degradation process. Fig. 5.1 shows a typical degradation process of a bearing. Fig. 5.1(a) is the plot of the vibration signal of the run-to-failure process. Fig. 5.1(b) shows the RMS of the vibration signal. The degradation level often stays low in the beginning, starts to increase at a small rate after approximately half of the lifetime, and increases dramatically before the system fails.

However, fitting the whole degradation process with a single regression model can be difficult due to the significant changes that occur at different stages. Moreover, training data of the whole degradation process is usually very limited and has different lengths, which makes it even more challenging to determine a precise model for the entire degradation process. Therefore, in this dissertation, a hierarchical DNN-based RUL prediction (DNNRULP) method is proposed to assess the RUL. The DNNRULP method contains three main parts: a DNN-based health stage classifier (DNNHSC), ANN-based RUL predictor (ANNRULP) for each health stage, and a smoothing operator. The degradation process is first segmented into \( n \) health stages based on its age in the entire degradation. A DNNHSC is built and trained by the training samples with labels of the \( n \) health stages. After
training, when the online monitoring signal arrives, the DNNHSC can output the probabilities of each health stage to which the current signal belongs. On the other hand, $n$ ANNRULPs are trained in each stage using features calculated from the raw signal in the training data. When the online signal arrives, the features are calculated first and inputted into the $n$ ANNRULPs to get $n$ RUL values. Finally, a smoothing operator is applied on the probabilities outputted from the DNNHSC and the RUL values outputted from the ANNRULPs to get the predicted RUL re-

**Figure 5.1:** Degradation process of a bearing. (a) Vibration signal of the run-to-failure process. (b) RMS of the vibration signal.
The flowchart of the proposed method is shown in Fig. 5.2. Rotating machinery is studied in this research as an illustration because of its significance and wide usage. With appropriate segmentation of health stages in the DNNHSC part and the utilization of features in the ANNRULP part, the proposed RUL prediction scheme can be applied to broad categories of components and systems.

5.2 Health Condition Classification using DNN

In this study, the degradation process is divided into five stages \( \{S^{(i)}\}_{i=1}^{5} \) as shown in Fig. 5.3. The intervals are selected based on the trend of the degradation of the training data which is the available \( l \) historical run-to-failure data \( \{H^{(i)}\}_{i=1}^{l} \). Each historical run-to-failure data \( H^{(i)} \) contains \( k \) time series items \( X^{(i)} \in \{X_1^{(i)}, X_2^{(i)}, X_3^{(i)}\} \) at each time cycle from the beginning to the end of the test; \( i \) indicates the \( i \)-th run-to-failure data item and \( j \in \{1,2,...,k\} \). Each time series contains \( m \) data point which is usually the same for all the time series items, as determined by the sampling rate and the sampling duration. Then, a DNN model is initialized and trained using the training data. Fig. 5.4 shows the structure of the DNN used in
the DNNHSC. The SDA-based feature learning procedure shown in Section 3.3 is adopted here.

The flowchart of the health stage classification is shown in Fig. 5.5.

The main steps are the following:

Step 1: Data pre-processing.

The condition monitoring data through various sensors based on the characteristics of the monitored system is collected. In this work, vibration signals are used for rotating machinery. The spectrum of the time signal is obtained by applying fast Fourier transform. The coefficients are then used as the input to the DNN.

Step 2: Initialization of the DNN.

A DNN with $N$ hidden layers is initialized with random parameters. New layers can be added until the model starts to overfit the training data as suggested by [73]. The number of nodes in the first layer can be set equal to or slightly higher than the dimension of the input data. The number of nodes of the following layers usually decrease gradually.

Step 3: Training of the DNN.

First, a denoising autoencoder is applied using the method described in section 3.3.1 for improved initialization of the parameters of the network. Layer wise
training is carried out before the layers are stacked. Then, supervised training is conducted to further update the parameters with all the training data with labels of health stage.

Step 4: Health stage classification with the trained DNN. With the trained DNN, health stage classification can be carried out on test data. The DNN outputs the five probabilities that an input signal belongs to each of the five health stages.

5.3 ANN-based RUL Predictor and Smoothing Operator

ANN with one or two hidden layers has been used by researchers for RUL prediction by capturing the nonlinearity in the relation between the condition monitoring signal and the progress in degradation. Instead of modeling the entire degradation process by one ANN model, in this dissertation, several ANN models are built based on different health stages of the system. The overall degradation process is
segmented into five health stages and the samples in each stage form the training samples for each of the ANN-based RUL predictor as shown in Fig. 5.6.

An ANN structure similar to the one in [37] and [38] is adopted in this dissertation to build each RUL predictor. The ANN model contains an input layer, two hidden layers, and an output layer as shown in Fig. 5.7.
Figure 5.7: ANN model for RUL prediction.

The input of the ANN model includes $t_i$, $t_{i-1}$, $\text{RMS}_i$, and $\text{RMS}_{i-1}$. Here, $t_i$ and $t_{i-1}$ are the age values of a component at the inspection point $i$, and the previous inspection point $i-1$. $\text{RMS}_i$ is the RMS value of the signal at the inspection point $i$, $\text{RMS}_{i-1}$ is the RMS value of the signal at the previous inspection point $i-1$. The output of the ANN model is $LP_i$, the life percentage of the component at the inspection point $i$. It is defined as the life time until the $i$-th inspection point over the duration of the entire life, as a percentage.

The ANN model considers the age values and the measurement values of the current and the previous inspection points to build an estimation model for RUL. Only the data at the two inspection time points are used instead of incorporating data at more past inspection points since an ANN can have better generalization capability with fewer input nodes [38].

After obtaining the predicted RUL using the predictors, a smoothing operator is applied to get the final RUL prediction through the following equation:

$$RUL(X) = \sum_{i=1}^{n} P(S_i|X) \cdot R_i(X) \quad (5.1)$$

Here $X$ is the current measurement of the monitored system; $i = 1, 2, ..., n$ rep-
resent the \( n \) different health stages; \( P(S_i|X) \) is the probability that \( X \) is assigned to class \( i \) using the DNN health stage classifier; and \( R_i(X) \) is the predicted RUL of the current measurement by the \( i \)th ANN predictor.

5.4 Experimental Study

This section presents a case study of RUL prediction using the proposed DNNRULP method. RUL prediction of bearings is considered here. The typical degradation process of a bearing is shown in Fig. 5.1.

5.4.1 Data Description

In the present experimental study, the dataset of IEEE 2012 PHM Data Challenge is used [92]. The experiment platform, PRONOSTIA, is designed to test and validate the methods of bearing fault diagnosis and prognosis. It provides real experimental data of degradation of bearings in only a few hours. Compared with other bearing testbeds, the PRONOSTIA provides degradation data without any initial initiated defects on the bearings. PRONOSTIA includes three main parts: a rotating part, a degradation generation part and a measurement part. The detailed components of the platform are shown in Fig. 5.8.

Run-to-failure experiments are performed on the PRONOSTIA platform. Vibration and temperature signals are collected from 17 run-to-failure bearing tests under three different operating conditions as listed in Table 5.1. In this study, only the vibration signal in the horizontal direction is used for the RUL prediction. The vibration signal is sampled at the sampling rate 25.6kHz at each 10s for a duration of 0.1s as shown in Fig. 5.9. Therefore, each sample contains 2560 points. The training dataset is formed by 16 run-to-failure vibration signals and the testing dataset is formed by one run-to-failure signal.

5.4.2 RUL Prediction Using DNNRULP

First, the DNNHSC is trained using the training samples. FFT is performed on each vibration signal from the bearing test. Each vibration signal contains 2560 data points. Thus, there are 1280 data points after FFT. A five layer DNN with one input layer, three hidden layers, and one output layer is initialized by random
Figure 5.8: PRONOSTIA bearing test setup.

Table 5.1: The operation conditions of the experiment

<table>
<thead>
<tr>
<th>Operation Condition</th>
<th>Motor Speed (rpm)</th>
<th>Radial Force (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1800</td>
<td>4000</td>
</tr>
<tr>
<td>2</td>
<td>1650</td>
<td>4200</td>
</tr>
<tr>
<td>3</td>
<td>1500</td>
<td>5000</td>
</tr>
</tbody>
</table>

parameters. The number of nodes of the hidden layers is selected as 800, 400, and 100. The samples from the 16 run-to-failure tests are used to train the DNN model. SDA is first performed to get a better initialization of the parameters. Then supervised fine-tuning is conducted using the training data with labels. Second, the ANNRULPs are trained separately by samples of the corresponding stage. The RMS value is first calculated for each vibration sample. Then, five ANNs with the structure shown in Fig. 5.7 are initialized with random parameters. The age values and the RMS values of the current inspection point and the previous point form the input of the ANN. The expected output is the life percentage of the current inspection point. The five ANNs are trained using the training dataset.
With the trained DNNHSC and five ANNRULPs, the RUL prediction is performed on the testing data, which is the vibration signal from one run-to-failure experiment. The testing sample has 434 data points. Ten trials are conducted in the experiment. Table 5.2 shows the training and testing results of the DNNHSC. Using Equation 5.1, the predicted life percentage of the testing data is shown in Fig. 5.10.

The average prediction error is used as the measure for the prediction performance using the following equation:

$$\bar{e} = \frac{1}{n} \cdot \sum_{i=1}^{n} |LP_i - \hat{LP}_i|$$  \hspace{1cm} (5.2)

where $\bar{e}$ denotes the average prediction error; $n$ is the number of inspection points of the testing sample; and $LP_i$ is the actual life percentage at inspection point $i$; $\hat{LP}_i$ is the predicted life percentage at inspection point $i$. The RUL prediction accuracy in later stage of the running system is usually more important as the system is close to possible failure when maintenance or replacement needs to be arranged. Thus, in this dissertation, the prediction performance of the last 10% and the last 5% of the total inspection points is calculated to evaluate the effectiveness of the proposed
Table 5.2: Training and testing result of the DNNHSC

<table>
<thead>
<tr>
<th>Trial Number</th>
<th>Training Accuracy (%)</th>
<th>Validation Accuracy (%)</th>
<th>Testing Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80.65</td>
<td>78.40</td>
<td>78.11</td>
</tr>
<tr>
<td>2</td>
<td>80.82</td>
<td>79.21</td>
<td>70.51</td>
</tr>
<tr>
<td>3</td>
<td>81.15</td>
<td>78.03</td>
<td>73.27</td>
</tr>
<tr>
<td>4</td>
<td>80.69</td>
<td>77.54</td>
<td>76.50</td>
</tr>
<tr>
<td>5</td>
<td>80.97</td>
<td>78.16</td>
<td>77.42</td>
</tr>
<tr>
<td>6</td>
<td>80.72</td>
<td>78.53</td>
<td>75.35</td>
</tr>
<tr>
<td>7</td>
<td>81.02</td>
<td>77.07</td>
<td>70.74</td>
</tr>
<tr>
<td>8</td>
<td>80.98</td>
<td>79.64</td>
<td>75.12</td>
</tr>
<tr>
<td>9</td>
<td>81.20</td>
<td>77.26</td>
<td>73.27</td>
</tr>
<tr>
<td>10</td>
<td>80.70</td>
<td>77.32</td>
<td>74.65</td>
</tr>
<tr>
<td>Average</td>
<td>80.89</td>
<td>78.12</td>
<td>74.49</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.19</td>
<td>0.81</td>
<td>2.45</td>
</tr>
</tbody>
</table>

Figure 5.10: RUL prediction result using the HDNNRULP.
Table 5.3: RUL prediction error using the HDNNRULP

<table>
<thead>
<tr>
<th>Trial Number</th>
<th>Eave (%)</th>
<th>Eave10 (%)</th>
<th>Eave5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.12</td>
<td>9.23</td>
<td>9.69</td>
</tr>
<tr>
<td>2</td>
<td>7.45</td>
<td>7.66</td>
<td>8.37</td>
</tr>
<tr>
<td>3</td>
<td>6.96</td>
<td>7.77</td>
<td>9.54</td>
</tr>
<tr>
<td>4</td>
<td>6.63</td>
<td>7.49</td>
<td>8.00</td>
</tr>
<tr>
<td>5</td>
<td>8.89</td>
<td>9.13</td>
<td>9.59</td>
</tr>
<tr>
<td>6</td>
<td>5.79</td>
<td>6.78</td>
<td>7.51</td>
</tr>
<tr>
<td>7</td>
<td>7.30</td>
<td>5.14</td>
<td>6.83</td>
</tr>
<tr>
<td>8</td>
<td>6.67</td>
<td>8.64</td>
<td>9.15</td>
</tr>
<tr>
<td>9</td>
<td>8.04</td>
<td>12.12</td>
<td>10.70</td>
</tr>
<tr>
<td>10</td>
<td>7.74</td>
<td>7.30</td>
<td>9.37</td>
</tr>
<tr>
<td>Average</td>
<td>7.26</td>
<td>8.13</td>
<td>8.88</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>0.85</td>
<td>1.84</td>
<td>1.17</td>
</tr>
</tbody>
</table>

method. The average prediction error of the last 10% and 5% of the total inspection points is calculated using Equation 5.2 by setting \( n \) equal to be 43 and 22. Table 5.3 lists the average prediction error of the total inspection points and that of the last 10% and 5%.

The average \( \bar{e} \) of the ten trails is 7.26%, with a standard deviation of 0.85%. The overall prediction performance is satisfactory. The prediction error of the last 10% and the last 5% of the entire degradation is 8.13% and 8.88%, respectively.

The RUL prediction result using the proposed prediction method, which uses hierarchical DNN, is compared with that using a single ANN predictor trained on using the whole degradation without any stage classification. Fig. 5.11 plots the prediction results using the two different approaches.

The proposed method has better overall performance than that with only one single predictor. Only in the very early stage, before 80 time circles, the single ANN predictor achieves a smaller error. However, afterwards, the single ANN cannot model the degradation satisfactorily and has significantly larger error. The RUL prediction by the proposed method achieves high accuracy.

In summary, a hierarchical DNN-based method has been proposed to predict the RUL of the components and subsystems of a monitored mechatronic system. Instead of modeling the entire degradation process by one model, the proposed
method first segments the overall degradation process into several health stages based on the degradation characteristics. A DNN-based classifier is trained to achieve the classification of different health stages. For each health stage, one ANN model is established to model the relation between the extracted features and the RUL in that health stage. Finally, a smoothing operation function is applied on the output of the DNNHSC and the output of the n ANNRLPs to obtain the RUL prediction. An experimental study of RUL prediction on bearings was conducted. The results demonstrate the effectiveness of the proposed method in RUL prediction. It achieves much higher prediction accuracy than that using only one ANN model on the entire degradation process.

Figure 5.11: Comparison of the RUL prediction results.
Chapter 6

Conclusions and Future Work

6.1 Conclusions

With enhanced requirements in reliability, flexibility, cost and intelligence of mechatronic systems, continuous and rapid design improvement of these systems is crucial. In this dissertation, new methodologies for applying MHM for the automated design optimization of mechatronic systems have been developed.

First, a closed-loop design evolution framework incorporating MHM is presented for mechatronic systems. In this approach, MHM is used for design weakness detection. It continuously determines design improvements for a mechatronic system through the stages of conceptual design, detailed design, and implementation. Specifically, a systematic approach for design weaknesses detection is developed, to guide the redesign process of an existing system by narrowing down the search space.

With the assistance of IoT and CC, the limitation of a traditional MHM approach in sensing, data transmission, data storage and data processing can be enhanced. However, it can generate massive monitoring data where more intelligent diagnostics and prognostics approaches are needed. This dissertation presented an intelligent fault diagnosis approach based on SDA and DNN. It overcomes the typical drawbacks of the traditional fault diagnosis methods: feature extraction with heavy prior knowledge, requirement of a large amount of labeled data, and model rebuilding for diagnosing new faults. Representative features can be learned auto-
matically from massive and unlabeled condition data. After fine-tuning by a few labeled data, the proposed approach can achieve precise fault diagnosis.

Moreover, the proposed approach can utilize the trained model to diagnose new conditions by further fine-tuning the DNN model with a few labeled data of the new conditions. A standard dataset of bearing faults is used to evaluate the effectiveness of the proposed approach as well as the robustness of the method to noise. It enhances the capability of MHM system for application in general and wider categories of mechatronic systems.

Sensor fusion can supply richer information for a more accurate and reliable estimation. In this dissertation, a CNN-based approach with multiple sensor fusion has been developed for fault diagnosis. Signals from multiple sensors are fused at the data level to form the input to the model. Representative features can be learned directly from raw signals by the CNN model without any hand-crafted features. Experimental studies on the fault diagnosis of a bearing and a gearbox illustrate the effectiveness of the CNN-based approach. The end-to-end feature learning capability enables its wide application in fault diagnosis of machinery where prior knowledge and handcrafted features are limited.

A hierarchical DNN-based RUL prediction approach is developed to improve the RUL prediction accuracy of the components and subsystems of the mechatronic system. Unlike modeling the entire degradation process by a single model, the degradation process is classified into several health stages and several ANNs are used to model the degradation in each stage. In the health stage classification, a DNN-based classifier is trained to achieve the health stage classification with the raw monitoring data. The degradation in each stage is then modeled by an ANN using the calculated features. A smoothing operator is applied on the health stage classification result and the RUL predictions of each stage to generate the RUL prediction. The performance of the HDNNRULP was evaluated by an experiment of bearing RUL prediction.

### 6.2 Possible Future Work

This dissertation presented a systematic approach for applying MHM for automated design optimization of a mechatronic system. The closed-loop framework
of design evolution of a mechatronic system with MHM can achieve continuous design improvement. This dissertation developed a design weakness index integrating system performance evaluation, fault diagnosis, and prognosis. The detailed and practical formulation of the system performance evaluation can be further investigated for real world applications. There is also room for research in how to interpret the deviated RUL of the components and subsystems with respect to the design quality. A case study of conceptual design improvement was conducted in Chapter 2. Further research can be done on the detailed design optimization with the determined information on the system design weaknesses.

In the DNN-based fault diagnosis approach, no hand-craft features are needed. It can learn representative features automatically from a massive quantity of unlabeled condition data. However, the features learned by the DNN from the training data are not interpreted in detail. Future research can focus on the analysis of the underlying meaning of these features.

The proposed CNN-based fault diagnosis approach, which incorporates sensor fusion at the data level achieves more accurate and reliable fault diagnosis results. The CNN model deals with multiple sensed data with the same length under the same sampling frequency. How to integrate data with different lengths could be further investigated to increase the capability of the CNN-based fault diagnosis method by fusing data having different sampling frequency.

Finally, in the hierarchical DNN-based RUL prediction, the segmentation of the health stages is based on the time of the degradation. One possible direction of research is to achieve optimized segmentation by using an AI learning method using the training data. The number of stages and the time intervals of each stage can be determined by minimizing a cost function such as the average RUL prediction error.
Bibliography


102
[71] Xue Feng, Yaodong Zhang, and James Glass. Speech feature denoising and
dereverberation via deep autoencoders for noisy reverberant speech recogni-
tion. *Proceedings of the 2014 IEEE International Conference on Acoustics,
Speech and Signal Processing (ICASSP)*, pages 1759–1763, may 2014. →

[72] Roberto DAmbrosio, Giulio Iannello, and Paolo Soda. Softmax regression
for ecoc reconstruction. In *International Conference on Image Analysis and

[73] Geoffrey E Hinton. A Practical Guide to Training Restricted Boltzmann Ma-
chines. In Grégoire Montavon, Geneviève B Orr, and Klaus-Robert Müller,

[74] K.A. Loparo. Case Western Reserve University Bearing Data Centre Website,
2012. →

[75] L J P Van Der Maaten and G E Hinton. Visualizing high-dimensional data

[76] J W Park, S H Sim, and H J Jung. Displacement Estimation Using Multimet-
1682, 2013. →

[77] B. Olofsson, J. Antonsson, H. G. Kortier, B. Bernhardsson, A. Robertsson,
and R. Johansson. Sensor fusion for robotic workspace state estimation.

composite health indices for degradation modeling and prognostic analysis.
*IEEE Transactions on Automation Science and Engineering*, 10(3):652–664,
July 2013. →

[79] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully con-
3119–3127, Dec 2015. →

[80] Zhiqiang Chen, Chuan Li, and René-vinicio Sanchez. Gearbox Fault Identi-
fication and Classification with Convolutional Neural Networks. *Shock and


