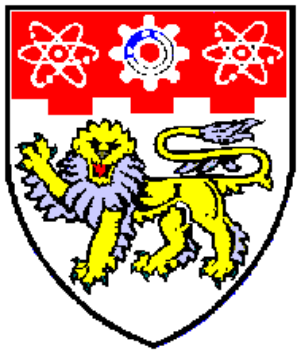
Fault Detection and Diagnosis for Chillers and AHUs of Building ACMV Systems



Li Dan

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy

January 2017

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

Date	Li Dan

Acknowledgements

Foremost, I would like to thank my supervisors, Professors Hu Guoqiang and Costas J. Spanos, for guiding me through my graduate studies. They have given me both patient guidance and thoughtful discussions to pursue my research goals in novel and interesting directions. Their specialty standard and rigorous attitudes towards research will affect the rest of my life. My gratitude towards them is beyond words.

I would like to thank Professors Xie Lihua and Sanjib Kumar Panda for being my Thesis Advisory Committee. They have given me very helpful feedback for my graduate work.

I would like to thank Zhou Yuxun for his effective discussions and selfless help. His unique views on science and technology have inspired me to explore more. He is an excellent co-researcher and even more, a good friend.

I would like to thank the lab colleagues and technician for their friendship and kindly help.

I would like to thank the SinBerBEST project for funding my graduate study.

Finally, I would like to express my special gratitude to my family for their love and support.

Abstract

Building worldwide contributes to 40% of the global energy consumption, and most of that is due to Heating, Ventilation, and Air-conditioning (HVAC) systems. A large part of this energy is wasted because of poor maintenance, inevitable degradation, and improperly controlled equipment. Therefore, it is of practical relevance and significance to study Fault Detection and Diagnosis (FDD) techniques for smart buildings aiming at saving energy and offering more comfortable dwelling environment. Researchers have been tackling building FDD task with a wide variety of techniques, such as analytical model-based, signal-based and data-driven methods. Recently the data-driven method has shown its advantage in dealing with complex systems with random penetrations. Most of the existing works tend to formulate the data-driven FDD as a pure fault types identification task. Problems such as severity levels identification, inter-dependence information incorporation, and essential features selection have long been ignored.

This dissertation addresses the aforementioned problems, and the details are summarized as follows.

First of all, the building FDD task is directly formulated as a multiple classification problem. A Discriminant Analysis-based Fault Classification (DAFC) method is driven to conduct the detection and diagnosis. Linear Discriminant Analysis (LDA) is firstly adopted to project the high dimensional data into a lower dimensional space so as to achieve optimal class separation and maximum original information

maintenance. Derived from the K-means Clustering, DAFC classification applies two criteria to make a decision. The testing data set is classified to a certain cluster if: 1), it is the closest to that cluster by Manhattan distance; 2), Manhattan distances between the testing data set and that cluster are within a certain range. By feeding the training and testing data to DAFC, fault type is diagnosed at the first stage, and the corresponding severity level is identified at the second stage. The proposed two-stage data-driven FDD strategy is validated by the experimental data collected by the ASHRAE Research Project 1043 (RP-1043). Results show that it can detect and diagnose chiller faults and the corresponding severity levels effectively.

Although the two-stage FDD strategy generates satisfactory results, it only works well when the number of included classes is small. Formulating the FDD task as a pure multiple classification problem is not effective enough when the number of included classes becomes large. Thus, a Tree-structured Fault Dependence Kernel (TFDK) method is proposed to identify fault type as well as fault severity level in a unified large margin learning framework. TFDK adopts structured labeling to incorporate the inter-class dependence information and deals with the streaming data with a corresponding on-line learning algorithm. As an improvement of traditional classification methods, it encodes the dependence information in its feature mapping and takes regularized misclassification cost as the learning objective. Similarly, following the ASHRAE Research Project 1043 (RP- 1043), TFDK is applied to solve the FDD for a 90-ton centrifugal water-cooled chiller. Experimental results show that compared to conventional classification methods, TFDK can significantly improve the FDD performance and recognize the fault severity levels with high accuracy.

Lastly, previous works have justified that buildings and their operation can greatly benefit from rich and relevant data sets. More specifically, data has been analyzed to detect and diagnose system and component failures that undermine energy efficiency.

Among the vast amount of measured information, some features are more correlated with the failures than others. However, there has been little research to date focusing on determining the types of data that can optimally support FDD. Thus, a novel optimal feature selection method, the Information Greedy Feature Filter (IGFF) method, is proposed to select essential features. On the one hand, the selection results would serve as a reference for configuring sensors in the data collection stage, particularly when the measurement resource is limited. On the other hand, with the most informative features selected by IGFF, the performance of building FDD could be improved and theoretically justified. A case study on Air Handling Unit (AHU) FDD is conducted based on the ASHRAE Research Project 1312 (RP-1312). Numerical results show that compared with several baselines, the FDD performances of conventional classification methods are greatly enhanced by IGFF.

In summary, this dissertation studies the data-driven techniques and proposes several effective strategies to solve the FDD problem for building chillers and AHUs. Compared with previous works, the proposed DAFC can identify the fault severity level at a second stage after fault types have been diagnosed. This dissertation also focuses on recognizing both fault types and the corresponding severity levels in a unified learning framework. Hence, the inter-class fault dependence information is included with tree-structured labeling by the proposed TFDK algorithm. Besides, by selecting essential subsets of variables that are more correlated to faults with the proposed IGFF algorithm, not only the FDD accuracy is improved, but also the FDD application becomes more convenient and practical.

Contents

List of Figures

List of Tables

Symbols and Acronyms

$*$	The complex conjugate symbol
$\boldsymbol{I}(.;.)$	Mutual information
$\Lambda\,(.)$	Attribute reweighing vector
$\mathcal{L}$	The Lagrangian formulation
μ	Class mean
$\otimes$	The tensor product symbol
Φ	Feature transformation
Σ	A common covariance matrix
σ	Deviation
ξ	Slack variable
$Accu$	Classification accuracy
CC	Cluster Center
$D_{KL}(.)$	The $K-L$ divergence
$H(.)$	Entropy
Neu_j	The jth neural

$O(.)$	Complexity of the problem
$P(.)$	Probability
S_b	Between-class scatter matrix
S_w	Within-class scatter matrix
$\hat{f}^{LDA}$	The LDA classification rule
$\hat{f}^{LR}$	The LR classification rule
$\hat{f}^{AdaBoost}$	The AdaBoost classification rule
$\hat{f}^{Boost}$	The boosting classification rule
$\Re^p$	A feature space with p features
$\hat{\delta}_j^L$	LDA classifier for class j
$\hat{\delta}_j^Q$	QDA classifier for class j
$\hat{f}^{Tree}$	The tree classification rule
AB	AdaBoost
AHU	Air Handling Unit
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
CAV	Constant Air Volume system
DWT	Discrete Wavelet Transform
FDD	Fault Detection and Diagnosis
HVAC	Heating, Ventilation and Air-Conditioning

KKT	The KarushKuhnTucker conditions
LDA/FDA	Linear/Fisher Discriminant Analysis
LR	Logistic Regression
mRMR	The maximum Relevance Minimum Redundancy method
NN	Neural Networks
QDA	Quadratic Discriminant Analysis
RP	Research Project
SR	The Sparse Regularization method
SVM	Support Vector Machines
VAV	Variable Air Volume system
VC	VapnikChervonenkis dimension
VFD	Variable Frequency Drive

Chapter 1

Introduction

1.1 Background and Motivation

Building energy consumption contributes to more than 40% of the total energy usage worldwide [1,2]. And almost 32% of that is due to Air-Conditioning and Mechanical Ventilation (ACMV) systems [3,4]. One the one hand, a large part of this energy is wasted because of poor maintenance, inevitable degradation, and improperly controlled equipment. On the other hand, the building energy demand will continuously rise due to the growth of population, the long-term use of buildings and the increasing demand for improved building comfort levels [5]. The Department of Energy (DOE), the International Energy Agency (IEA), International governmental Panel on Climate Change and other agencies have declared a necessity for commercial buildings to become 70-80% more energy efficient [6]. Moreover, field experience shows that 5-30% of energy savings can be achieved in buildings by applying early FDD and correcting the detected and diagnosed faults [7]. Recently, the ASHRAE Handbook has put special emphasis on automated Fault Detection and Diagnosis (FDD) for smart building systems. In particular, the new standard highlights the necessity of maintaining the whole building system in good working

conditions through FDD techniques as well as the significance of saving energy and improving occupancy comfort level and building safety level via automated FDD system [8]. Consequently, there is an increasing need for automated fault identification in smart buildings aiming at saving energy and offering more comfortable dwelling environment [9,10]. Although the building designers and managers have already offered many effective methods for regular maintenance, improving the whole building system's performance through FDD is still of great importance in terms of reducing the maintenance and repair costs [11–17].

A fault in the building means a deviation in the value of at least one characteristic variable from its normal expected behavior, including improper performances of ACMV systems, excessive building peak electrical demand, wrong system set-points and operation schedules, equipment malfunction, inaccurate sensor measurements, and so on. Research on ACMV divides faults into two categories [11]: (1) hard failures that occur abruptly and cause the system to stop suddenly, and (2) soft faults that cause performance degradation but the system still functions. Hard faults are dangerous and may cause heavy economic losses once happen. Usually, traditional Building Management System (BMS) alarms once glaring faults occur based on the system thresholds. Thus, hard faults will not be studied in this dissertation since they can be detected by inexpensive measurements and simple analysis. Soft faults, such as a slow loss of refrigerant or partial condenser fouling, are not obvious enough to be detected and diagnosed without well-designed analysis strategies. As a result, in the past decades, researchers have dedicated significant efforts to develop FDD algorithms and strategies that aim to detect soft faults [18–21].

By constantly monitoring system operations, FDD aims at detecting faults ahead of time, diagnosing their causes and providing feedback to building operators, who can make prompt reaction before severe damage is added to the system or additional economic loss occurs [22]. Hence, advantages of building FDD including prolonging

equipment life, improving the indoor comfort level, reducing the building operating costs and maintenance expenses, and so on [23]. A typical FDD-based operation and maintenance process includes the following four distinct functional processes [24]: 1), detect anomalies through monitoring performances of building systems; 2), diagnose the causes and identify the locations; 3), evaluate the severity level and make decisions; 4), make alerts or recommendations to the facility managers, and in some cases even automatically generate working orders or take corrective actions directly within the BMS. This dissertation focuses on the detection and diagnosis of typical soft faults for building chillers and Air Handling Units (AHUs).

Researchers have been tackling the building FDD task with a wide variety of techniques, such as analytical model-based, signal-based and knowledge-based methods [24–28]. Recently the data-driven method has shown its advantage in dealing with complex systems with random penetrations. To achieve an efficient data-driven FDD design, it depends on well-grained sensory data harvested from the Wireless Sensor Networks (WSNs) [29]. Usually, in commercial buildings, the empirical data is collected through sensor networks and stored in the Building Management System (BMS) [30,31]. The data records outside environmental factors, internal loads, and mechanical system working conditions [32]. To be specific, sensors are deployed to collect data periodically according to fixed protocols. Base-station gathers sensor data and transmits it to a remote data management system, which is exactly the BMS or a separate server. Experts and researchers then analyze the empirical data and give feedback to building operators if any fault is found. A typical data-driven FDD system for smart buildings is depicted in Figure 1.1, including deployed wireless sensor networks, a database management systems, and a decision support system.

Figure 1.2 shows how the FDD tool integrates with the BMS and updates its results successively. Firstly, the labeled historical data is accumulated into a fault

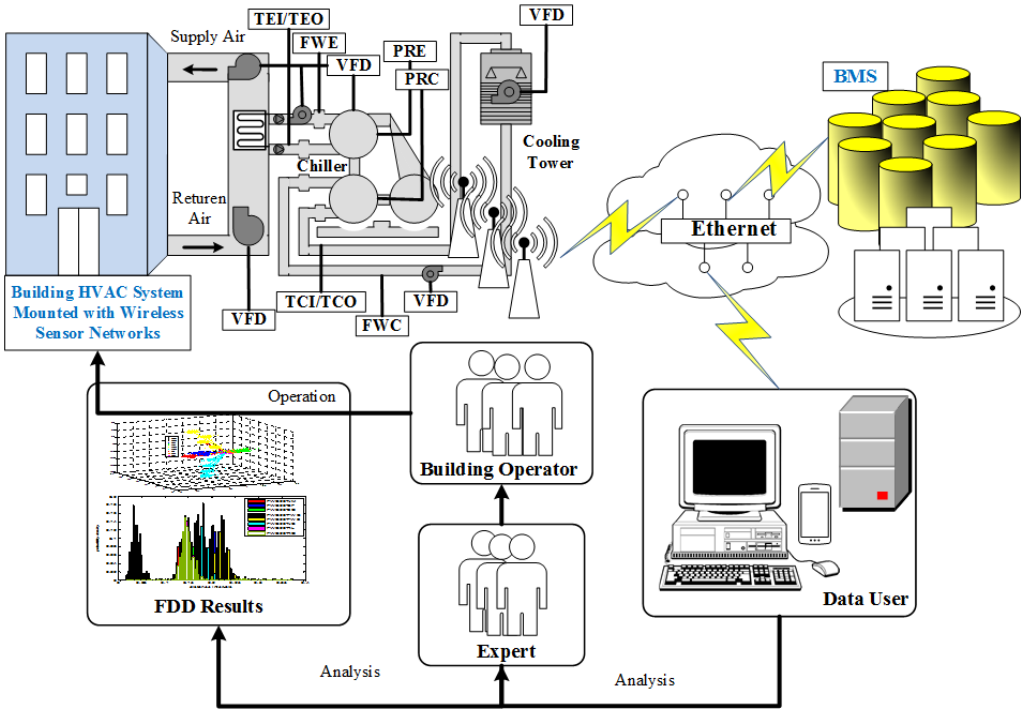


Figure 1.1: Data-driven building FDD scheme, including deployed sensor networks, a database management system, and a decision support system.

library and stored in a database as the "batch training" data set, based on which an initial data-driven FDD model is obtained (by running the proposed algorithms). Then, at each round of detection, real-time sensor measurements and monitoring data are collected as the input of a stand-alone program, which implements the classifier, to conduct FDD. Next, detection and diagnosis results are provided to building managers and operators for further inspection and possible corrections. Their feedback, i.e., another labeled instance, constitutes the "real-time training" input of the algorithm, which is designed incrementally. With this on-line training phase, the data-driven FDD model is refined and prepared for future FDD. The process goes on iteratively as mentioned above.

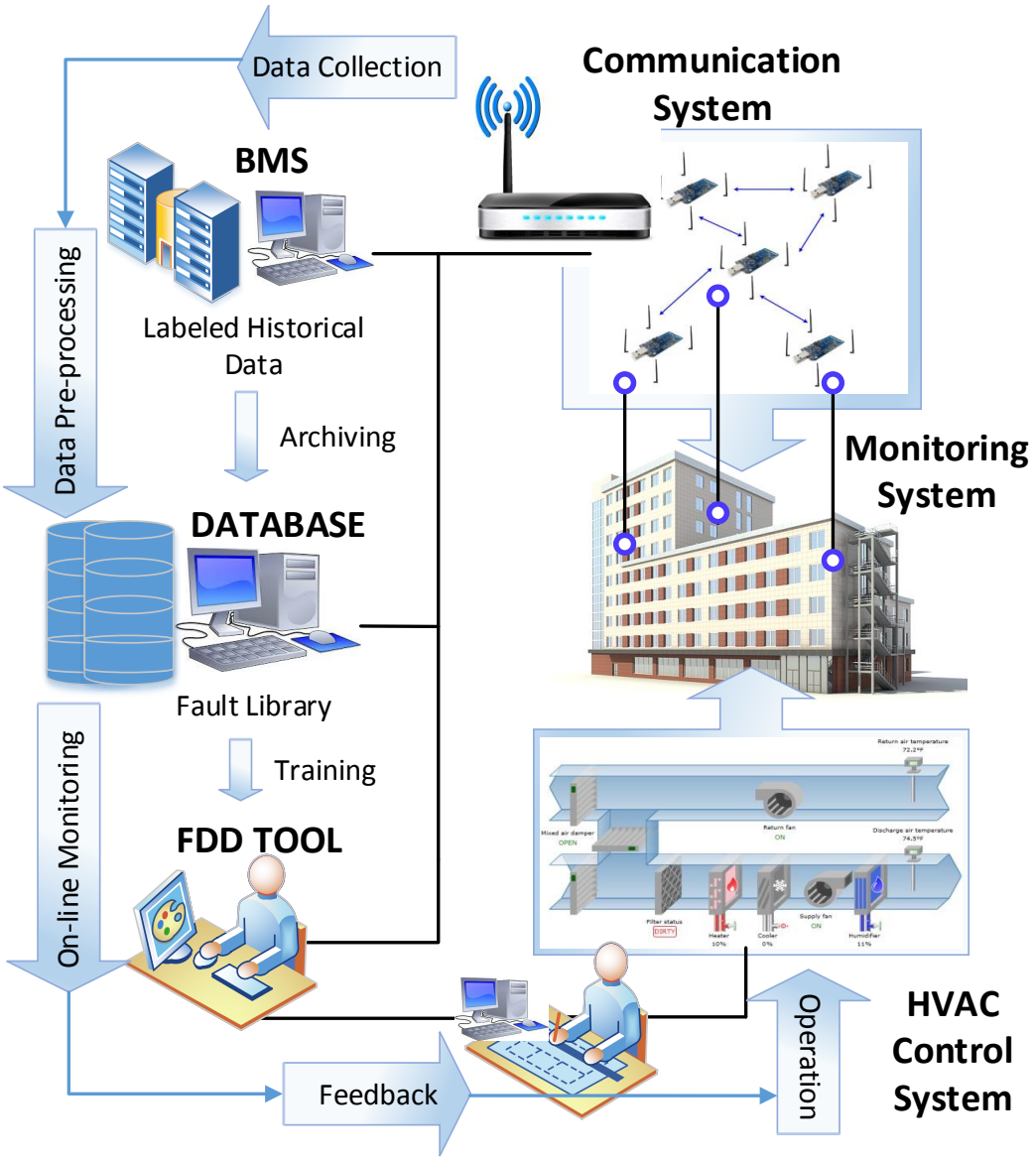


Figure 1.2: Schematic showing how to integrate the FDD tool into the BMS.

1.2 Objectives

Motivated by both the theoretical and practical importance of studying building FDD, the main focuses and objectives of this dissertation are summarized as follows.

1. **FDD with Distance-based Classifier**

   Formulate the building FDD as a pure multiple classification problem. Com-

mon fault types are detected and diagnosed at the first stage. The corresponding fault severity level is identified at the second stage. The strategy achieves relatively high FDD accuracy with low computational cost. It is also friendly to BMS integration and real-time monitoring.

2. **FDD with Tree-structured Learning Method**

   Identify both fault type and the corresponding severity level in a unified learning framework, and improve the FDD performance when more classes are included. Extract inter-class independent information from the expertise system analysis. The FDD algorithm incorporates the fault dependence information with structured labeling and describes fault severity levels in a large margin learning framework. The corresponding on-line updating procedure is also developed to monitor real-time streaming data.

3. **FDD with Feature Selection Method**

   Determine the types of data that can optimally support FDD. The proposed feature selection algorithm selects the optimal subset of features [1] by maximizing the mutual information between candidate variables and the fault labels. The selected features can not only guide future experimenters and operators to deploy sensors, but also improve the building FDD accuracy.

In summary, this dissertation focuses on studying data-driven building FDD techniques. The fist objective is achieved by projecting the high dimensional data into a lower dimensional space and then apply the proposed distance-based multiple classifier to diagnose fault types and the corresponding severity level at two separate

[1] The notion "feature" generally means a single variable in some cases or the combination of variables in others. In this dissertation, the "feature" means a single variable but the proposed feature selection method is not limited to that.

steps. Following the ASHRAE Research Project 1043 (RP-1043) [33], the proposed classifier is validated by the fault data of a 90-ton centrifugal water-cooled chiller. The second objective is to improve the FDD performance by incorporating system information with tree-structured labels instead of pure fault type classification. The inter-class fault dependence information is incorporated to identify both fault type and the corresponding severity level in a unified learning framework. At the same time, FDD performance is improved compared with the conventional classification methods. Similarly, the second FDD objective is justified with the experimental data of the ASHRAE RP-1043. By achieving the first two objectives, it has been justified that data can be analyzed to detect and diagnose system and component failures that undermine energy efficiency. Among the huge amount of information, some features are more correlated with the failures than others. To determine the features that can optimally support FDD, naive dimension reduction or empirical selection is far from enough. As a result, the third objective is to pick out prime streams and improve the FDD performance with less sensors by an efficient theoretically guaranteed algorithm. This objective is validated with the experimental data of the ASHRAE Research Project 1312 (RP-1312) [34].

1.3 Literature Review

Previous review papers have presented enough reports on the investigating of various faults for sensors, Air Handling Unit (AHU), Fan Coil Unit (FCU), chillers, boilers, heat pumps, and so on [3,24,26–28,33–42]. This dissertation will not repeat the related content. In this section, we mainly review existing FDD methods in the building application field.

1.3.1 FDD Methods Overview

FDD techniques focus on how to detect the occurrence of a failure automatically [2] and diagnose its location and cause as early as possible [21]. Over the past decade, automated FDD has been brought to building energy efficiency field, especially to the ACMV system, from the aerospace process control, manufacturing, nuclear and national defense fields [3]. Reviewed FDD methods can be roughly divided into three categories [6,39]:

1. **Analytical model-based FDD method.**

   At the early stage, FDD method merely checks sensor measurements and system outputs with predefined limits. However, the simple over-threshold checking method becomes invalid when the system becomes complex. To overcome difficulties with limit checking raised by system complexity, model-based FDD was proposed based on the sophisticated state-space modeling and system identification techniques [43–45]. However, those model-based FDD methods stay in theoretical level, and few of them can be directly inserted to the BMS to conduct real-time monitoring [46–48]. In a word, the model-based FDD can detect and diagnose faults for a few on-line measurements and system outputs, but it requires an explicit input-output model of the target system, which is hardly available for large complex building system.

2. **Signal-based FDD method.**

   Signal-based FDD method was first developed as a result of the significant improvement of signal processing techniques. The signal-based FDD method investigates the correlation between faults and system output signals, and improved performance can be achieved by adding the signal pattern of healthy

[2] As has mentioned in Section 1.1, the "fault" considered in this dissertation means a "soft fault".

status as a priori [25,27]. Usually, signal-based FDD methods mainly focus on electronic signal and vibrations of motors and rotary machines due to requirements of signal analysis techniques [49,50]. Its performance may degrade if the system works in an unknown or unbalanced condition.

3. **Knowledge-based FDD method.**
   Unlike model-based FDD method and signal-based FDD method, the knowledge-based method relies on mass historical data and is entirely model free. The knowledge-based FDD method discovers the hidden information buried in historical data that represents the information redundancy among the system's variables [35,51,52]. This kind of method pays the highest computational costs compared with other two kinds of methods. Due to those facts, the knowledge-based method is commonly referred to as data-driven method [39]. Because of the *learn-by-example* peculiarity, the performance of knowledge-based FDD relies on how reliable the training data is.

All in all, thanks to the development of sensor network techniques and machine learning methods, the data-driven FDD method has shown apparent advantages over model-based and signal-based methods recently [39]. As an alternative to the complex physical models, data-driven methods are derived only from readily available sensor data [53]. It generates satisfactory performance under the big data background. This dissertation mainly focuses on studying data-driven FDD techniques.

1.3.2 Data-driven FDD Methods with Applications to Buildings

In the literature, miscellaneous works about data-driven building FDD have been done, such as chiller fault detection and diagnosis [21,54–56], AHU sensor fault

detection and diagnosis [57,58], variable air ventilation system sensor fault detection and diagnosis [59–61], and soft fault detection in the whole ACMV system [62,63].

In the aforementioned FDD works, a wide range of statistical and machine learning techniques have been explored as data-driven methods, including Principal Component Analysis (PCA) [56,62,64], Statistical Process Control (SPC) [12,13,65], Multivariate Regression Models [66], Bayes Classifier [67–69], Neural Networks (NN) [70–72], Fisher Discriminant Analysis (FDA) [58], Gaussian Mixture Model [73], Support Vector Data Description (SVDD) [74,75], and Support Vector Machines (SVM) [76–80]. Among these approaches, PCA and SPC are unsupervised methods that do not require expert knowledge for fault labeling, but others like NN and FDA are supervised multiple classification methods that depend on the availability of labeled training data. Once the hypothesis/model is fitted from the training phase, new measurements will be tested by the classifiers and be assigned to corresponding categories (normal or faulty) automatically.

Notwithstanding existing works on data-driven FDD has shown promising results in both detection accuracy and efficiency [28,35,81], problems such as interdependence information incorporation, severity levels identification, and essential features selection have long been ignored or over-simplified.

First of all, although it is quite intuitive to build fault dependence by analyzing the connections and structures of each component of ACMV system, this prior knowledge is rarely considered in current data-driven FDD literature. For example, Zhao proposed a chiller fault detection method based on Support Vector Domain Description (SVDD), which is a one-class classification technique describing the support vector of data distribution. By training SVDD models for each fault type, they extended the similar idea to a chiller fault diagnosis strategy in [75]. Noticing that training a one-class classification model for each particular fault type is computationally costly, alternative methods were adopted to formulate the FDD task

directly as a multiple classification problem. To list a few, Du proposed to utilize Fisher Discriminant Analysis (FDA) and Principal Component Analysis (PCA) to diagnose multiple sensor faults in AHU [58]. Keigo employed semi-supervised FDA to detect building energy faults and adopted Decision Boundary Analysis (DBA) to discover the hidden relationship between the extracted features and the corresponding faults [82]. However, all of the aforementioned work is restricted to modelling each type of fault separately with single ("flat") class labels and ignores valuable prior information on fault dependence, which could otherwise be exploited ("fused") to improve the detection performance of the machine learning method [83]. Moreover, when dealing with complex building systems, the number of fault types (classes) is expected to be large, while usually, only small amount of labeled data for each fault class is available. From a statistical learning perspective, adopting a "flat" multi-class learning method and ignoring prior information will result in loss of valuable information, thus leading to degraded performance [84,85].

Secondly, the presence of different fault severity levels is well acknowledged in experiments but has long been ignored for FDD purpose. In a real building cooling system, faults naturally exhibit at various levels of severity due to different system /component degradations [33,41,86,87]. For instance, in the research of typical chiller faults, condenser fouling is a physical obstruction which is caused by the aggregation of non-decomposable chemical substances in the condenser tubes. It lowers the effective heat transfer coefficient and decreases the water flow rate in a manner consistent with the degree of aggregation. Hence the severity/degree of fault provides researchers/system managers valuable information to optimize maintenance actions, as well as to set priorities for different system scenarios. On the other hand, the advancement of the sensor network technology has considerably improved the capability to monitor temperature, flow rate, pressure, etc. with a refined spatial-temporal granularity [33]. In short, detecting severity level in a data-driven

framework is not only favorable but also doable. Until now no work has tried to identify how serious the identified fault is.

Thirdly, due to noisy and non-informative variables, direct utilization of raw data may lead to degraded classification performance [88,89]. Motivated by the success of multivariate statistics, researchers tend to use dimension reduction techniques to eliminate noisy information. In existing works, the most extensively used dimension reduction method is Principal Component Analysis (PCA) [64]. However, PCA does not explicitly reveal the cause-and-effect relationships and thus is only suitable for detecting abnormal conditions. For the purpose of diagnosis, supervised techniques such as Linear Discriminant Analysis (LDA) [14], Joint Angle Analysis (JAA) [90], and Partial Least Squares (PLS) [91] have been applied to diagnose building faults. Those methods merely project the data in high dimension into a lower dimensional subspace while the essential variables are still unintelligible. In addition, dimension reduction is insufficient to warrant maximum improvement of FDD in the projected subspace. Thus, more efficient techniques, which can pick out prime streams and improve the classification, should be employed to the building FDD.

Moreover, unlike in laboratory environment, deploying sensors in active building ACMV systems is expensive and challenging. Noticing that not all the experimental measurements are essential or accessible, recently researchers have proposed to choose features through empirical knowledge or experience. Authors of [15] selected 24 primary variables according to their availability and control requirement in real systems. Based on expert knowledge of chiller system structure and fault coverage, authors of [92] selected 12 variables to construct an FDD model. Zhao et al. added extra 8 variables to the base selection of [93], and the results showed that 16 variables increased the fault detection rate [74]. Nevertheless, there is no rigorous guarantee that the empirically chosen variables are optimal. Therefore, a systematic strategy that could rationally select optimal variables with theoretical guarantees is

in need for building FDD.

1.4 Contributions

The main contributions of this dissertation are based on the building FDD study. Firstly, the distance-based multiple classifier is investigated to identify fault types as well as the corresponding severity level in a two-stage data-driven FDD strategy. Then, to recognize both fault type and the corresponding severity level in a unified learning framework, a learning algorithm that incorporates inter-class fault dependence information is developed. Besides, the feature selection task that aims at improving FDD performance is studied. The contributions are summarized as follows:

1. **FDD with Distance-based Multiple Classifier.**

   1) The proposed DAFC method does not require an in-depth understanding of chiller system's physics and it is more efficient and flexible compared with model-based methods;

   2) it diagnoses typical faults with high accuracy and distinguishes the corresponding severity level;

   3) since sensor data is collected continuously and stored in the BMS, the strategy may update the training data sets each time an unknown fault is detected;

   4) due to the advantages of data-based algorithms, the proposed strategy can be easily integrated into the BMS and then can be configured to display FDD results as desired.

2. **FDD with Tree-structured Learning Method.**

   1) A TFDK method is deduced to incorporate the fault dependence information, and thus higher FDD accuracy is achieved compared with traditional

classification methods;

2) an on-line learning method is developed to accommodate streaming data, which enables seamless integration to the BMS and sequential decision-making for ACMV schedule;

3) detailed information about the building performance is presented by identifying fault severity levels, hence providing researchers and building managers with more options on taking actions to handle the faults.

3. **FDD with Feature Selection.**

1) The IGFF-based FDD strategy can be applied to different kinds of building working conditions regardless of the adopted FDD algorithms;

2) IGFF outperforms state-of-the-art feature selection methods in terms of the FDD performance;

3) theoretically, IGFF achieves a guaranteed near optimal solution for mutual information maximization with an optimum upper bound;

4) by applying the IGFF method to building fault data collected by ASHRAE RP-1312, relevant variables have been selected and can be referred directly by both experiment designers and AHU FDD researchers.

1.5 Outline of Contents

The remaining part of this report is organized as follows:

Chapter 2 introduces preliminary works about the data-driven fault detection and diagnosis method.

Chapter 3 considers the FDD problem from the dimension reduction aspect. A Discriminant Analysis-based Fault Classification (DAFC) method is proposed to

diagnose fault types as well as the corresponding severity levels at two separate stages.

Chapter 4 tries to improve the FDD performance and identify the fault types as well as the corresponding severity level in a unified framework. A Tree-structured Fault Dependence Kernel (TFDK) method is proposed to incorporate the inter-class information.

Chapter 5 considers the fact that among the huge amount of measured variables, some features are more correlated with the failures than others. The Information Greedy Feature Filter (IGFF) method is proposed to tackle the feature selection objective.

Chapter 6 gives conclusions and suggests possible future research directions.

Chapter 2

Preliminaries

2.1 Classification Overview

Classifying an observation means predicting a qualitative response for it since the classifier assigns the observation to a category or class. Usually, the classification methods firstly predict the probability for each category, which is the basis for making the classification.

Classification methods applied in this dissertation are common ones, including Linear and Quadratic Discriminant Analysis (LDA and QDA), Logistics Regression (LR), Decision Tree (DT), AdaBoost (AB), Neural Networks (NN), and Multiple Support Vector Machine (MSVM). In the following subsections, these methods will be briefly introduced. For more details, interested readers are referred to [94] and [95].

2.2 Linear and Quadratic Discriminant Analysis

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) are among the most popular classifiers and have been wildly used in building FDD field due to its simplicity and good track record.

To start with, let's recall that Bayes Classifier predicts the most likely class label $C = j$ given the feature measurements $X = x \in \Re^p$

$$\begin{aligned} f(x) &= \underset{j=1,\ldots,K}{\arg\max} P(C = j \,|X = x) \\ &= \underset{j=1,\ldots,K}{\arg\max} P(X = x \,|C = j)\, \pi_j \end{aligned} \tag{2.1}$$

where $\pi_j = P(C = j)$ is the prior probability of class j.

LDA approximates this rule by modeling the conditional class densities as multivariate normal

$$h_j(x) = P(X = x \,|C = j) = N(\mu_j, \Sigma) \tag{2.2}$$

where each class j has its own mean $\mu_j \in \Re^p$, but shares a common covariance matrix $\Sigma \in \Re^{p \times p}$. Hence

$$h_j(x) = \frac{1}{(2\pi)^{p/2} \det(\Sigma)^{1/2}} \exp\left\{-\frac{1}{2}(x - \mu_j)^T \Sigma^{-1} (x - \mu_j)\right\} \tag{2.3}$$

Since $\log(\cdot)$ is a monotone function, let's consider maximizing $\log(h_j(x)\, \pi_j)$ over $j = 1, \ldots K$

$$\begin{aligned} f^{LDA}(x) &= \underset{j=1,\ldots K}{\arg\max} \delta_j(x) \\ &= \underset{j=1,\ldots K}{\arg\max} \left[x^T \Sigma^{-1} \mu_j - \tfrac{1}{2} {\mu_j}^T \Sigma^{-1} \mu_j + \log \pi_j\right] \end{aligned} \tag{2.4}$$

Then, the classification rule reduces to

$$\hat{f}^{LDA}(x) = \underset{j=1,\ldots,K}{\arg\max} \hat{\delta}_j(x) \tag{2.5}$$

where $\hat{\delta}_j(x)$ is the estimated discriminant function of class j. Thus, the LDA classifier [1] is

$$\hat{\delta}_j^L(x) = x^T \hat{\Sigma}^{-1} \hat{\mu}_j - \frac{1}{2} \hat{\mu}_j^T \hat{\Sigma}^{-1} \hat{\mu}_j + \log \hat{\pi}_j \tag{2.6}$$

[1] Here, LDA classifier is formulated from the probability aspect. It is different from the LDA algorithm described in Chapter 3 which is driven in terms of dimension reduction.

Getting back to the general discriminant problem Eq. (2.3), if the Σ_j are not assumed to be equal, then the convenient cancellations in Eq. (2.4) will not occur; particularly, the pieces quadratic in x remains. Then, the QDA classifier is

$$\hat{\delta}_j^Q(x) = -\frac{1}{2}\log|\Sigma_j| - \frac{1}{2}(x-\hat{\mu}_j)^T\hat{\Sigma}_j^{-1}(x-\hat{\mu}_j) + \log\hat{\pi}_j \tag{2.7}$$

A quadratic equation $\{x : \delta_i(x) = \delta_j(x)\}$ describes the decision boundary between each pair of classes i and j.

The QDA formulation for FDD purpose will treat normal and faulty conditions as different classes, train the QDA model based on the historical data, and assign the testing data to a certain class based on the outputs of classifier Eq. (2.7).

2.3 Logistic Regression

On the contrast with LDA and QDA, instead of estimating $\hat{\mu}_j$, $\hat{\Sigma}_j$, and $\hat{\pi}_j$, one can directly estimate the parameters of a linear classifier $\log\left(\frac{P(C=1|X=x)}{P(C=2|X=x)}\right) = \beta_0 + \beta^T x$ (take the two-class classification as an example). This formulation is the Logistics Regression (LR). LDA and QDA are quite efficient when the true class conditional densities are normal, while LR is more robust to situations where they are not normal. In real practice they tend to perform similarly in a variety of situations.

Note that $P(C=2|X=x) = 1 - P(C=1|X=x)$, and

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta^T x \Leftrightarrow \frac{p}{1-p} = \exp\left(\beta_0 + \beta^T x\right) \Leftrightarrow p = \frac{\exp\left(\beta_0+\beta^T x\right)}{1+\exp(\beta_0+\beta^T x)} \tag{2.8}$$

Therefore the assumption is,

$$P(C=1|X=x) = \frac{\exp\left(\beta_0 + \beta^T x\right)}{1+\exp\left(\beta_0 + \beta^T x\right)} \tag{2.9}$$

$$P\left(C=2\,|X=x\right)=\frac{1}{1+\exp\left(\beta_0+\beta^T x\right)} \tag{2.10}$$

Suppose given samples (x_i, y_i), $i = 1, ..., N$, and assume that the class labels are conditionally independent given $x_i, ..., x_n$. Since the Likelihood is $L(\beta_0, \beta) = \prod_{i=1}^{n} P\left(C = y_i\,|X = x_i\right)$, and the Log likelihood is $\ell(\beta_0, \beta) = \sum_{i=1}^{n} \log P\left(C = y_i\,|X = x_i\right)$. For convenience, define

$$u_i = \begin{cases} 1 & y_i = 1 \\ 0 & y_i = 2 \end{cases} \tag{2.11}$$

Then, the Log Likelihood can be written as

$$\begin{aligned} &\ell\left(\beta_0, \beta\right) = \sum_{i=1}^{n} \log P\left(C = y_i\,|X = x_i\right) \\ &= \sum_{i=1}^{n} \{\log\left(\frac{\exp\left(\beta_0+\beta^T x\right)}{1+\exp(\beta_0+\beta^T x)}\right) u_i \\ &+ \log\left(\frac{1}{1+\exp(\beta_0+\beta^T x)}\right)(1-u_i) \\ &= \sum_{i=1}^{n} \left\{u_i\left(\beta_0+\beta^T x\right) - \log\left(1+\exp\left(\beta_0+\beta^T x\right)\right)\right\} \end{aligned} \tag{2.12}$$

Thus, the coefficients are estimated by maximizing the likelihood,

$$\widehat{\beta}_0, \widehat{\beta}^T = \underset{\beta_0\in,\beta\in p}{\arg\max} \sum_{i=1}^{n} \left\{u_i\left(\beta_0+\beta^T x\right) - \log\left(1+\exp\left(\beta_0+\beta^T x\right)\right)\right\} \tag{2.13}$$

After computing $\hat{\beta}_0$, $\hat{\beta}$, the classification of an input $x \in \Re^p$ is given by

$$\hat{f}^{LR}(x) = \begin{cases} 1 & \text{if } \hat{\beta}_0+\hat{\beta}^T > 0 \\ 0 & \text{if } \hat{\beta}_0+\hat{\beta}^T \leqslant 0 \end{cases} \tag{2.14}$$

Extend Eq. (2.13) to multiple classes,

$$\widehat{\beta}_{0,j}, \widehat{\beta}_j{}^T = \underset{\beta_0\in,\beta\in p}{\arg\max} \sum_{i=1}^{n} \left\{u_i\left(\beta_{0,j}+\beta_j{}^T x\right) - \log\left(1+\exp\left(\beta_{0,j}+\beta_j{}^T x\right)\right)\right\} \tag{2.15}$$

2.4 Decision Tree

The tree-based method for predicting y from a feature vector $x \in \Re^p$ divides the feature space into rectangles, and then fits a very simple model in each rectangle. Rectangles are achieved by making successive binary splits on the predictor variables $X_1, ..., X_p$. Since the splitting rules that segment the feature space can be summarized as a tree, this approach is known as the Decision Tree (DT) method.

Suppose $(x_i, y_i), i = 1, ..., N$ is the training data, where $y_i \in \{1, ..., K\rceil$ and $x \in \Re^p$. The classification tree defined M regions (rectangles) $R_1, ..., R_M$ that correspond to the tree leaves, respectively. Also, each region R_j is assigned with a class label $c_j \in \{1, ..., K\}$. Since each region R_j contains a subset of the training data, say including n_j points, the predicted class label is just the most commonly occurring class among those points. A new point is then classified by

$$\hat{f}^{Tree}(x) = \sum_{j=1}^{M} c_j \cdot 1\{x \in R_j\} = c_j \text{ such that } c \in R_j \tag{2.16}$$

Usually, a classification tree is built by the recursive binary splitting with the classification error rate as the splitting criterion. Since an observation is assigned to the most frequently occurring class, the classification error rate equals the fraction of the training observations in that region, which do not belong to the most frequent class:

$$E = 1 - \max_k (\hat{p}_{ik}) \tag{2.17}$$

where $\hat{p}_{ik}$ is the percentage of training observations in the ith region that belong to the kth class. Nevertheless, in practice classification error is not sensitive enough for tree-growing; thus, two other preferable measures are introduced in the following.

The first one is the Gini index which is defined as

$$Gini = \sum_{k=1}^{K} \hat{p}_{ik} \left(1 - \hat{p}_{ik}\right) \tag{2.18}$$

It is a measure of total variance across K classes. Since its value is small if all the $\hat{p}_{ik}$s are close to zero or one, it is referred to as a measure of node purity. A node that contains predominantly observations from a single class would generate a small Gini index.

The other one is the cross-entropy, given by

$$CE = -\sum_{k=1}^{K} \hat{p}_{ik} \log \hat{p}_{ik} \tag{2.19}$$

Since $0 \leq \hat{p}_{ik} \leq 1$ and $-\hat{p}_{ik} \log \hat{p}_{ik} \geq 0$, the cross-entropy is near zero if all the $\hat{p}_{ik}$s are near zero or near one. Consequently, similar to the Gini index, the cross-entropy is small if the ith node is pure. Actually, the aforementioned two measures are quite similar numerically.

Due to their higher sensitivity to node purity compared with that of the classification error rate, the Gini index and the cross-entropy are typically used to evaluate the quality of a particular split when growing a classification tree. However, the classification error rate is preferable if the objective is the prediction accuracy of a final pruned tree.

2.5 AdaBoost

Boosting generally improves the accuracy of a given learning algorithm that combines the results of several classifiers, such as classification trees. This is achieved by combining B hypotheses (with a weighted voting) generated by repeating training with different training subsets.It tries to transform a weak learning algorithm into

a strong one.

A weak classifier only performs slightly better than random guess (i.e., the error rate of a binary decision task is less than 50%). The hypothesis h^{weak} is obtained by applying a weak classifier. Then, the Boosting classifier (a strong one) is computed as the linear combination of the B weak classifiers

$$\hat{f}^{Boost}(x) = \text{sign}\left(\sum_{b=1}^{B} \alpha_b \hat{f}^b(x)\right) \tag{2.20}$$

AdaBoost (AB) is a basic boosting algorithm. Given training data $(x_i, y_i), i = 1, ..., N$, the algorithm initializes the weights by $w_i = \frac{1}{n}$ for each i. Then, for $b = 1, ..., B$, a classifier, which is usually a classification tree $\hat{f}^{Tree}$, is fitted to the training data with weights $w_1, ..., w_n$. Next, calculate the weighted misclassification error

$$e_b = \frac{\sum_{i=1}^{N} w_i \cdot \{y_i \neq \hat{f}^{Tree,b}(x)\}}{\sum_{b=1}^{B} w_i} \tag{2.21}$$

By letting $\alpha_b = \log\{\frac{(1-e_b)}{e_b}\}$, weights are updated as

$$w_i \leftarrow w_i \cdot \exp\left(\alpha_b \cdot 1\{y_i \neq \hat{f}^{Tree,b}(x)\}\right) \tag{2.22}$$

At last, the AdaBoost classifier is returned as

$$\hat{f}^{Boost}(x) = \text{sign}\left(\sum_{b=1}^{B} \alpha_b \hat{f}^{Tree,b}(x)\right) \tag{2.23}$$

2.6 Neural Networks

Neural Networks (NN) are biologically inspired to mimic the human brain. Neuron is the basic processing element in NN. Let $X = (X_1, X_2, ..., X_N)$ be the N inputs

applied to the neuron Neu_j, α_i is the weight for input X_i and b is a bias, then the output of the neuron is

$$Neu_j = \sum_{i=0}^{i} x_i \alpha_i - b, \text{ and } V = f(u) \tag{2.24}$$

Neurons are connected with connection links. Each link has a weight that multiplied with transmitted signal in the network. Each neuron has an activation function to determine the output. Among all kinds of activation functions, nonlinear ones, such as sigmoid and step functions, are commonly uses.

2.7 Support Vector Machine

Support Vector Machine (SVM) implements the Structural Risk Minimization (SRM) principle to minimize the VC[2] confidence interval. Rather than simply minimizing the training error, SVM minimizes the structural risk which expresses an upper bound on generalization error. Assuming that the data is linearly separable, the algorithm aims at finding the smallest possible w or maximum separation (margin) between the two classes. This can be formally expressed as a quadratic optimization problem with soft margin

$$\begin{aligned} &\min \tfrac{1}{2}\|w\|^2 + C\sum \xi_i \\ &s.t. \begin{cases} y_i\left(w^T x_i + b\right) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \end{aligned} \tag{2.25}$$

The margin between two classes is extended by allowing some amount of slackness which is represented by ξ's. By definition, ξ's should be larger or equal to 0. If $0 < \xi \leq 1$, it can be inferred that the data point lies somewhere between the margin

[2]VC dimension (for VapnikChervonenkis dimension) is a measure of the capacity (complexity, expressive power, richness, or flexibility) of a space of functions that can be learned by a statistical classification algorithm.

and the correct side of hyperplane. On the contrary, if $\xi > 1$, it means the data point is misclassified.

Then, the Lagrangian for this problem is

$$\mathcal{L}(w, b, \xi, \alpha, r) = \min_{w} \tfrac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i \left\{ y_i \left(w^T x_i + b \right) - 1 + \xi_i \right\} - \sum_i r_i \xi_i \tag{2.26}$$

Firstly, take the partial derivative of $\mathcal{L}$ with respect to w, b, ξ,

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial \xi} = 0 \tag{2.27}$$

which results in,

$$\begin{aligned} w &= \sum \alpha_i y_i x_i \\ \sum \alpha_i y_i &= 0 \\ \alpha_i &= C - r_i, \forall i \end{aligned} \tag{2.28}$$

Since the dual solution also have to satisfy KKT conditions

$$\begin{aligned} \alpha_i \left\{ y_i \left(w^T x_i + b \right) - 1 + \xi_i \right\} &= 0 \\ r_i \xi_i &= 0 \end{aligned} \tag{2.29}$$

Plug back into the original problem Eq. (2.26), the dual optimization objective is

$$\begin{aligned} &\max_{\alpha \geq 0} L(\alpha) = \sum_i \alpha_i - \tfrac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ &s.t. \begin{cases} \sum_i \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \forall i \end{cases} \end{aligned} \tag{2.30}$$

Chapter 3

FDD with Distance-based Classifier

3.1 Introduction

As introduced in Chapter 1, data-driven method for building FDD is becoming more and more popular due to the rapid development of machine learning techniques. Unlike traditional black-box model whose parameters describe the correlations between inputs and outputs, data-driven methods use pure mathematical models to uncover the hidden information buried in the historical data [24,27,35].

Motivated by the fact that the distance-based classification is an effective fault detection and diagnosis technique but hampered by the high dimension. In this chapter, the author proposes a distance-based classifier, the Discriminant Analysis-based Fault Classification (DAFC) method, which is based on Linear Discriminant Analysis and K-means Clustering technique. The FDD for a typical chiller system is formulated as a multiple classification problem. Firstly, the proposed DAFC method is applied to project the original data into a lower dimensional space and classify the seven typical chiller faults as well as the normal condition. Then, fault severity

level is identified similarly by DAFC.

The rest of this chapter is arranged as follows. Section 3.2 formulates the FDD problem for chillers as a multiple classification problem. Section 3.3 introduces the necessary techniques applied in this chapter. The details of the proposed two-stage data-driven FDD strategy are presented in Section 3.4. The proposed strategy is tested by the experimental data in Section 3.5. Section 3.6 summarizes the chapter.

3.2 Problem Statement

3.2.1 Chiller and Faults

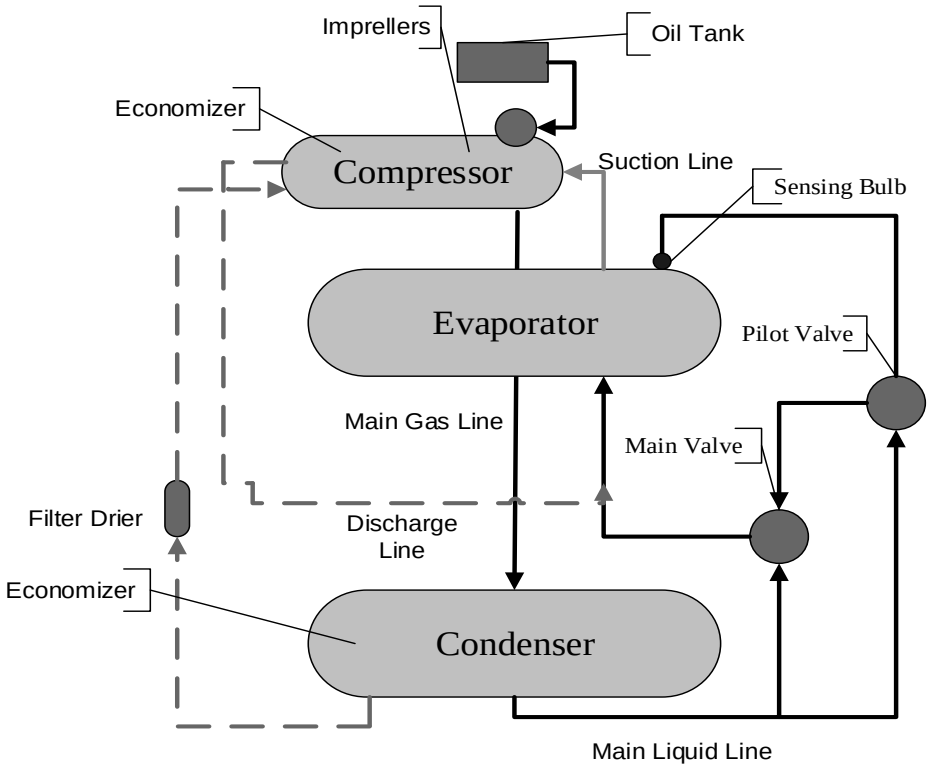


Figure 3.1: Schematic diagram of chiller components and refrigerant flow paths. A typical centrifugal chiller system consists of the evaporator, compressor, condenser, economizer, motor, pumps, fans, distribution pipes, etc..

In tropic area, such as Singapore, cooling systems, especially those with centrifugal

chillers, account for a large portion of the energy usage of ACMV systems [92]. The Chiller studied in this dissertation is a standard centrifugal water-cooled chiller with a motor-driven compressor. As shown in Figure 3.1, the chiller system consists of the following main components: evaporator, compressor, condenser, economizer, motor, pumps, fans, distribution pipes, etc..

At the beginning of a refrigerant cycle, liquid refrigerant is distributed along the evaporator and sprayed through small holes with high pressure in a distributor to uniformly coat each evaporator tube. Here the liquid refrigerant absorbs enough heat from the chiller water that is circulating through the evaporator tubes, thus turning into refrigerant vapor. The chiller water is cooled down during this process. Then the gaseous refrigerant is drawn through the eliminators (which remove droplets of liquid refrigerant from the gas) and delivered into impellers where the gas will be compressed. Once the compression is completed, the gas is discharged into the condenser, where baffles distribute the compressed refrigerant gas evenly across the condenser tube bundle. Cooling tower water which circulates through the condenser tubes absorbs heat from the refrigerant, thus turning the gaseous refrigerant into liquid. The liquid refrigerant then drains from the bottom of the condenser and passes through an expansion valve, where its pressure and temperature are decreased. At last, the low-pressure mixture enters the evaporator and starts the next cycle.

As for chiller faults study, the ASHRAE RP-1043 has reported a significant number of possible faults and failures [33]. However, not all of them are practical for further examination as part of the fault detection and diagnostics scheme [11]. Faults chosen for experimental testing are expected to be detected and diagnosed by monitoring the thermodynamic states of the chiller. According to the survey of ASHRAE RP-1043, researchers and building operators should pay attention to seven typical chiller faults (regarding their frequency of occurrence and the economic losses caused by

them):

- Condenser Fouling (CF)
- Excess Oil (EO)
- Reduced Condenser Water Flow Rate (FWC)
- Reduced Evaporator Water Flow Rate (FWE)
- Non-Condensable in the Refrigerant (NC)
- Refrigerant Leak/undercharge (RL)
- Refrigerant Overcharge (RO)

3.2.2 Distance-based Classification and Dimension Reduction

Usually, the BMS acquires and stores mass sensor data from various infrastructure systems of a building to monitor and control the building's performance. Thus, the BMS provides a large amount of data. Nevertheless, data collected under certain circumstances (normal or faulty) is high dimensional due to the complexity of building systems.

The proposed classification approach for distinguishing fault working conditions from the normal working condition is driven based on the K-means Clustering. The testing data set is identified as belonging to one cluster according to two criteria [1]. Namely, 1), the testing data set is the closest to the corresponding cluster center (there are several clusters for fault and normal conditions); 2), distances between the testing data set and the cluster center (the closest one) are within the distance

[1]Note that in Section 3.4.1, three decision rules are utilized since DAFC method outputs three kinds of diagnosis results: normal condition, known fault, or unknown fault. They are different from the two classification criteria mentioned here.

range of that cluster. Here, the "closest" means that the average distance between the testing data points and the corresponding center is the smallest. However, due to the "curse of dimensionality", a fixed number of data points become increasingly sparse as the Euclidean dimension increases. Thus, in high dimensional space, the distances between the data points become relatively uniform, and the notion of the nearest neighbor of a point is meaningless [96]. In general, the distances between points are calculated as L_k norms. The meaningfulness of distances among points in high dimension is sensitive to the value of k. Aggarwal has proved that the Manhattan distance measure (L_1 norm) is consistently more preferable than the Euclidean distance measure (L_2 norm) for high dimensional data mining applications [97]. Moreover, Linear Discriminant Analysis (LDA) has been proved to be an effective dimension reduction method that preserves as much of the class discriminatory information as possible when projecting high dimensional data into a lower dimensional space [98,99].

As a result, in this chapter, LDA and Manhattan distance are adopted to tackle the "curse of dimensionality". In the following section, LDA is introduced from the aspect of dimension reduction.

3.3 LDA Algorithm for Dimension Reduction

The optimal transformation matrix in LDA is obtained by minimizing the within-class distance and maximizing the between-class distance simultaneously, thus achieving maximum class discrimination [100].

3.3.1 LDA for Two Classes

In the two-class case, first define the within-class distance as

$$S_1^2 = \sum_{x_1 \in \text{Class}\,1} (x_1 - \mu_1)(x_1 - \mu_1)^t, \tag{3.1}$$

$$S_2^2 = \sum_{x_2 \in \text{Class}\,2} (x_2 - \mu_2)(x_2 - \mu_2)^t, \tag{3.2}$$

where x_1 and x_2 are samples from Class 1 and Class 2; μ_1 and μ_2 are the centers of Class 1 and Class 2.

Define v as the transformation vector which projects the original data sets to a lower dimensional space. Thus, $v^t x_i$ is the projection of x_i on the direction of v, and the projection of class center μ_i is $v^t \mu_i$. By considering projected samples $\tilde{x}_i = v^t x_i$ and projected class centers $\tilde{\mu}_i = v^t \mu_i$, the within-class distance for Class i $(i = 1, 2)$ after projection is derived as

$$\begin{aligned}
\tilde{S}_i^2 &= \sum_{x_{i \in \text{Class}i}} (\tilde{x}_i - \tilde{\mu}_i)(\tilde{x}_i - \tilde{\mu}_i)^t \\
&= \sum_{x_{i \in \text{Class}i}} (v^t x_i - v^t \mu_i)(v^t x_i - v^t \mu_i)^t \\
&= \sum_{x_{i \in \text{Class}i}} (v^t (x_i - \mu_i))(v^t (x_i - \mu_i))^t \\
&= \sum_{x_{i \in \text{Class}i}} v^t (x_i - \mu_i)(x_i - \mu_i)^t v \\
&= v^t \left(\sum_{x_{i \in \text{Class}i}} (x_i - \mu_i)(x_i - \mu_i)^t \right) v \\
&= v^t S_i^2 v,
\end{aligned}$$

By defining the within-class scatter matrix as $S_w = S_1^2 + S_2^2$, which measures the variances of original classes, the within-class scatter matrix after projection can be written as

$$\tilde{S}_w = \tilde{S}_1^2 + \tilde{S}_2^2 = v^t S_1^2 v + v^t S_2^2 v = v^t (S_1^2 + S_2^2) v = v^t S_w v. \tag{3.3}$$

Secondly, define the between-class scatter matrix as

$$S_b = |\mu_1 - \mu_2|^2 = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t, \tag{3.4}$$

where S_b measures separation between the means of two classes. Then, the projected between-class distance is

$$\begin{aligned}\tilde{S}_b = |\tilde{\mu}_1 - \tilde{\mu}_2|^2 &= |v^t\mu_1 - v^t\mu_2|^2 \\ &= v^t(\mu_1 - \mu_2)(\mu_1 - \mu_2)^t v \\ &= v^t S_b v.\end{aligned}$$

Since the aim of LDA is to find the optimal transformation vector v by maximizing the ratio of between-class distance and within-class distance after projection, the objective function can be written as

$$J(v) = \frac{\tilde{S}_b}{\tilde{S}_w} = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{S}_1^2 + \tilde{S}_2^2} = \frac{v^t S_b v}{v^t S_w v}. \tag{3.5}$$

Then, calculate the derivative of v and setting the function to be 0

$$\begin{aligned}\frac{d}{dv}J(v) &= \frac{(\frac{d}{dv}v^t S_b v)v^t S_w v - (\frac{d}{dv}v^t S_w v)v^t S_b v}{(v^t S_w v)^2} \\ &= \frac{(2S_b v)v^t S_w v - (2S_w v)v^t S_b v}{(v^t S_w v)^2} = 0.\end{aligned}$$

By solving $(S_b v)v^t S_w v - (S_w v)v^t S_b v = 0$, one can get a generalized eigenvalue problem

$$S_b v = \lambda S_w v, \ (\lambda = \frac{v^t S_b v}{v^t S_w v}). \tag{3.6}$$

If S_w has full rank (its inverse exists), one can convert Eq. (3.6) to a standard eigenvalue problem

$$S_w^{-1} S_b v = \lambda v. \tag{3.7}$$

Since $S_b = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t$, points of $S_b x$ are in the same direction as $(\mu_1 - \mu_2)$

for any vector x from the original space

$$\begin{aligned} S_b x &= (\mu_1 - \mu_2)(\mu_1 - \mu_2)^t x \\ &= \alpha(\mu_1 - \mu_2), (\alpha = (\mu_1 - \mu_2)^t x). \end{aligned} \tag{3.8}$$

Thus, the eigenvalue problem can by solved as

$$\begin{aligned} {S_w}^{-1} S_b v &= {S_w}^{-1} \alpha(\mu_1 - \mu_2) \\ &= \alpha[{S_w}^{-1}(\mu_1 - \mu_2)] \\ &= \lambda v, \end{aligned} \tag{3.9}$$

where the corresponding parameters are $\alpha = \lambda$ and $S_w^{-1}(\mu_1 - \mu_2) = v$. So the solution is

$$v = S_w^{-1}(\mu_1 - \mu_2). \tag{3.10}$$

3.3.2 LDA for K Classes

Next, the above calculation process is extended to K-class case. The within-class scatter matrix S_w and the between-class scatter matrix S_b are expressed as

$$S_w = \sum_{i=1}^{K} S_i^2 = \sum_{i=1}^{K} \sum_{x_i \in \text{Class}\, i} (x_i - \mu_i)(x_i - \mu_i)^t, \tag{3.11}$$

$$S_b = \sum_{i=1}^{K} n_i (\mu_i - \mu)(\mu_i - \mu)^t, \tag{3.12}$$

$$\mu = \frac{1}{N} \sum_{\forall x} x = \frac{1}{N} \sum_{i=1}^{K} n_i \mu_i, \tag{3.13}$$

where n_i is the sample number of cluster i, N is the sample number of clusters, and the maximum rank of S_b is $K - 1$.

Define V as the transformation matrix which projects the original data sets to a lower dimensional space. Based on the derivation of the two-class problem, the

scatter matrices for the projected samples is written as

$$\tilde{S}_w = V^t S_w V, \tag{3.14}$$

$$\tilde{S}_b = V^t S_b V. \tag{3.15}$$

Thus, the objective function is

$$J(V) = \frac{\det\left(\tilde{S}_b\right)}{\det\left(\tilde{S}_w\right)} = \frac{\det\left(V^t S_b V\right)}{\det\left(V^t S_w V\right)}. \tag{3.16}$$

Then, solve the generalized eigenvalue problem

$$S_b V = \lambda S_w V, \ (\lambda = \frac{V^t S_b V}{V^t S_w V}). \tag{3.17}$$

The problem in (3.17) has at most $K-1$ linearly independent eigenvectors, represented by v_1, v_2, ...v_{K-1} correspondingly. The optimal projection matrix V for a D-dimensional subspace is given by the eigenvectors corresponding to the largest D eigenvalues. Thus, LDA can project the original data set to a subspace of dimension (K - 1) at the most.

3.4 Two-stage FDD Strategy with DAFC Method

The proposed FDD strategy is a two-stage data-driven method which formulates the chiller FDD as a multiple classification problem. The goal is to distinguish the 7 fault working conditions from the normal working condition for a centrifugal water-cooled chiller and recognize different fault severity levels based on the predefined severity level information. The fault types are recognized at the first stage, and the corresponding fault severity levels are identified at the second stage. To

formulate the FDD problem for chiller system, N observations are considered for each training cluster at the fault type recognition stage, n observations for each training cluster at the severity recognition stage, and d measurements (dimensions) in each observation.

At the fault type diagnosis stage, centers for the fault clusters are defined as

$$\mu_k = \text{mean}([P_k]_{N\times d}), \tag{3.18}$$

where μ_k is the center of cluster k ($k = 1, ..., K$), and $[P_k]_{N\times d}$ is an $N \times d$ matrix which represents data set from cluster k. Each fault cluster includes data from all severity levels.

If all the clusters are well-separated, the average distance between points within one cluster and its center should be smaller than that between those points and other cluster centers

$$\text{mean}(\text{Dis}(p_k, \mu_k)) < \text{mean}(\text{Dis}(p_k, \mu_{\bar{k}})), \tag{3.19}$$

where p_k represent the points from cluster k, $\mu_{\bar{k}}$ represent other cluster centers, and $Dis(.,.)$ represents the distance between two points.

Here the distances between points of cluster k and the center μ_k are defined as Manhattan distances

$$\text{Dis}(p_{kj}, \mu_k) = \sum_{i=1}^{d} |p_{kij} - \mu_{ki}|, \tag{3.20}$$

where p_{kj} is the jth point in cluster k, $j = 1, 2, ..., N$, c_{ki} and p_{kij} are the ith coordinate values of p_{kj} and μ_k ($i = 1, 2, ..., d$).

Since the real-time monitoring aims at detecting and diagnosing faults and the corresponding severity levels for data set rather than a single data point. The monitoring data set (which contains data points collected during a period) is diagnosed

as a particular fault if it is the closest to the corresponding pre-defined fault cluster center and within its trained Manhattan distance range, which is defined as

$$U_k = \max(\text{Dis}(p_{kj}, \mu_k)), \tag{3.21}$$

$$L_k = \min(\text{Dis}(p_{kj}, \mu_k)). \tag{3.22}$$

Similarly, at the fault severity level diagnosis stage, cluster centers are defined as

$$\mu_k^l = \text{mean}([P_k^l]_{n\times d}), \tag{3.23}$$

where μ_k^l represents the cluster center of the lth $(l = 1, ..., L)$ severity level of fault type k, and $[P_k^l]_{n\times d}$ is an $n \times d$ matrix. The severity level can be identified by

$$\text{mean}(\text{Dis}(p_k^l, \mu_k^l)) < \text{mean}(\text{Dis}(p_k^l, \mu_k^{\bar{l}})), \tag{3.24}$$

where p_k^l represent the points from the lth severity level of fault type k, and $\mu_k^{\bar{l}}$ represent cluster centers of other severity levels (except level l) of fault type k.

Next, the FDD algorithms applied are explained in the two stages.

3.4.1 Stage One: Fault Type Detection and Diagnosis

The flow diagram of the first stage is depicted in Figure 3.2. In the off-line learning process, the historical data sets are taken from a fault library where a large amount of data measured under normal and fault conditions are stored. Firstly, define one normal cluster and seven fault clusters as eight training data sets. Secondly, remove obvious outliers (apparent measurement errors) and normalize the eight training clusters in the data pre-processing procedure. Thirdly, deal with the eight training data sets by LDA, which transforms the original high dimensional data into a lower

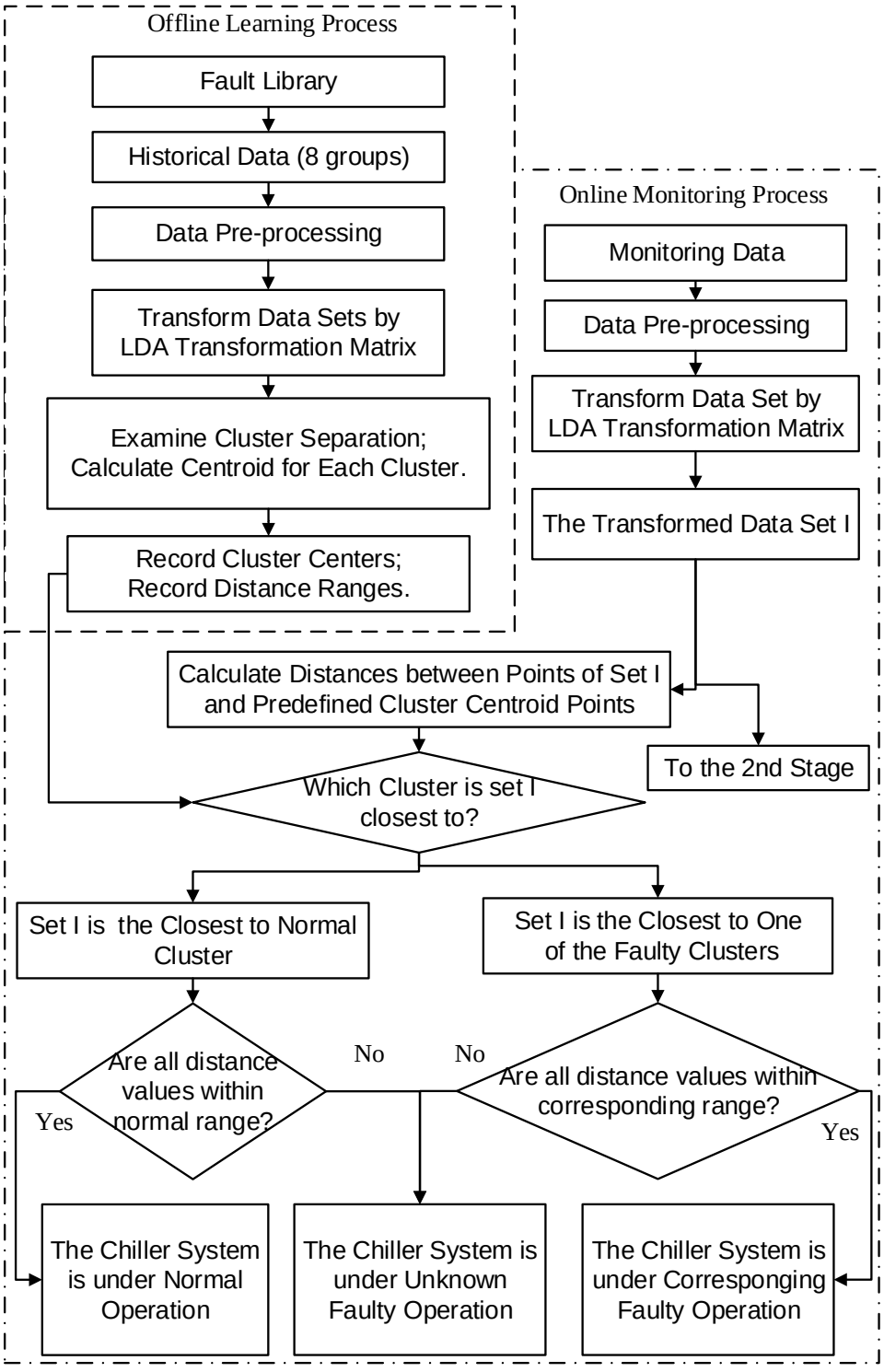


Figure 3.2: First stage of the DAFC-based FDD strategy.

dimensional space. Next, record the Cluster Centers (for simplification, named as CC points) in the lower dimensional space as comparison criteria. Here, Manhattan distance ranges between data points from one cluster and the corresponding cluster center are also recorded.

In the on-line monitoring process, the monitoring data set could be real-time streaming data collected from the BMS. After similar outlier removing and data normalization procedures, the monitoring data set is firstly projected to the lower dimensional space, which is exact the same as that in the training process. In the following step, distances between monitoring data points and the eight CC points are calculated. The diagnosis results are then generated according to the following decision rules:

- If the monitoring data set is the closest to the pre-defined normal cluster, and the distance values between the monitoring data points and the normal CC point are within the corresponding Manhattan distance range, the chiller system is diagnosed as working under normal condition.

- If the monitoring data set is the closest to the pre-defined fault cluster k, and the distance values between the monitoring data points and the CC point of fault k are within the corresponding Manhattan distance range, the chiller system is diagnosed as working under the kth fault condition.

- If the monitoring data set is the closest to one of the pre-defined clusters, but the distance values between the monitoring data points and the corresponding CC point are beyond the corresponding Manhattan distance range, the chiller system is diagnosed as working under an unknown fault condition.

The idea and calculation steps described above are depicted in details in Algorithms 3.1 and 3.2 (Fault Type Recognition) [2]. The jth monitoring data set is

[2]Due to the page length limit, DAFC algorithm for fault type recognition is written in two separate parts, namely, Learning Algorithm 3.1 and Monitoring Algorithm 3.2

Algorithm 3.1 DAFC for Fault Type Recognition I (Learning)

Input $[X_i]_{N\times d}, [X_j]_{m\times d}$
$TRX \Leftarrow \emptyset, TEX \Leftarrow \emptyset, CC \Leftarrow \emptyset, U \Leftarrow \emptyset$
$TRD \Leftarrow \emptyset, TED \Leftarrow \emptyset, D \Leftarrow \emptyset, J \Leftarrow \emptyset, Y \Leftarrow \emptyset, I \Leftarrow \emptyset$
for $i = 1, ..., K$ **do**
  $[TRX_i]_{N\times D} = \text{LDA}\left([X_i]_{N\times d}\right)$
  $CC_i = \text{mean}([TRX_i]_{N\times D})$
  for $t = 1 : N$ **do**
    $TRD_i(t) = \text{norm}((TRX_i(t,:) - CC_i), 1)$
  end for
  $U_i = \max([TRD_i(t)])$
end for

recognized as under the Ith fault condition (except that "$I = 1$" represents normal condition) if $JUDGE = 1$, and under an unknown fault working condition if $JUDGE = 2$.

3.4.2 Stage Two: Fault Severity Level Recognition

Figure 3.3 shows the flow diagram for the strategy of fault severity level recognition. Similar to the first stage, in the off-line learning process, historical sensor data collected under all the severity levels of each pre-defined fault condition is transformed into the lower dimensional space by LDA separately. After examining cluster separation, the severity level cluster centers are recorded as CC points. In the on-line monitoring process, the monitoring data set is transferred from the first stage, which has already been classified to a specific fault type. Firstly, as described in Algorithm 3.3 (Fault Severity Level Recognition), the monitoring data set is linked to the trained severity level CC points of the corresponding fault type. Then, severity level diagnosis results are generated by comparing the Manhattan distance values between all the monitoring data points and trained CC points.

Algorithm 3.2 DAFC for Fault Type Recognition II (Monitoring)

for $j = 1 : T$ **do**
  $[TEX_j]_{m\times D} = \text{LDA}\left([X_j]_{m\times d}\right)$
  for $i = 1 : K$ **do**
    for $t = 1 : m$ **do**
      $TED_{ji}(t) = \text{norm}\left((TEX_j(t,:) - CC_i), 1\right)$
    end for
    $DD_j(i) = \text{mean}([TED_{ji}(t)])$
  end for
  $J_j = arg\min\left([DD_j(i)]\right)$
  $Y_j = [TEX_j]_{m\times D}$, (transfer to the 2nd stage)
  $I = J_j$
  if $\max\left(TED_{jI}\right) < U_I$ **then**
    $JUDGE \Leftarrow 1$
  else
    $JUDGE \Leftarrow 2$
  end if
end for
Output $JUDGE, Y_j, I$

3.5 Experiments and Results

3.5.1 Data Description

The proposed fault detection framework is tested with the data collected from the ASHRAE RP-1043 project. As a brief introduction, one primary goal of the project was to obtain state measurements for a typical cooling system under normal, and various fault conditions. A 90-ton centrifugal water-cooled chiller is used, which is relatively small such that a comprehensive experiment design is possible, and it also bears enough representatives of chillers used in larger installations [33]. The experiment was conducted in an indoor environment with a nearly constant ambient temperature of 72°F, and the specifications of ARI (Air-Conditioning and Refrigeration Institute) Standard 550 for Centrifugal and Rotary Screw Water-Chilling Packages were adopted as the test requirements [33]. Sensor measurement is transferred to a database from the MicroTech controllers, which are mounted on the

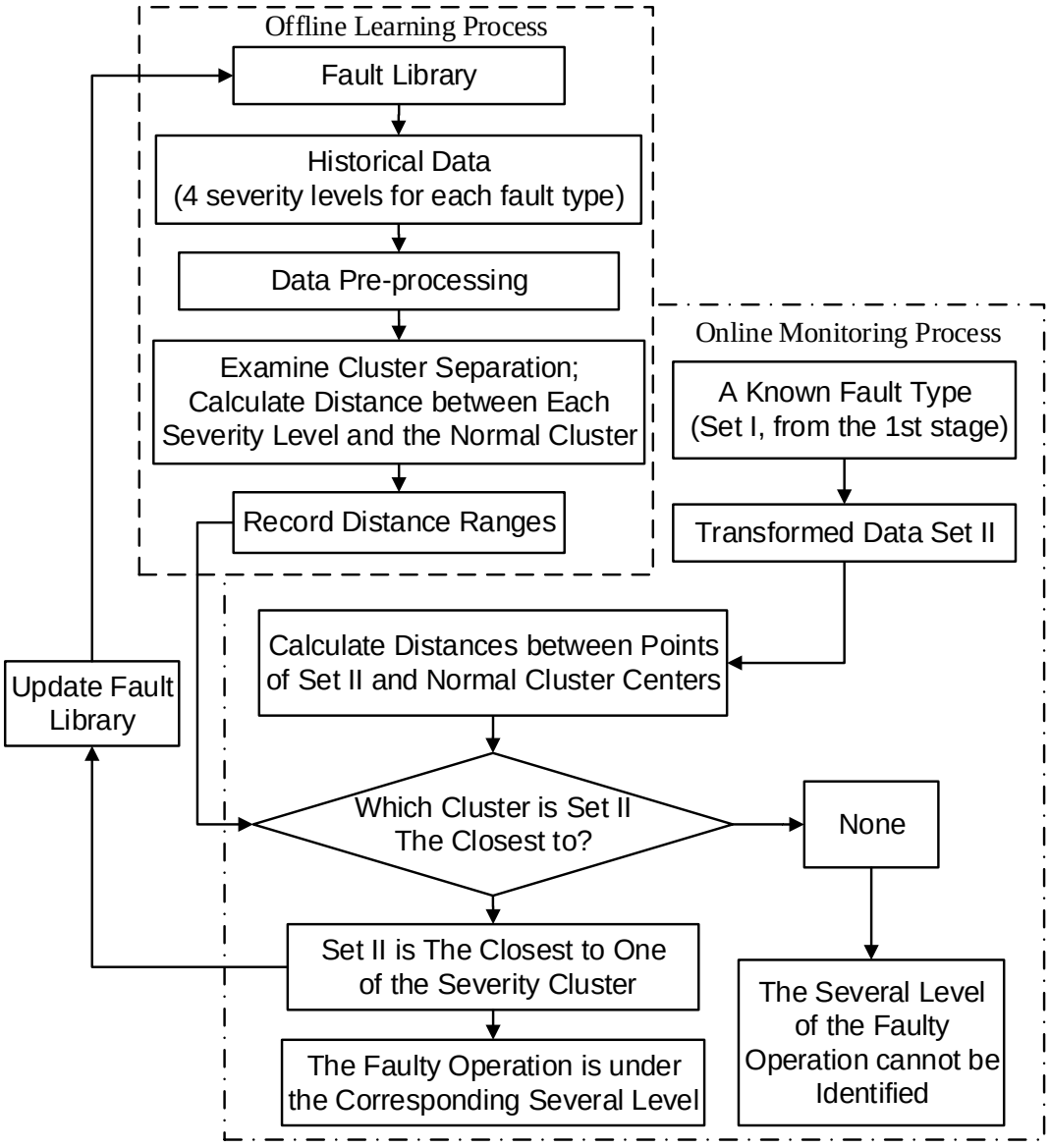


Figure 3.3: Second stage of the DAFC-based FDD strategy.

chiller. The test standard is controlled by three Johnson Controls Inc Air Handling Unit (JCI AHU) controllers on an N2 bus which is an RS-485 network. As shown in Figure 3.4, RS-485 is connected to the PC through COM Port 1 via an RS-485 to RS-232 converter.

During the experiment, nine typical faults suggested by the ASHRAE RP-1043 survey were introduced at multiple severity levels. More than 60 tests were con-

Algorithm 3.3 DAFC for Fault Severity Level Recognition

Input $Y_j = [TEX_{ji}]_{m\times D}(i = J_j), [X_l{}^i]_{n\times d}$
Learning
$SLX \Leftarrow \emptyset, CC \Leftarrow \emptyset, SLD \Leftarrow \emptyset, D \Leftarrow \emptyset, SL \Leftarrow \emptyset$
for $i = 1, ..., K$ **do**
  for $l = 1 : L$ **do**
    $[SLX_l{}^i]_{n\times D} = \text{LDA}\left([X_l{}^i]_{n\times d}\right)$
    $CC_l{}^i = \text{mean}([SLX_l{}^i]_{n\times D})$;
  end for
end for
Monitoring
for $j = 1 : T$ **do**
  for $l = 1 : L$ **do**
    for $t = 1 : m$ **do**
      $SLD^i_{jl}(t) = \text{norm}\left((TEX_j i(t,:) - CC_l{}^i), 1\right), (i = J_i)$
    end for
    $DD^i_j(l) = \text{mean}([SLD^i_{jl}(t)])$
  end for
  $SL_j = arg\ \min\left([DD^i_j(l)]\right)$
end for
Output SL_j

ducted, and for each test, 64 variables, including direct sensor measurement and calculated physical indexes, were recorded once every 10 seconds. In this chapter, seven commonly encountered faults are taken into account. Those faults are emulated by various experimental methods as is summarized in Table 3.1. Three tests under normal conditions were named test 0, test 1, and test 2. The Condenser Fouling (CF) fault was emulated by plugging tubes into condenser. The Reduced Condenser Water Flow Rate (FWC) fault and Reduced Evaporator Water Flow Rate (FWE) fault were emulated directly by reducing water flow rate in the condenser and evaporator. The Refrigerant Overcharge (RO) fault and Refrigerant Leakage (RL) fault were emulated by reducing or increasing the refrigerant charge respectively. The Excess Oil (EO) fault was emulated by charging more oil than nominal. And the Non-Condensable in Refrigerant (NC) fault was emulated by adding Nitrogen to the refrigerant.

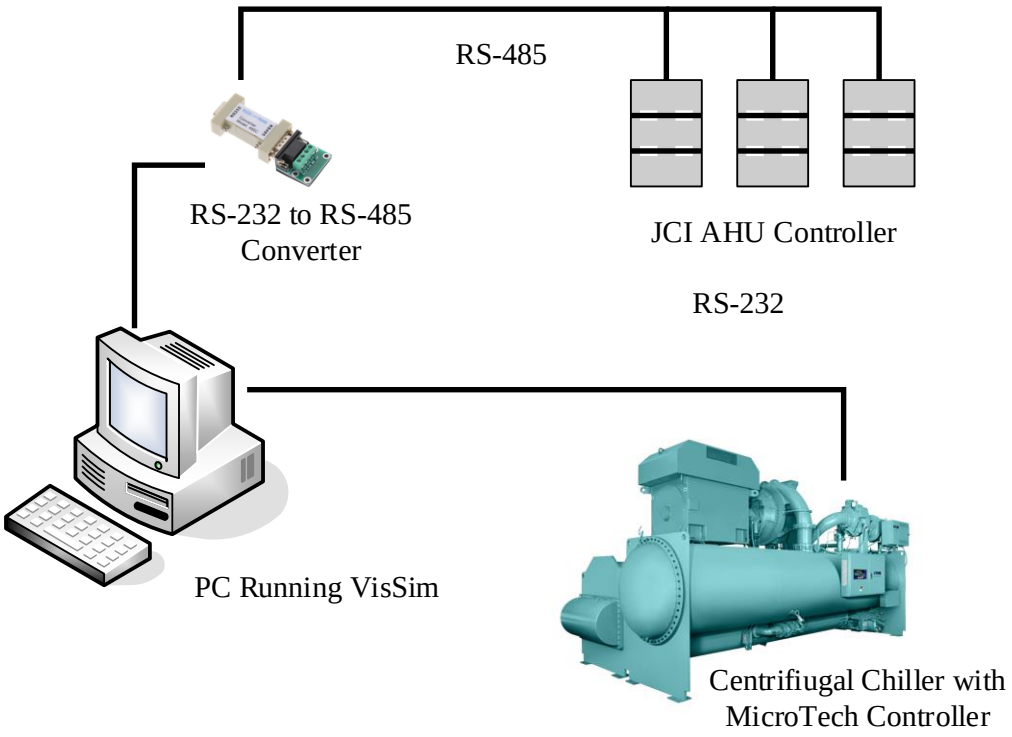


Figure 3.4: Schematic showing chiller test stand control interface.

3.5.2 Tackle the "Curse of Dimensionality"

Firstly, eight clusters (one normal cluster and seven fault clusters) are considered as the pre-defined training data sets. For each cluster, there are 433 measurements, and 64 features for each measurement. To check how are they separated in the $64-d$ space, one needs to calculate the distances between the data points and each cluster center, and then examine whether data points from one cluster are the closest to the corresponding center. In Figure 3.5, the distances between points from the normal cluster, which is named "NMpoints" in the figure, and all the pre-defined cluster centers, which are named "NM" and "fault 1-7" in the figure, are calculated as L_1 norms. The interval between 0 and the largest distance value is divided into 100 small distance intervals evenly. Figure 3.5 shows the probability densities of data points that fall into each interval, where different colors represent the prob-

Table 3.1: Typical chiller faults and corresponding experimental methods.

Name	SL	Description
NM	0/1/2	Tests run under normal conditions
CF	1/2/3/4	Plugged 20/33/49/74 tubes (out of 164) in the condenser
EO	1/2/3/4	Oil charge 14%/32%/50%/68% more than nominal
FWC	1/2/3/4	Reduce condenser water flow rate by 10%/20%/30%/40%
FWE	1/2/3/4	Reduce evaporator water flow rate by 10%/20%/30%/40%
NC	1/2/3/4	Adding 0.1/0.16/0.22/0.54 lbs Nitrogen to the refrigerant; displacing about 1.0%/1.8%/2.4%/5.6% of the volume at room temperature
RL	1/2/3/4	Refrigerant charge 10%/20%/30%/40% less than nominal
RO	1/2/3/4	Refrigerant charge 10%/20%/30%/40% more than nominal

*Note: SL means severity level; experiments of four severity levels are conducted for each fault; the normal experiment is repeated three times.

ability densities of distances between different pre-defined cluster centers and the normal data points. It is evident in Figure 3.5 that distributions of Manhattan distance between testing data points and various cluster centers are similar. This is a direct evidence of the "curse of dimensionality" which has been addressed in Fan's work-classification methods using all features are proved to be infeasible in high dimensional space due to the noise accumulation when estimating a significant number of noise features [101].

By projecting the high dimensional data to a lower dimensional space, LDA reduces the dimension. However, none of the original 64 input features is eliminated since the projected variables are linear combinations of the original features. Notice choosing the most significant features among the original inputs is a fascinating topic, the problem is addressed as the "feature selection" task for building FDD in Chapter 5. Figure 3.6 shows different distributions of Manhattan distance between

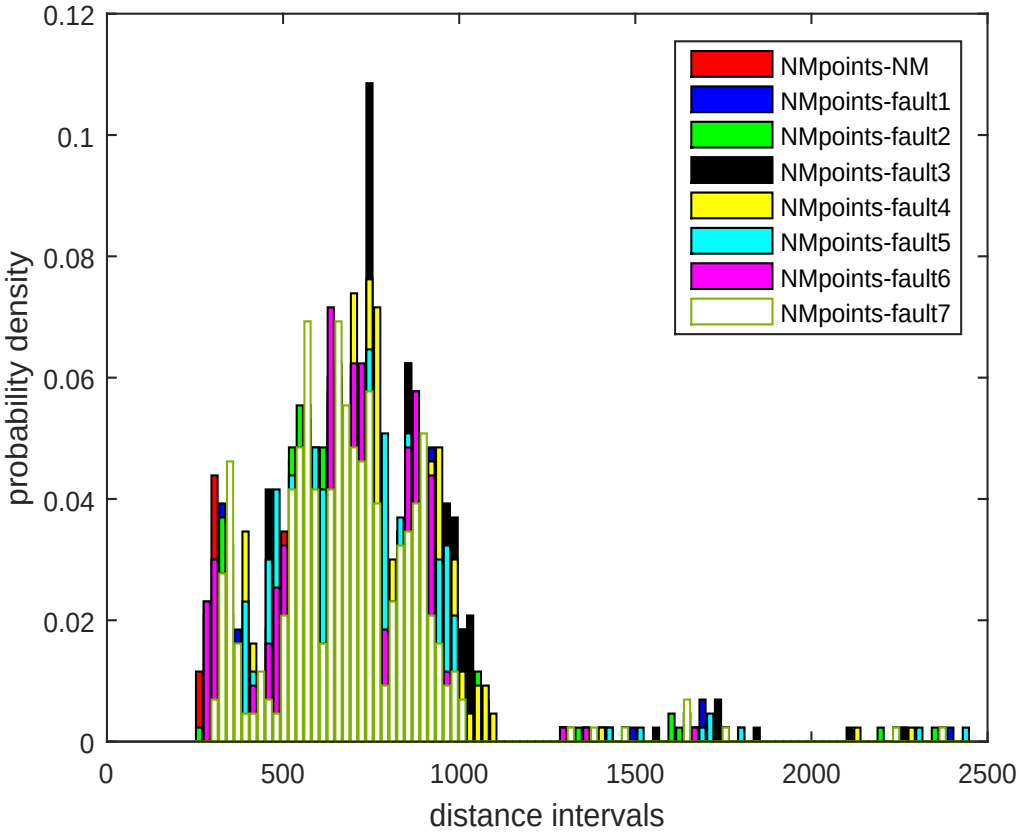


Figure 3.5: Distances between normal data points and each pre-defined fault cluster in the original high dimensional space. Distributions of Manhattan distance for different clusters are similar.

testing data points and different cluster centers after being dealt with by LDA. It shows that normal cluster is separated from other fault clusters in the lower dimensional space.

3.5.3 Fault Type Detection and Diagnosis Results

At the first stage, data sets of all fault types with four severity levels as well as the three normal tests are defined as the training data. They are named as NM, CF, EO, FWC, FWE, NC, RL, and RO. Given that the proposed strategy outputs FDD results for a bunch of monitoring data points, and the raw data of ASHRAE RP-1043 are collected every 10 seconds, sample size also represents the time duration spent on data collection and how long the FDD algorithm can recognize a fault after

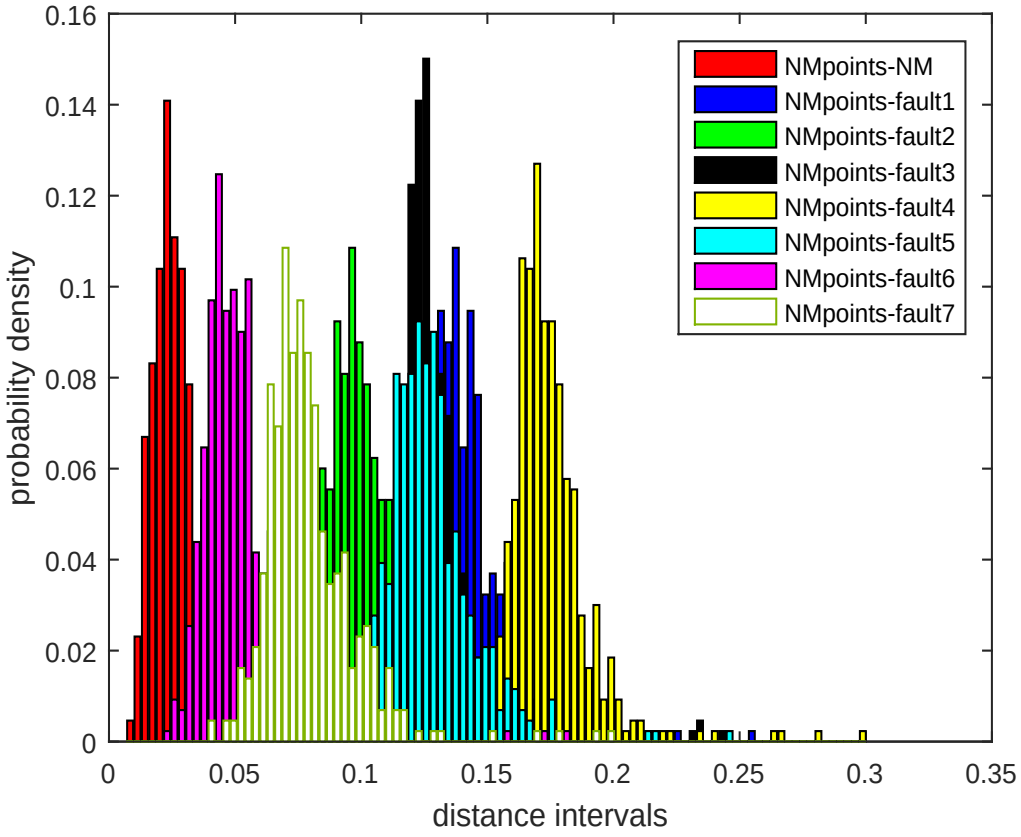


Figure 3.6: Distances between normal data pointss and each pre-defined fault cluster in the projected low dimensional space; processed by LDA. Distributions of Manhattan distance for different clusters are different.

its occurrence. For example, the FDD strategy is stated to be able to recognize a fault 10 minutes after its occurrence if 60 successive monitoring data points are diagnosed. In this chapter, training data is sampled from RP-0143 data files every 2 minutes over an 866-minute-long period; and testing data sets are picked from raw data files of RP-1043 with set size varying from 1 point to 714 successive points (represents data collecting time from 10 seconds to 119 minutes). To verify how the proposed algorithm diagnoses faults at the first stage, testing data sets sampled from different severity levels (and normal condition) are dealt with by Algorithms 3.1 and 3.2 correspondingly. Fault type diagnosis results are generated based on the algorithm outputs.

The fault type diagnosis accuracy is depicted in Figure 3.7. It shows that for all

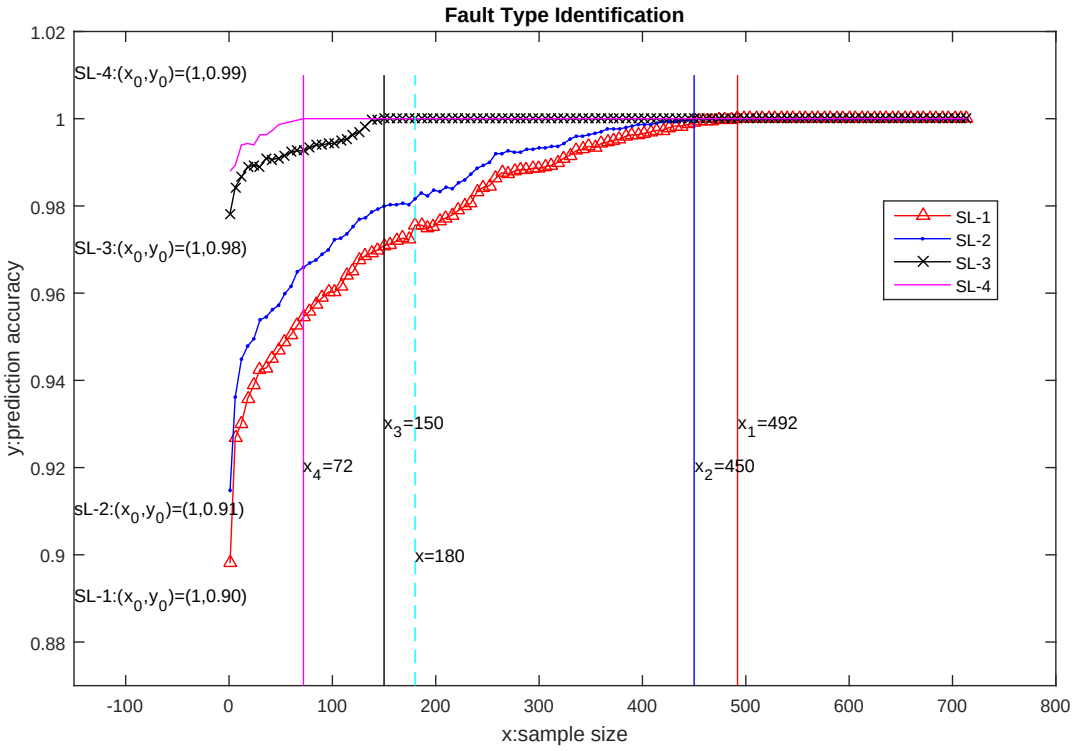


Figure 3.7: Accuracy of fault type detection and diagnosis as a function of the incremental testing sample size when LDA reduces the dimension to 7. Parallel lines $x_1/x_2/x_3/x_4$ mark where the accuracy curves (SL-1/2/3/4) converge to 1. Starting points for $SL-1$, $SL-2$, $SL-3$ and $SL-4$ are $(1, 0.90)$, $(1, 0.91)$, $(1, 0.98)$ and $(1, 0.99)$, respectively.

the testing groups Algorithms 3.1 and 3.2 present 100% prediction accuracy if the testing sample size is large enough. More specifically, when dealing with testing data from different severity levels, the accuracy converges to 100% at $x = 72$, $x = 150$, $x = 450$ and $x = 492$, respectively. Those results mean that the FDD strategy identifies a fault with 100% accuracy within 12 minutes, 25 minutes, 75 minutes, and 82 minutes respectively after its occurrence. Of course, when the fault is at low severity levels, it takes more than one hour to collect enough data to reach 100% diagnosis accuracy, which seems too long. However, Figure 3.7 also tells that even if the testing sample size is quite small Algorithms 3.1 and 3.2 still diagnose fault types with high accuracy (89.81%, 91.48%, 87.83% and 98.80%, respectively).

LDA can reduce the original data up to the dimension of $C-1$, where C is the num-

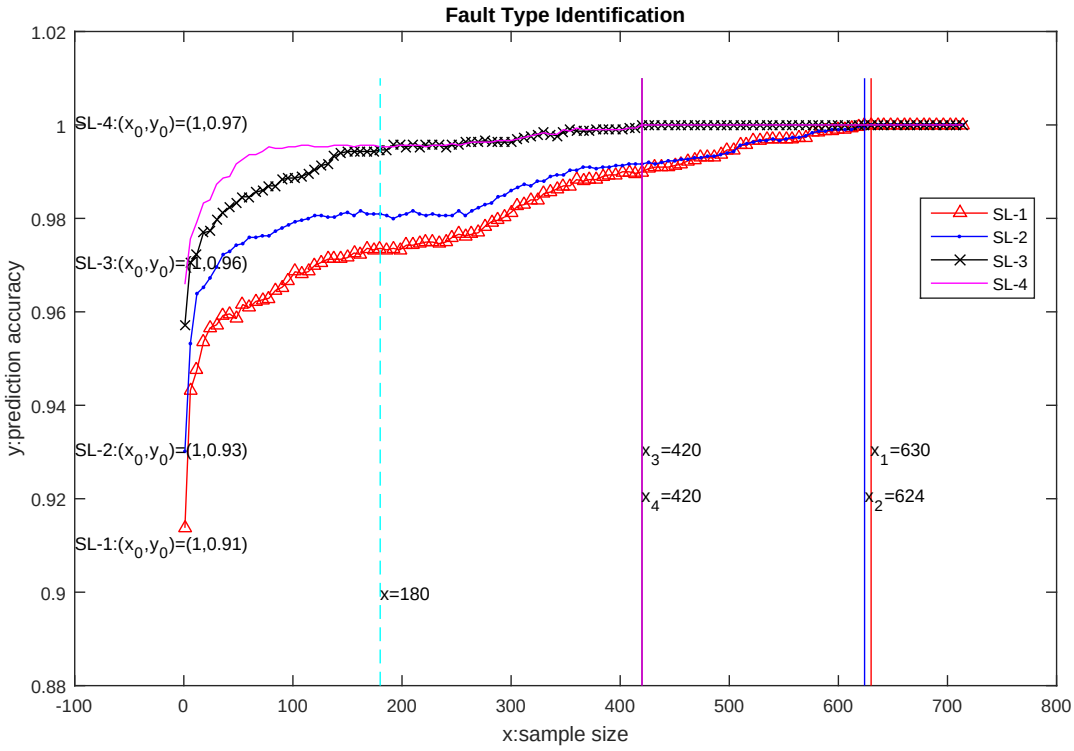


Figure 3.8: Accuracy of fault type detection and diagnosis as a function of the incremental testing sample size when LDA reduces the dimension to 6. Parallel lines $x_1/x_2/x_3/x_4$ mark where the accuracy curves (SL-1/2/3/4) converge to 1. Starting points for $SL-1$, $SL-2$, $SL-3$ and $SL-4$ are $(1, 0.91)$, $(1, 0.93)$, $(1, 0.96)$ and $(1, 0.97)$, respectively.

ber of total categories. Although none of the information provided by the original input features is eliminated via LDA projection, some of the original information is hidden under the linear combinations, which means that some information is "lost" after projection. Compared with the original dimension, the category number $C = 8$ (1 normal condition plus 7 faults) is small, C-1 (i.e., 7) is directed selected as the projected dimension at the first stage. In Figure 3.8,the accuracy of fault type detection and diagnosis is shown as a function of the incremental testing sample size when the dimension is reduced to 6. Based on the comparison one can see that when the dimension is reduced to 6, the DAFC method needs larger testing sample sizes than that in the case shown in Figure 3.7 (where the reduced dimension is 7) to reach 100% accuracy. Similarly at the severity level recognition stage, the original

data is reduced to dimension 3 since there are 4 severity levels for each fault type.

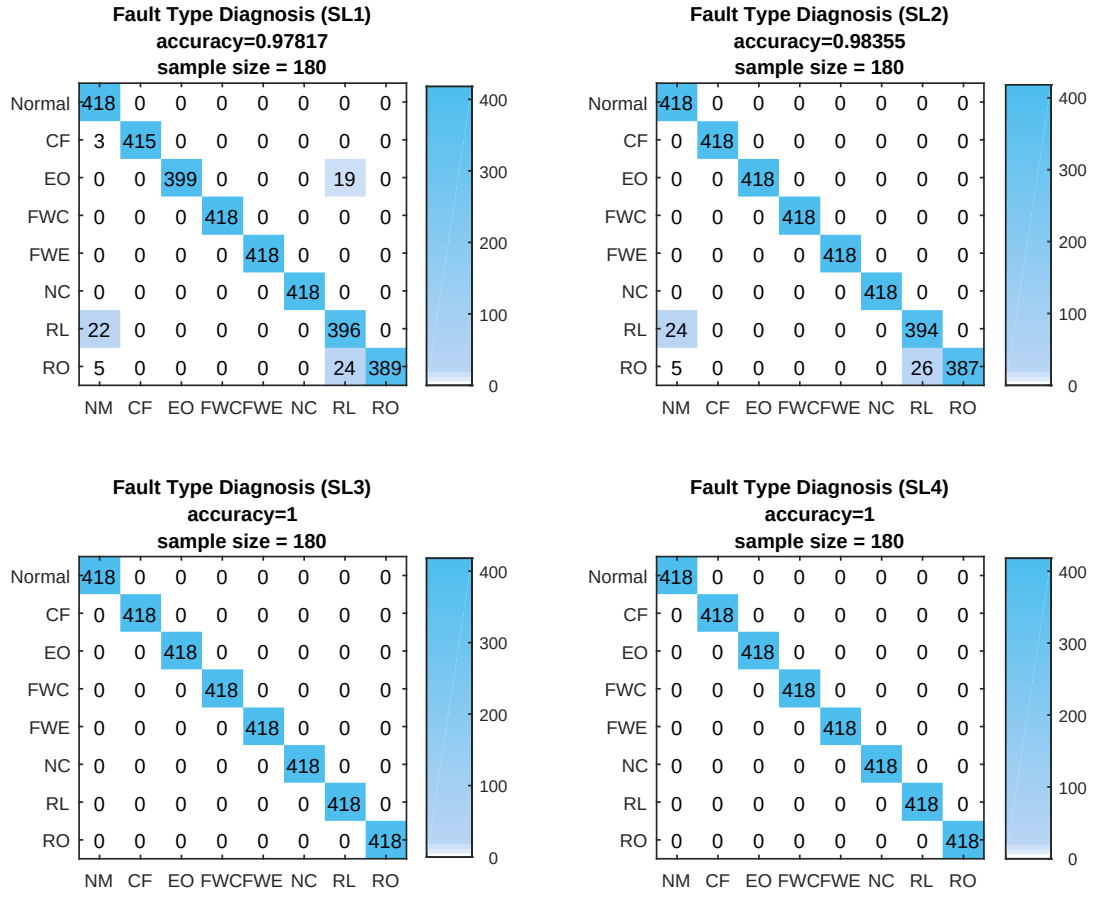


Figure 3.9: Confusion matrix for classifying testing datasets sampled from all categories by DAFC, which is trained by seven typical chiller faults.

In the real application,it is acceptable that the FDD strategy can identify a fault in 30 minutes after it occurs. Thus, one can set the monitoring sample size as 180, in which case the fault type diagnosis accuracy is higher than 97% (which is also acceptable) for all the tested data sets. In Figure 3.9 the confusion matrix of how the DAFC method recognize testing groups from all conditions (1 normal condition and 7 faults) when the testing sample size is 180 is shown. To be specific, $L = 180$ successive data points are picked to form one testing data set and shift the starting point successively along the raw data files, which can be described as $X_{tes} = X_{org}(1 + sift(i) : 1 + sift(i) + L, :)$. Since the shift vector is defined as $shift = 1 : 12 : (len_{raw} - L)$ and the raw data file includes $len_{raw} = 5191$

observations, there are 418 (which is the length of shift matrix) testing data sets for each category.

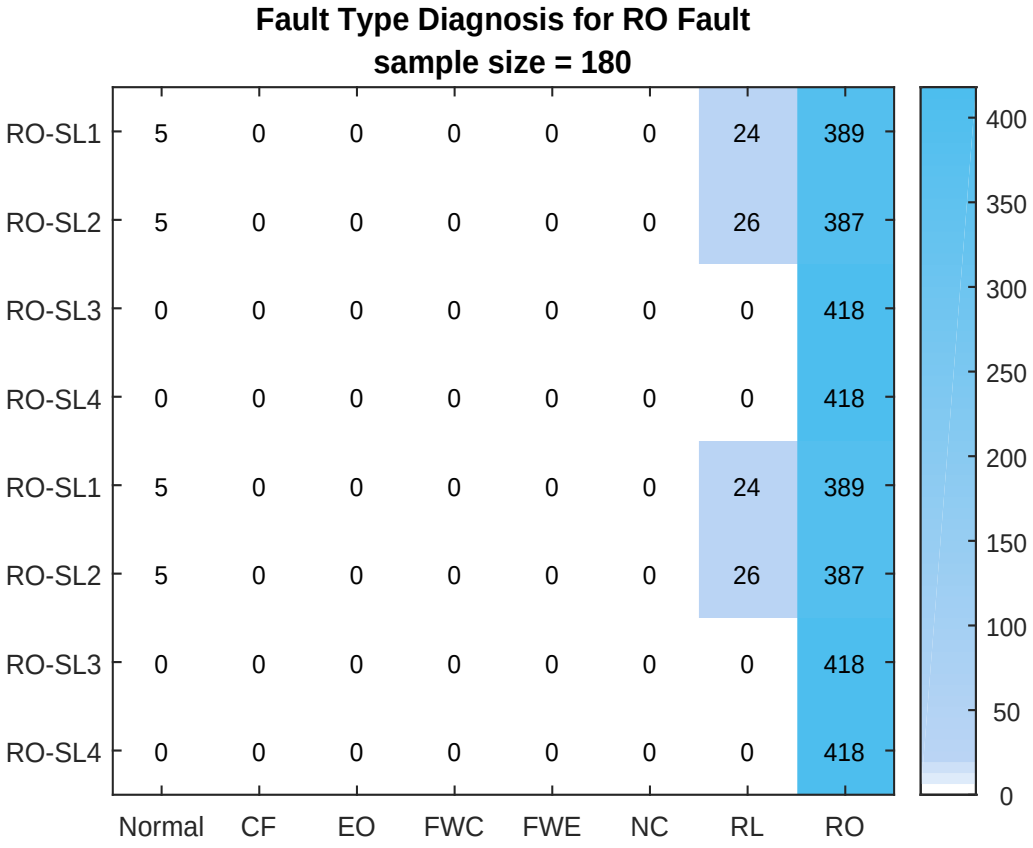


Figure 3.10: Confusion matrix for classifying testing datasets sampled from the RO fault file by DAFC, which is trained by seven typical chiller faults.

In particular, take the RO fault as an example to show how the proposed FDD strategy recognizes a known fault. The raw data of RO fault provided by RP-1043 is firstly packed into 8 testing data sets. Each severity level of NC fault provides two testing data sets, and each set contains 180 successive data points sampled from the raw data file. Just as mentioned, there are 418 testing sets in total. In the testing process, the 8 testing sets are labeled from 1 to 8 which are the same as the training labels of the normal condition and seven typical faults. The confusion matrix in Figure 3.10 shows that the DAFC method diagnoses all the testing data sets as RO fault, and Figure 3.11 tells that all the testing data is within the corresponding Manhattan distance range for RO fault (only one testing set for each severity level

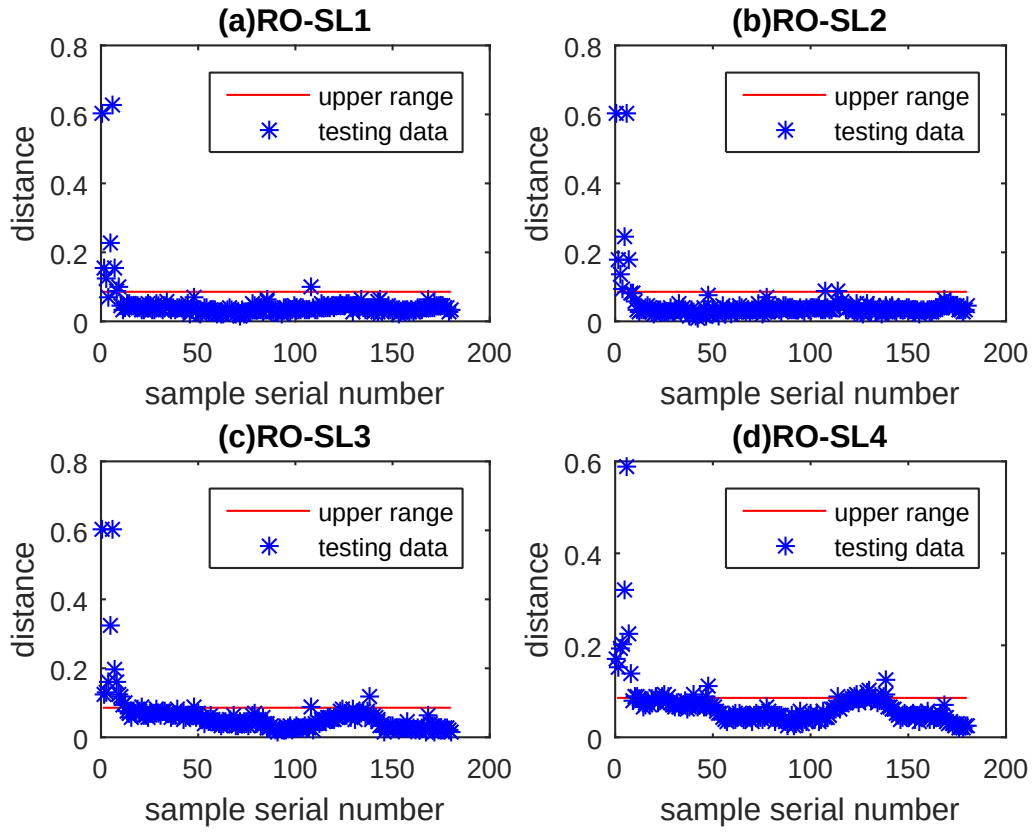


Figure 3.11: Distance values between testing datasets sampled from the RO fault file and the pre-defined RO center & the trained RO distance range.

is listed). In this case, Algorithms 3.1 and 3.2 successfully identify all the testing data sets as a known fault - the RO fault.

Other than the seven typical faults, the Defective Pilot Valve (DPV) fault is another fault which is tested in RP-1043, but not considered in this study since it is relatively rare compared with the seven typical ones. To show how the proposed FDD strategy deals with an unknown new fault, sample 8 testing data sets from the raw data file of DPV. Each of them contains 180 successive data points. Since the shift vector is defined as $shift = 1 : 1 : (len_{raw} - L)$ and raw data file includes $len_{raw} = 433$ observations, there are 253 (which is the length of shift matrix) testing data sets in total. According to the results shown in Figure 3.12, the DAFC classifier trained by the seven typical faults claims all the testing sets to be normal. However, by examining the Manhattan distance range, which is shown in Figure 3.13, one can

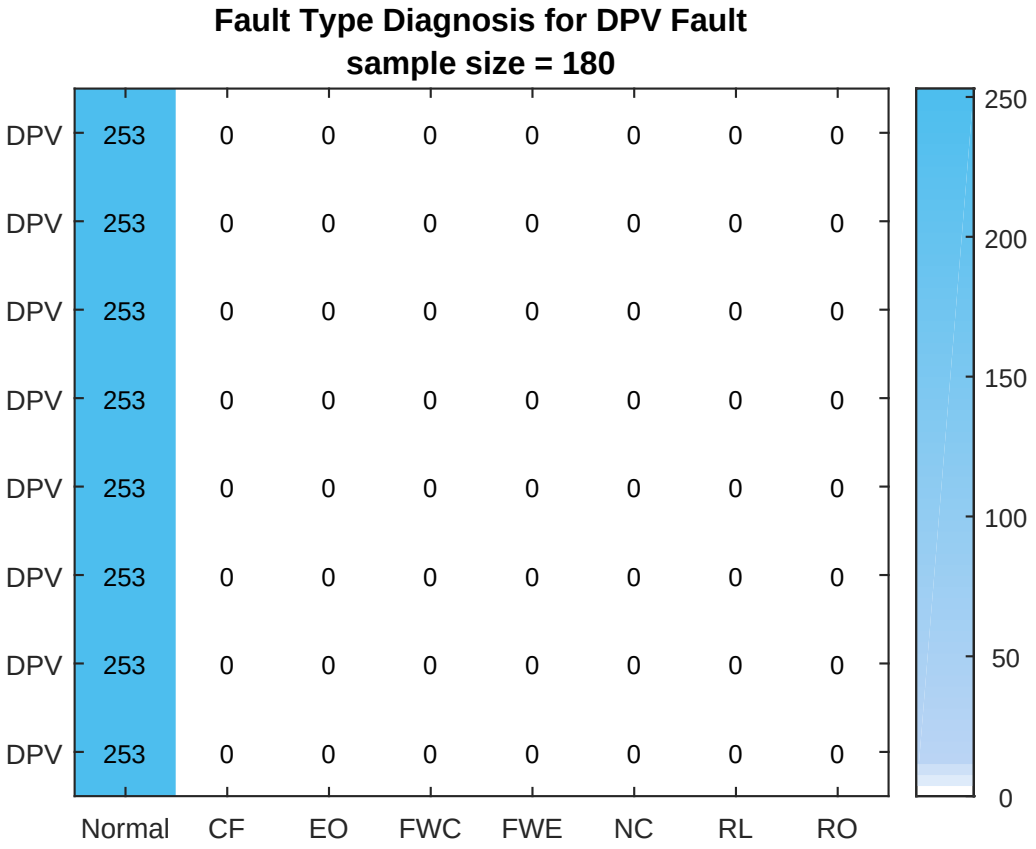


Figure 3.12: Confusion matrix for classifying testing datasets sampled from DPV fault file by DAFC, which is trained by seven typical chiller faults.

find that the distances between the testing data and pre-defined cluster centers are out of range (only one data set is shown in the figure). As a result, Algorithms 3.1 and 3.2 diagnose the testing data sets obtained from RP-1043 DPV fault test as an unknown new fault.

3.5.4 Fault Severity Level Recognition Results

Different fault severity levels in RP-1043 are defined by experimenters according to experience [33]. Marking all the faults with four severity levels for each fault qualitatively describes all the faults in a more detailed way. The severity level diagnosis provides more information about the faults after recognizing the fault types (such as how serious the fault is). Such information gives researchers and

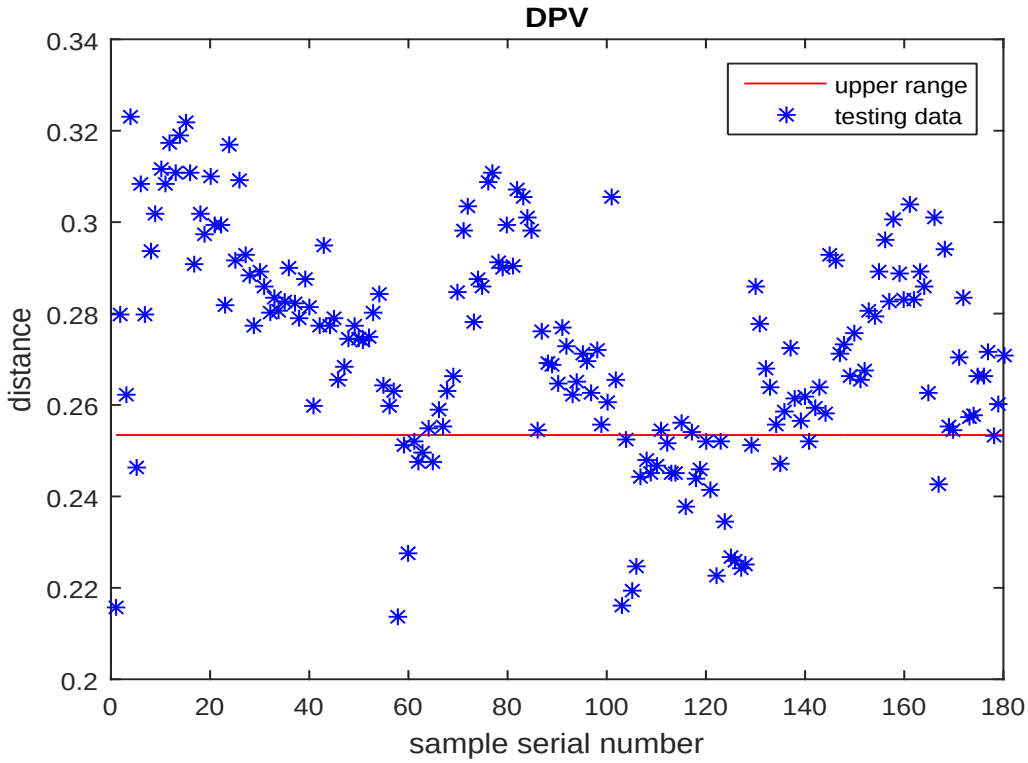


Figure 3.13: Distance values between testing datasets sampled from DPV fault file and the pre-defined NM center & the trained NM distance range.

building managers more options on taking actions to handle the faults.

To validate the algorithm for fault severity level recognition, the same testing data sets as that used at the previous stage are processed by Algorithm 3.3. For each fault, the severity level diagnosis accuracy as a function of testing sample size is generated. As shown in Figure 3.14, the diagnosis accuracy is relatively small with small testing sample size. However, when the sample size is larger than 500 the diagnosis accuracy converges to a value higher than 90% for each fault type. The process of recognizing fault severity level is the same as identifying fault types,which will not be repeated in this part. The confusion matrix for the recognition of severity level for the EO fault and FWE fault are shown in Figures. 3.15 and 3.16. Note that the severity level prediction accuracy for the EO fault is 84.17% when testing sample size is 180, which is still acceptable and much high than random guess (25% accuracy).

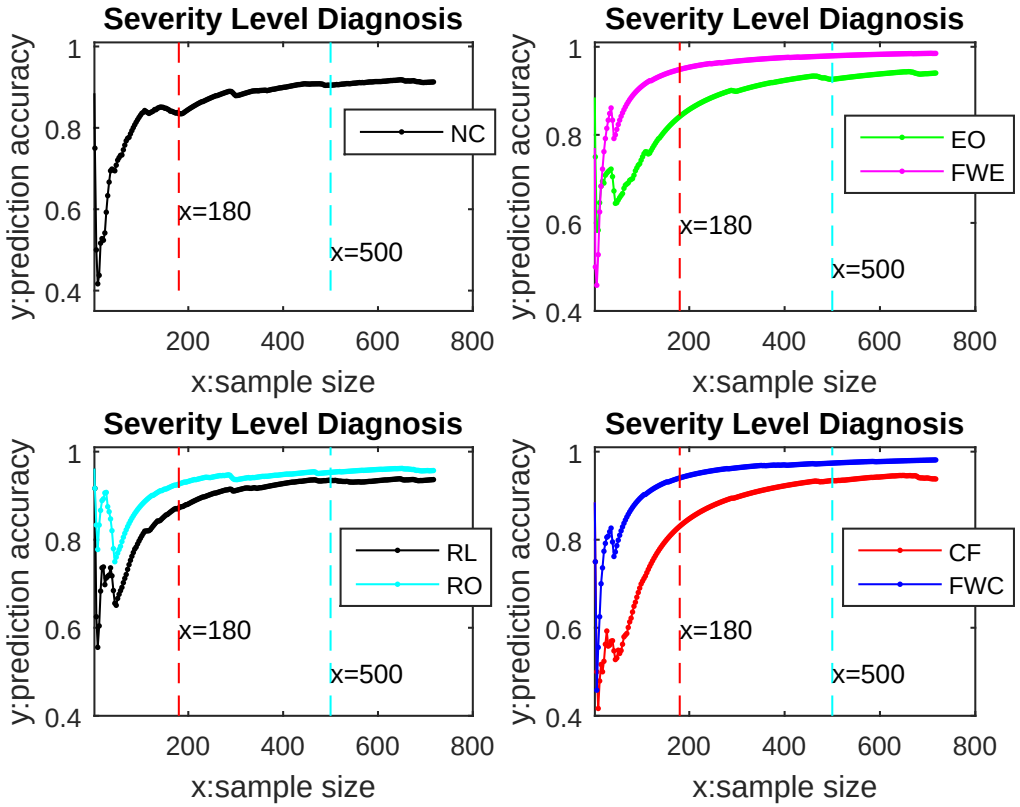


Figure 3.14: Accuracy of fault severity level diagnosis as a function of the incremental testing sample size.

3.6 Conclusion

In this chapter,a two-stage data-driven strategy for building chiller FDD is proposed. The strategy formulates the FDD as a multiple classification problem. LDA is adopted to project high dimensional data into a lower dimensional space where the data gets maximum class separation and remains its original class information as much as possible. The fault type and the corresponding severity level are detected and diagnosed if the monitoring data set is the closest to one of the pre-defined fault clusters and within the corresponding trained severity level Manhattan distance range. The proposed FDD strategy is validated using the ASHRAE RP-1043 data. It successfully detects and diagnoses seven typical faults and reports unknown faults. In addition, it also outputs how serious the diagnosed fault is. Compared with previous works, the proposed strategy is highly efficient and flexible.

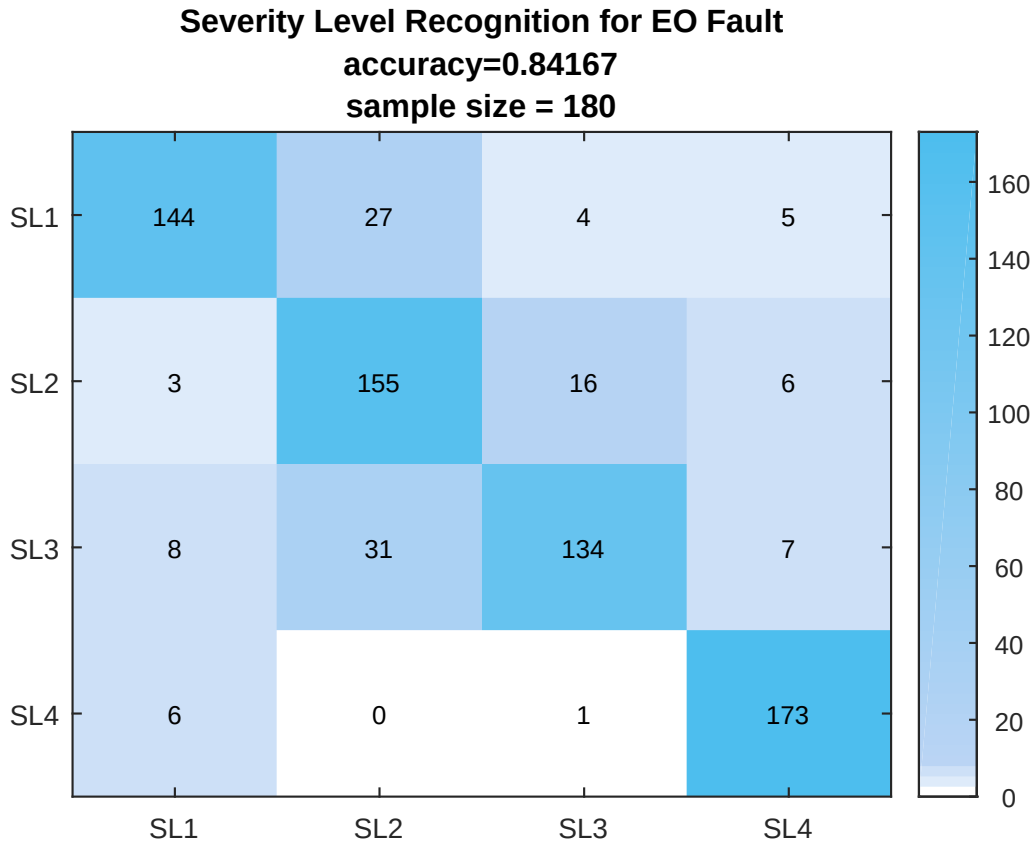


Figure 3.15: Confusion matrix for classifying testing datasets sampled from the EO fault file by DAFC, which is trained by its pre-defined four severity level.

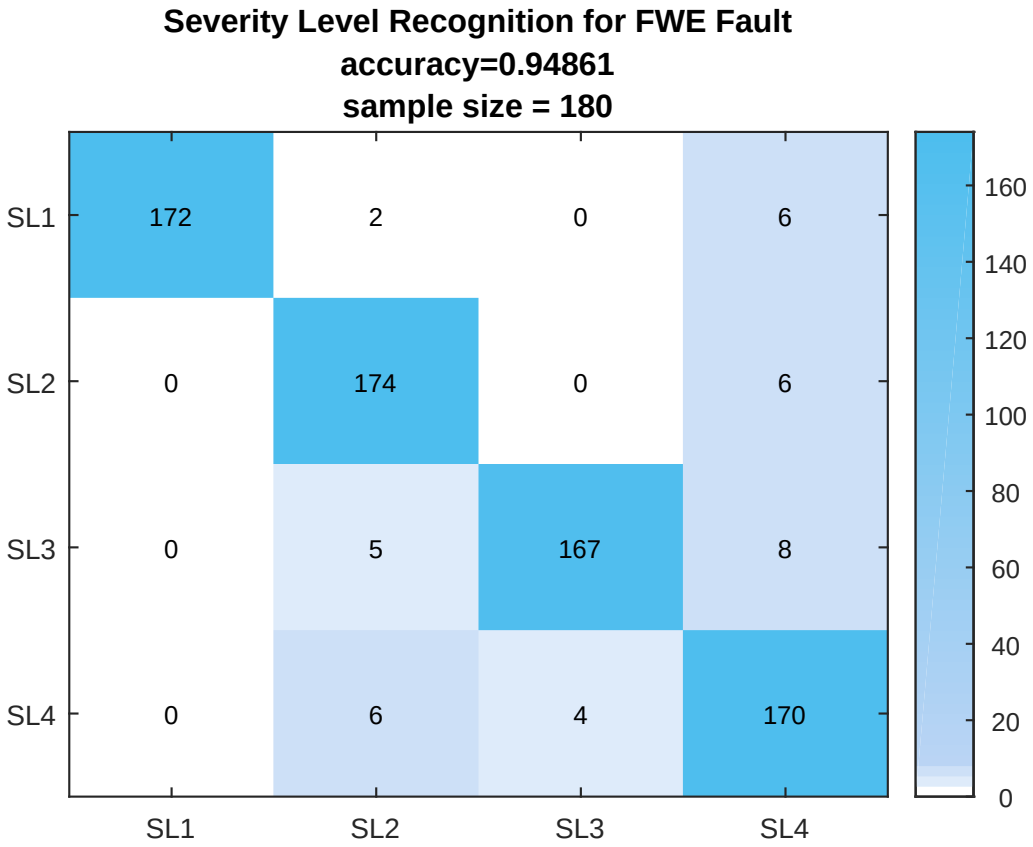


Figure 3.16: Confusion matrix of classifying testing datasets sampled from the FWE fault file by DAFC, which is trained by its pre-defined four severity level.

Chapter 4

FDD with Tree-structured Learning Method

4.1 Introduction

By inspecting the existing works on building data-driven FDD, it is found that they merely formulate the task as a pure fault type classification problem, whereas fault severity levels and their inter-dependence have long been ignored. The proposed DAFC method has been successfully applied in Chapter 3 to detect and diagnose typical chiller faults and identify the corresponding severity levels at two separate stages. However, DAFC tends to generate relatively poor classification results when the number of included classes becomes larger, and it cannot recognize fault type and severity level at a unified step.

In addition, it is widely recognized in machine learning field that prior expert knowledge conveys valuable dependence information which classification methods should consider [85]. When dealing with complex building systems, the number of fault types (classes) is expected to be large, while usually, only a small amount of labeled data for each fault class is available. From a statistical learning perspective,

adopting a "flat" multi-class learning method and ignoring the prior information may result in loss of valuable information, thus leading to degraded performance [84].

The objective of this chapter is to design a novel data-driven FDD method, so as to recognize faulty working conditions and identify the fault type and determine how serious the identified fault is in a unified framework (instead of in two separate stages). Hence, the structured labeling technique is adopted to include the dependence information and describe the severity levels in a large margin learning framework. A Tree-structured Fault Dependence Kernel (TFDK) method is driven and a corresponding on-line learning algorithm is developed for streaming data. As an improvement of traditional classification methods, TFDK presents better FDD accuracy by encoding tree-structured fault dependence in its feature mapping and taking regularized misclassification cost as its learning objective. Also, TFDK is applied to the FDD for a 90-ton centrifugal water-cooled chiller which is introduced in Chapter 3. Experimental results show that compared to previous data-driven methods, TFDK can greatly improve the FDD performance as well as recognize the fault severity levels with high accuracy.

The remaining part of this chapter is arranged as follows. Section 4.2 presents the formulation of structured fault dependence information in the building cooling system. The derivation of TFDK which is based on structured building FDD formulation is given in Section 4.3. Section 4.4 describes how to pre-process the chiller sensor data. Section 4.5 presents the FDD results by TFDK and compares TFDK with the state-of-the-art methods. Section 4.6 summaries this chapter.

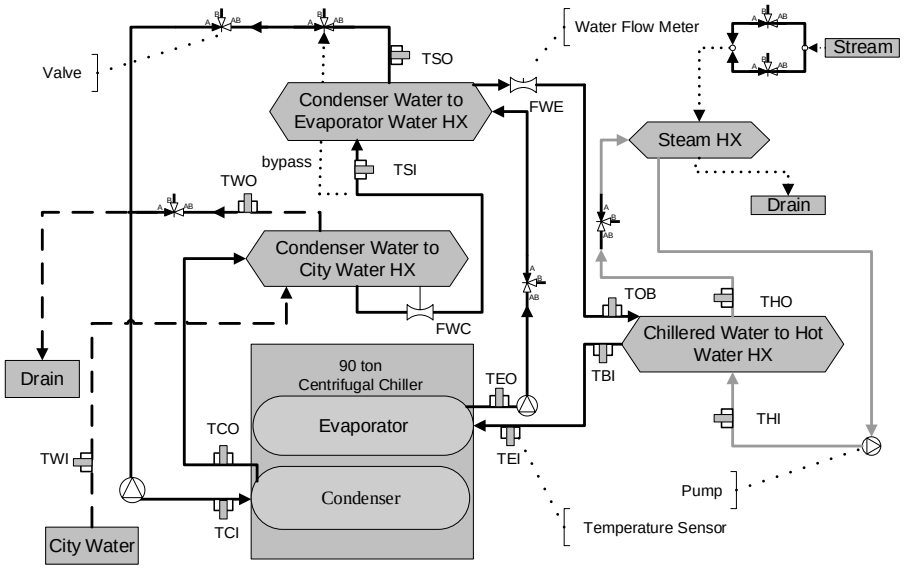


Figure 4.1: Schematic of the cooling system test facility and sensors mounted in the related water circuits.

4.2 Problem Statement

4.2.1 Cooling System with Centrifugal Water-cooled Chiller

In this chapter, the FDD application is also conducted on the 90-ton centrifugal water-cooled chiller which has been described in Chapter 3. As shown in Figure 3.1, a typical centrifugal chiller system consists of: evaporator, compressor, condenser, economizer, motor, pumps, fans, distribution pipes, etc. The refrigerant flow path is depicted in Section 3.2.1, which will not be repeated here. The following paragraph introduces how the cooling system is controlled.

In order to minimize the energy consumption of cooling systems, the entering condenser water temperature set-point should be as low as possible. At the same time, it should be at or above the lowest temperature attainable by cooling tower at certain (wet-bulb) air temperature to avoid wasting fan energy for saturated value. The chilled water supply (leaving evaporator water) temperature is maintained at its set-point by regulating the cooling coil inlet valve position. The valve motor opens

or closes to tune the valve position by a feedback controller which maintains the pre-set chilled water supply temperature based on the difference between the set-point and measured temperature value [102]. There are a great number of sensors implemented within cooling systems in purpose of monitoring and controlling. In order to detect and diagnose typical faults, analysis of sensor measurements for the most essential variables of cooling system is included in this study. In this chapter, 24 essential variables are analyzed according to [74] and [92]. They are common parameters for controlling and monitoring in cooling system, as listed in Table 4.1. Also, a part of sensors mounted within the cooling system is illustrated in Figure 4.1.

4.2.2 Tree Structure Formulation

As introduced in Chapter 3, a large number of possible faults and failures have been identified by the ASHRAE RP-1043, while not all of them would be practical for further examination as part of the FDD scheme [33]. The faults chosen for experimental testing are expected to be able to be detected and diagnosed by monitoring the thermodynamic states of the chiller. Based on how often one fault occurs and how much economical loss it causes, seven typical faults are picked as our research content (the same as that in Chapter 3):

- Condenser Fouling (CF)
- Excess Oil (EO)
- Reduced Condenser Water Flow Rate (FWC)
- Reduced Evaporator Water Flow Rate (FWE)
- Non-Condensable in the Refrigerant (NC)
- Refrigerant Leak/undercharge (RL)

Table 4.1: Definitions of 24 essential variables in a typical cooling system.

No.	Label	Description	Units
1	TEI	Temperature of entering evaporator water	F
2	TEO	Temperature of leaving evaporator water	F
3	TCI	Temperature of entering condenser water	F
4	TCO	Temperature of leaving condenser water	F
5	kW	Compressor motor power consumption	kW
6	FWC	Condenser water flow rate	gpm
7	FWE	Evaporator water flow rate	gpm
8	TEA	Evaporator approach temperature	F
9	TCA	Condenser approach temperature	F
10	TRE	Refrigerant temperature in evaporator	F
11	PRE	Pressure of refrigerant in evaporator	psig
12	TRC	Refrigerant temperature in condenser	F
13	PRC	Pressure of refrigerant in condenser	psig
14	TRC_sub	Subcooling temperature	F
15	T_suc	Refrigerant suction temperature	F
16	Tsh_suc	Refrigerant suction superheat temperature	F
17	TR_dis	Refrigerant discharge temperature	F
18	Tsh_dis	Refrigerant discharge superheat temperature	F
19	P_lift	Pressure lift across compressor	F
20	TO_sump	Temperature of oil in sump	F
21	TO_feed	Temperature of oil feed	F
21	PO_feed	Pressure of oil feed	F
22	TWCD	Condenser temperature	F
24	TWED	Evaporator temperature	F

- Refrigerant Overcharge (RO)

This chapter aims to distinguishing the seven faulty working conditions from the normal working condition for a typical cooling system as well as recognize corresponding severity levels in a unified FDD framework (instead of in two separate stages). Conventional FDD methods which assign the faults and their severity levels

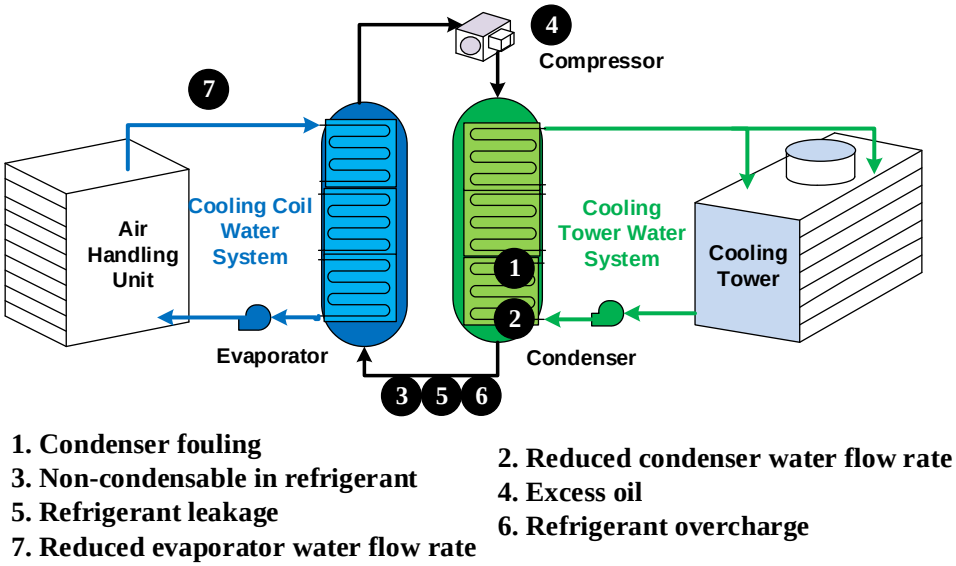


Figure 4.2: Seven typical faults and their locations in the cooling system. Faults 1 and 2 occur in the cooling tower water circle; faults 3, 5 and 6 occur in the refrigerant circle; fault 4 occurs in the compressor; fault 7 occurs in the cooling coil water circle.

with plain labels and formulate the FDD task as pure multiple classification problem. Here,the fault dependence information is included into the feature mapping, and the fault types as well as their severity levels are encoded with tree-structured labels. Next paragraph introduces the tree-structured fault dependence information.

In the light of the expert knowledge about the cooling system configuration, the chosen faults occur in different places within the cooling system, which leads to structured inter-class relationships. As shown in Figure 4.2, chiller water flows through the evaporator pipes and the cooling coil in Air Handling Unit. Therefore the FWE fault, which occurs in the cooling coil water circuit, is relatively not closely related to other faults that occur in other components or subsystems. Similarly, the EO fault which occurs in the compressor motor oil tank is also relatively not closely related to other faults. The NC fault and the RL/RO fault are correlated since they are relevant to the refrigerant. The CF fault, which means condenser pipes are partly blocked, and the FWC fault share the closest correlation because they locate

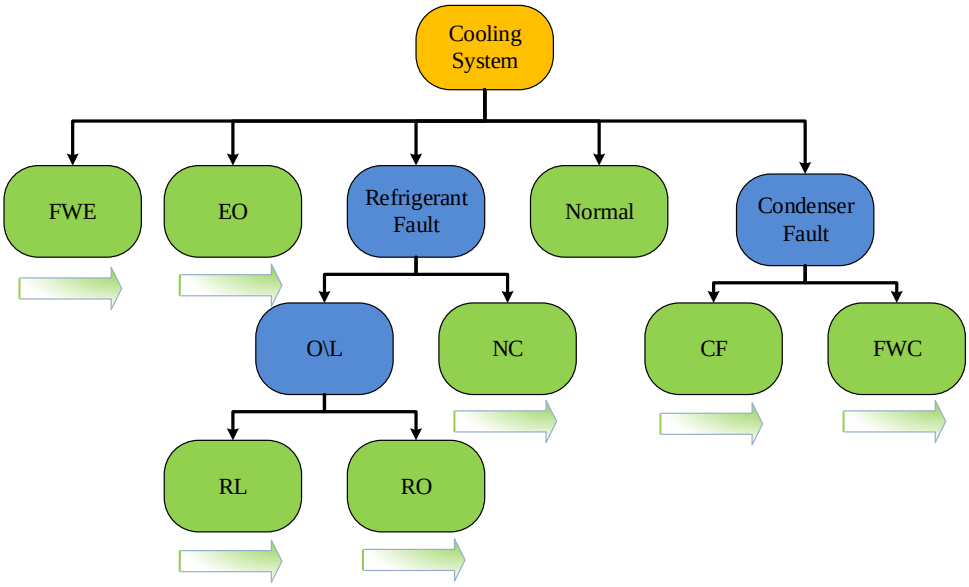


Figure 4.3: Chiller faults with tree-structured labeling. Gradient arrows represent severity levels under each fault type.

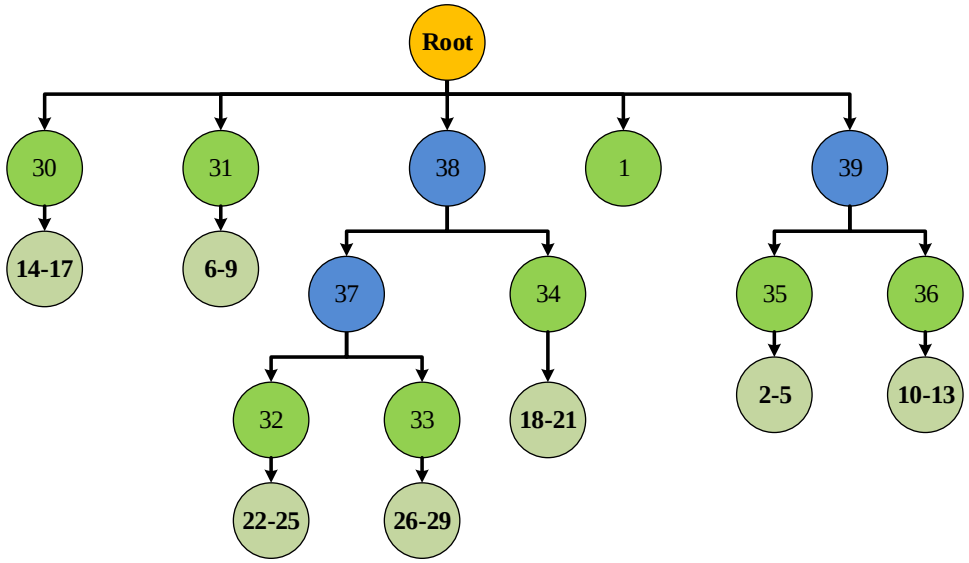


Figure 4.4: Structured labels as a tree which encodes typical chiller faults and corresponding severity levels.

in the cooling tower water circuit and will influence the condenser performance in the first place once occur. On the basis of those prior expert knowledge, one can describe the relationship among the faults and their severity levels with a "tree",

where different fault types as well as the normal condition are described as the branch nodes (non-leaf nodes) and severity levels for one fault are regarded as the leaf nodes rooted from the same parent node. The "tree" is depicted in Figure 4.3, where the gradient arrows represent severity levels under each fault type.

4.3 TFDK Algorithm and Convergence Argument

4.3.1 Feature Mapping

This section introduces the feature mapping which incorporates the prior knowledge about faults and severity level dependence. To begin with, let $\{(x_i, y_i)\}_{i=1}^{n}$ be a set of labeled training data, where $x_i \in \mathcal{R}^d$ denotes the i^{th} record of sensor measurements (d streams) under system condition $y_i \in \mathcal{Y} \triangleq \{1, ..., q\}$ (Detailed data description is presented in Section IV). The tree-structured relationships between chiller faults depicted in Figure 4.3 can be encoded as Figure 4.4. Each node, including the leaves for severity levels but except the root, is numbered with an integer $k \in \{1, 2, \cdots, s\}$. In this case, nodes $1 - 29$ are classification categories, the normal situation is encoded as node 1, and the seven faults as well as their four severity levels are encoded as nodes $2 - 29$. Nodes $30 - 39$ are intermediate nodes that represent the fault dependence. Next, to incorporate the tree-structured information in each data sample, an attributes reweighing vector $\Lambda(y) \in \mathcal{R}^s$ and the transformation $\Phi : \mathcal{R}^d \to \mathcal{R}^{d \times s}$ are considered according to [85], such that

$$\Phi(x, y) = \Lambda(y) \otimes x \tag{4.1}$$

where $\otimes$ denotes a tensor product, i.e. $\Phi(x, y) \in \mathcal{R}^{d \times s}$ is a vector containing all products of coefficients from the first and second vector argument. Writing out $\Phi(x, y)$,

$$\Phi(x,y)=\begin{pmatrix}\lambda_1(y)\times x\\ \lambda_2(y)\times x\\ \cdots\\ \lambda_s(y)\times x\end{pmatrix} \tag{4.2}$$

in which the attributes reweighing vector is defined as

$$\lambda_z(y)=\begin{cases}v_z, & \text{if}\, z\preceq y\\ 0, & \text{otherwise}\end{cases} \tag{4.3}$$

where the relation $\preceq$ denotes that a node z is y or the ancestor of y. The re-weighting parameter $v_z\geq 0$ could be used to include the different influence of node z on node y. In the simplest case it can be set to 1, and λ_z becomes an indicator function. In a more refined configuration, one can set v_z to a positive number that reflects the depth of node z in the tree.

Based on the above transformation one normal class and four severity levels for each of the seven faults are numbered as 29 categories and their dependence constitutes the additional 10 parent nodes in the tree. For example, the labeling and the transformation for level 1 of the RL fault is

$$\Lambda(22)=[0,\cdots,v_{22},\cdots,v_{32},\cdots,v_{37},v_{38},0]^T$$
$$\Phi(x,22)=[0,\cdots,v_{22}x,\cdots,v_{32}x,\cdots,v_{37}x,v_{38}x,0]^T$$

With the feature mapping, consider a general version of discriminant functions F for classification purpose,

$$F(x,y;\mathbf{w})\triangleq\langle\mathbf{w},\Phi(x,y)\rangle \tag{4.4}$$

For simplicity, let $\langle\mathbf{w},\Phi(x,y)\rangle=\langle\mathbf{w}_y,x\rangle$. It is a straightforward consequence of

the linearity of Eq. (4.4) to show that one can re-write F as an additive superposition of linear discriminant as follows,

$$F(x, y; \mathbf{w}) = \sum_{z=1}^{s} \lambda_z(y) \langle \mathbf{w}_z, x \rangle \tag{4.5}$$

where $\mathbf{w}_z \in \mathcal{R}^d$ is a weight vector associated with the rth class attribute. As a concrete example, the discriminant function for node 22 in Figure 4.4 is

$$\langle \mathbf{w}, \Phi(x, 22) \rangle = \langle w_{22}, x \rangle + \langle w_{32}, x \rangle + \langle w_{37}, x \rangle + \langle w_{38}, x \rangle$$

4.3.2 TFDK Learning Method

The learning objective is to find optimal parameters $\mathbf{w}$ for the classification function f, which can be written as,

$$f(x; \mathbf{w}) \triangleq \arg\max_{y \in Y} F(x, y; \mathbf{w}) \tag{4.6}$$

This chapter adopts a large margin learning formulation [103]. Firstly the multi-class margin of a data sample (x_i, y_i) with respect to a parameterization $\mathbf{w}$ can be defined as

$$\gamma_i \triangleq F(x_i, y_i; \mathbf{w}) - \max_{y \neq y_i} F(x_i, y; \mathbf{w}) \tag{4.7}$$

Consider a category dependent cost $\Delta(y_i, y)$ for misclassifying y_i as y (which is clarified in subsection 4.4.1, and interested readers can refer to [104] for more information), one arrives at the following L_2 regularized soft-margin learning objective

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i \tag{4.8}$$

Algorithm 4.1 On-line Update Algorithm

Input (x_{t+1}, y_{t+1})
$S_{t+1} \Leftarrow \emptyset$
while $S_1 ... S_{t+1}$ still change **do**
  for $\imath = t+1 : -1 : 1$ **do**
    $\mathbf{w} = \sum_{j=1}^{t+1} \sum_{y' \in S_j} \alpha_{jy'} \delta\Phi_j(y')$
    $\mathrm{H}(y) = (1 - \langle \mathbf{w}, \delta\Phi_i(y) \rangle) \Delta(y_i, y)$
    $y^* = \arg\max_y H(y)$
    $\xi_i = \max_{y \in S_i} \{H(y)\}$
    if $H(y^*) > \xi_i + \varepsilon$ **then**
      $S_i \leftarrow S_i \cup \{y^*\}$
      $\alpha_S \leftarrow$ Solve dual with S
    end if
  end for
end while
Output $S_1' ... S_t', S_{t+1}'$

$$\text{s. t.} \begin{cases} \xi_i \geq 0 & \\ & \forall\, i \\ \gamma_i(\mathbf{w}) \geq 1 - \frac{\xi_i}{\Delta(y_i, y)} & \end{cases} \tag{4.9}$$

where C is a hyper-parameter that tunes the margin loss penalty.

According to Eq. (4.4) and Eq. (4.7)

$$\begin{aligned} \gamma_i(\mathbf{w}) &= \langle \mathbf{w}, \Phi(x_i, y_i) \rangle - \max_{y \neq y_i} \langle \mathbf{w}, \Phi(x_i, y) \rangle \\ &\leq \langle \mathbf{w}, \Phi(x_i, y_i) \rangle - \langle \mathbf{w}, \Phi(x_i, y) \rangle \, (\forall y \neq y_i) \end{aligned} \tag{4.10}$$

Letting $\delta\Phi_i(y) \triangleq \Phi(x_i, y_i) - \Phi(x_i, y)$, for $\forall i, \forall y \neq y_i$, one can get

$$\langle \mathbf{w}, \delta\Phi_i(y) \rangle \geq \gamma_i(\mathbf{w}) \geq 1 - \frac{\xi_i}{\Delta(y_i, y)} \tag{4.11}$$

Thus the second constraint of Eq. (4.9) can be rewritten as

$$\langle \mathbf{w}, \delta\Phi_i(y) \rangle - 1 + \frac{\xi_i}{\Delta(y_i, y)} \geq 0 \tag{4.12}$$

The dual formulation of the above primal problem with the Lagrangian multiplier method [105] is

$$\begin{aligned}\mathcal{L}(\mathbf{w},\xi,\alpha,\eta) &= \tfrac{1}{2}\|\mathbf{w}\|^2 + \tfrac{C}{n}\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\eta_i\xi_i \\ &- \sum_{i=1}^{n}\sum_{y\neq y_i}\alpha_{iy}\left(\langle\mathbf{w},\delta\phi_i(y)\rangle - 1 + \tfrac{\xi_i}{\Delta(y_i,y)}\right)\end{aligned} \tag{4.13}$$

Computing the derivations of $\mathcal{L}$ with respect to the primal variables by KKT conditions [106] results in

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n}\sum_{y\neq y_i}\alpha_{iy}\delta\Phi_i(y) \tag{4.14}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = 0 \Rightarrow \eta_i = \frac{C}{n} - \sum_{y\neq y_i}\frac{\alpha_{iy}}{\Delta(y_i,y)} \tag{4.15}$$

Plugging $\mathbf{w}$ and η_i into Eq. (4.13), one can get

$$\mathcal{L} = -\tfrac{1}{2}\sum_{i,j}\sum_{y\neq y_i,y'\neq y_j}\alpha_{iy}\alpha_{jy'}\langle\delta\Phi_i(y),\delta\Phi_j(y')\rangle + \sum_{i=1}^{n}\sum_{y\neq y_i}\alpha_{iy} \tag{4.16}$$

Since Eq. (4.8) is equivalent to $\min_{\alpha} -\mathcal{L}$, with the condition that the primal variables are non-negative, one can get

$$\eta_i = \frac{C}{n} - \sum_{y\neq y_i}\frac{\alpha_{iy}}{\Delta(y_i,y)} \geq 0 \tag{4.17}$$

Thus the primal-dual transition of the soft-margin learning objective is

$$\begin{aligned}\min_{\alpha}\ &\frac{1}{2}\sum_{i,j}\sum_{\substack{y\neq y_i\\ y'\neq y_j}}\alpha_{iy}\alpha_{jy'}\langle\delta\Phi_i(y),\delta\Phi_j(y')\rangle - \sum_{i}\sum_{y\neq y_i}\alpha_{iy} \\ &\text{s. t.}\begin{cases}\alpha_{iy}\geq 0 \quad \forall\, i, \quad \forall\, y\neq y_i \\ \sum_{y\neq y_i}\frac{\alpha_{iy}}{\Delta(y_i,y)} \leq \frac{C}{n} \quad \forall\, i\end{cases}\end{aligned} \tag{4.18}$$

Notice that the dual form only involves the inner product [107] of $\delta\Phi_i(y)$ and $\delta\Phi_i(y')$, which admits direct calculation as follows

$$\langle \delta\Phi_i(y), \delta\Phi_j(y') \rangle = \langle \Lambda(y) - \Lambda(y_i), \Lambda(y') - \Lambda(y_j) \rangle \langle x_i, x_j \rangle \tag{4.19}$$

Hence, define the Tree-structured Fault Dependence Kernel (TFDK) as

$$K_{(i,y)(j,y')} = \begin{cases} 0 & \text{if } y = y_i \text{ or } y' = y_j \\ \langle \delta\Phi_i(y), \delta\Phi_j(y') \rangle & \text{otherwise} \end{cases} \tag{4.20}$$

where $K_{(i,\cdot)(j,\cdot)}$ is a $|\mathcal{Y}| \times |\mathcal{Y}|$ matrix and constitute the (i,j)th block of the overall kernel matrix K. It's straightforward to check that K is positive semi-definite, and thus the learning problem belongs to a convex quadratic program (QP). Although various methods exist in literature to solve convex QP, for the problem at hand, the presence of many linear constraints and the requirement for learning with streaming data motivate us to design an online active set algorithm. Key update steps for the algorithm is summarized in Algorithm 4.1, where one can simply use empty set for the initialization of active sets. The convergence argument is included in the next subsection.

4.3.3 Convergence Argument of the On-line Update Algorithm

Our notation in this paper follows the large margin formulation in [103]. Interested readers are referred to [108,109] for more background information.

Step One

To prove that sufficient improvement can be obtained for the objective function Eq. (4.18) in each iteration, firstly consider the dual formulation in Eq. (4.16) as

$$J(\alpha) = -\frac{1}{2}\alpha^T K\alpha + n^T\alpha \tag{4.21}$$

Define β as the update step size and τ as the update direction. Then

$$\begin{aligned}
&\delta J(\beta) \triangleq J(\alpha+\beta\tau) - J(\alpha) \\
&= -\tfrac{1}{2}\tau^T K\beta\tau - \tfrac{1}{2}\tau^T\beta^T K\alpha - \tfrac{1}{2}\tau^T\beta^T K\beta\tau + \beta n^T\tau \\
&= -\beta\alpha^T K\tau - \tfrac{1}{2}\beta^2\tau^T K\tau + \beta n^T\tau
\end{aligned} \tag{4.22}$$

Thus by denoting $\langle \nabla J(\alpha), \tau\rangle = n^T\tau - \alpha^T K\tau$

$$\begin{aligned}
&\tfrac{\partial\delta J(\beta)}{\partial\beta} = -\beta\tau^T K\tau - \left(\alpha^T K\tau - n^T\tau\right) = 0 \\
&\Rightarrow \beta^* = \tfrac{n^T\tau - \alpha^T K\tau}{\tau^T K\tau} = \tfrac{\langle \nabla J(\alpha), \tau\rangle}{\tau^T K\tau}
\end{aligned} \tag{4.23}$$

Now substitute β^* into $\delta J(\beta)$

$$\delta J(\beta^*) = \frac{1}{2}\frac{\left(\langle \nabla J(\alpha), \tau\rangle\right)^2}{\tau^T K\tau} \triangleq \frac{1}{2}\cdot\frac{D_{\alpha\tau}^2}{\tau^T K\tau}\left(D_{\alpha\tau} = \langle \nabla J(\alpha), \tau\rangle\right) \tag{4.24}$$

Since β is within a bounded section $0 \le \beta \le B$

(I) If $\beta^* \le B$, then

$$\delta J(\beta^*) = \frac{1}{2}\cdot\frac{D_{\alpha\tau}^2}{\tau^T K\tau}; \tag{4.25}$$

(II) If $\beta^* \ge B$, since J is Convex Quadratic

$$\begin{aligned}
&\delta J(\beta^*) \ge \delta J(B) \\
&= B\left(n^T\tau - \alpha^T K\tau\right) - \tfrac{1}{2}B^2\tau^T K\tau \\
&= BD_{\alpha\tau} - \tfrac{B^2}{2}\cdot\tau^T K\tau
\end{aligned} \tag{4.26}$$

Note that $\tau^T K\tau > 0$ and $B \leq \beta^* = \frac{D_{\alpha\tau}}{\tau^T K\tau}$, then

$$\delta J(\beta^*) \geq B\left(D_{\alpha\tau} - \frac{1}{2}\cdot\frac{D_{\alpha\tau}}{\tau^T K\tau}\cdot\tau^T K\tau\right) = \frac{1}{2}BD_{\alpha\tau} \tag{4.27}$$

Hence, from (I) and (II), one can get

$$\max_{0\leq\beta\leq B}\delta J(\beta) \geq \tfrac{1}{2}\min\left\{\tfrac{D_{\alpha\tau}^2}{\tau^T K\tau}, BD_{\alpha\tau}\right\} = \tfrac{D_{\alpha\tau}}{2}\min\left\{\tfrac{D_{\alpha\tau}}{\tau^T K\tau}, B\right\} \tag{4.28}$$

Step Two

At each step, assume (x_i, y_i) is newly added. Optimize α_{iy} in Eq. (4.18) with the upper bound

$$\alpha_{iy} \leq \Delta(y_i, y)\frac{C}{n} \triangleq B \tag{4.29}$$

Consider the dual formulation in Eq. (4.16). It is easy to see

$$\frac{\partial\mathcal{L}(\alpha)}{\partial\alpha_{iy}} = 1 - \sum_{j,y}\alpha_{jy'}K_{(i,y)(j,y')} = 1 - \langle\mathbf{w}, \delta\Phi_i(y')\rangle \tag{4.30}$$

Since $H(y) = (\mathbf{w}, \delta\Phi_i(y'))\,\Delta(y_i, y)$ and $H(y^*) \geq \xi_i + \varepsilon$, then

$$\frac{\partial\mathcal{L}(\alpha)}{\partial\alpha_{iy}} \geq \frac{\xi_i + \varepsilon}{\Delta(y_i, y)} \geq \frac{\varepsilon}{\Delta(y_i, y)}\,(\Delta(y_i, y) > 0, \xi_i \geq 0) \tag{4.31}$$

Assuming the step size $\tau = 1$, one can obtain

$$D_{\alpha\tau} = n^T\tau - \alpha^T K\eta = 1 - \alpha^T K = \frac{\partial\mathcal{L}(\alpha)}{\partial\alpha_{iy}} \tag{4.32}$$

Substituting Eq. (4.29) and Eq. (4.32) to the result of Step One (Eq. (4.28)), one can get

$$\delta\mathcal{L}(\beta) \geq \frac{1}{2}\min\left\{\frac{1}{K}\cdot\frac{\partial\mathcal{L}(\alpha)}{\partial\alpha_{iy}}, \frac{C}{n}\cdot\Delta(y_i, y)\right\}\cdot\frac{\partial\mathcal{L}(\alpha)}{\partial\alpha_{iy}} \tag{4.33}$$

Due to Eq. (4.31)

$$\delta\mathcal{L}(\beta) \geq \tfrac{1}{2}\min\left\{\tfrac{1}{K}\cdot\tfrac{\varepsilon}{\Delta(y_i,y)}, \tfrac{C}{n}\cdot\Delta(y_i,y)\right\}\cdot\tfrac{\varepsilon}{\Delta(y,y_i)} = \tfrac{1}{2}\min\left\{\tfrac{\varepsilon^2}{K[\Delta(y_i,y)]^2}, \tfrac{C\varepsilon}{n}\right\} \quad (4.34)$$

If (x_i, y_i) is already in the active set, the search direction τ could be tuned and with a similar argument. One can obtain

$$\delta\mathcal{L}(\beta) \geq \frac{1}{2}\min\left\{\frac{\varepsilon^2}{4K[\Delta(y_i,y)]^2}, \frac{C\varepsilon}{n}\right\} \quad (4.35)$$

Step Three

By denoting Eq. (4.18) as $L(\alpha)$ and Eq. (4.8) as $P(\mathbf{w})$, based on the Primal-Dual Theory one knows that

$$L(\alpha) \leq \min P(\mathbf{w}) \quad (4.36)$$

Let $\mathbf{w} = 0$, according to Eq. (4.9) (thus $\xi_i \geqslant \Delta(y_i, y)$)

$$P(\mathbf{w}) \triangleq \min \tfrac{1}{2}\|\mathbf{w}\|^2 + \tfrac{C}{n}\sum_{i=1}^{n}\xi_i \geq 0 + \tfrac{C}{n}\sum_{i=1}^{n}\Delta(y_i,y) \quad (4.37)$$

$$L(\alpha) \leq \min P(\mathbf{w}) = \tfrac{C}{n}\sum_{i=1}^{n}\Delta(y_i,y) \geq C\cdot\max_y \Delta(y_i,y) \triangleq C\cdot\Delta_{\max} \quad (4.38)$$

Hence the optimal improvement of $L(\alpha)$ is at most $C\cdot\Delta_{\max}$. For each step the improvement is at least $\frac{1}{2}\min\left\{\frac{\varepsilon^2}{4K\Delta_{\max}^2}, \frac{C\varepsilon}{n}\right\}$ as depicted in Eq. (4.35). Now one can conclude that the algorithm will converge in the following steps

$$\frac{C\cdot\Delta_{\max}}{\frac{1}{2}\min\left\{\frac{\varepsilon^2}{4K\Delta_{\max}^2}, \frac{C\varepsilon}{n}\right\}} = 2\max\left\{\frac{4CK\Delta_{\max}^3}{\varepsilon^2}, \frac{n\Delta_{\max}}{\varepsilon}\right\} \quad (4.39)$$

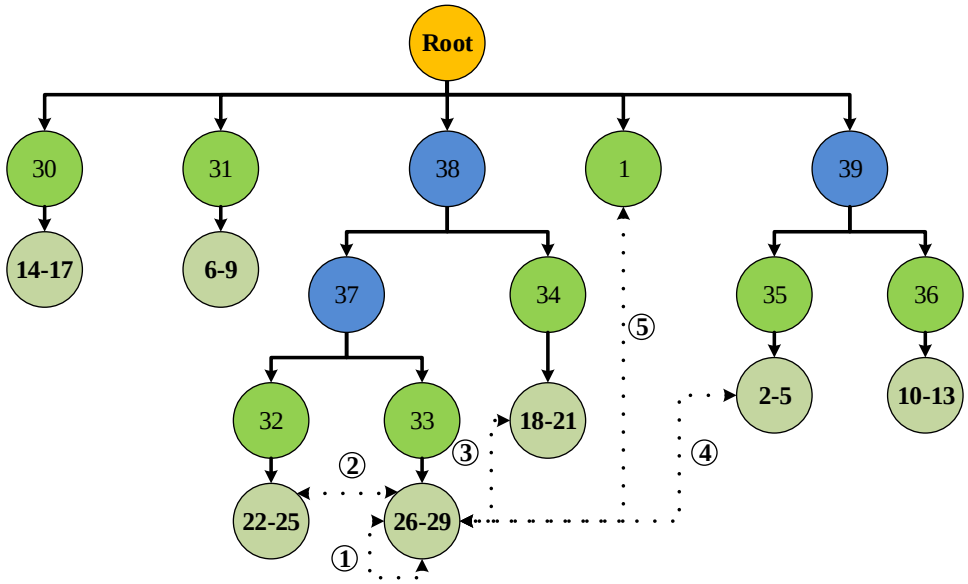


Figure 4.5: Structured labels as a tree for typical chiller faults and corresponding severity levels. Examples of misclassification cost among severity levels and fault types.

4.4 Data Pre-processing

4.4.1 Evaluation Measures

To evaluate the effectiveness of the tree-structured classification method on a more rigorous basis, two measures are employed : testing accuracy and testing cost.

Testing Accuracy

The goal is to estimate the chance that the predictor $f(x)$ is correct on future unseen data, i.e., the generalization performance of the predictor. In this work, the empirical accuracy on a batch testing data set is used as an unbiased estimator. Let sign $[.,.]$ be 1 if the predicted label of one testing data point accorded with its

original label and 0 otherwise, the testing accuracy is

$$Accu\left(f\right)=\frac{1}{n}\sum_{i=1}^{n}\text{sign}\left[f\left(x_i\right),y_i\right] \tag{4.40}$$

$$\text{sign}\left[f\left(x_i\right),y_i\right]=\begin{cases}1 & f\left(x_i\right)=y_i\\ 0 & f\left(x_i\right)\neq y_i\end{cases} \tag{4.41}$$

where $f\left(x_i\right)$ is the predicted label for testing data point x_i as in Eq. (4.6), which represents that the data point is recognized as a certain severity level of one fault type, and y_i is the true label that records the real experiment condition.

Testing Cost

While testing accuracy is an unbiased estimator of classification correctness, it treats all errors equally important, i.e., all types of errors induce the same cost. In practice, however, the seriousness and the consequence of committing different types of errors may vary significantly. In particular based on the tree-structured relationships between different faults in Figure 4.4, the misclassification of one category to another category will cause different losses. In order to incorporate this consideration, the cost of misclassification among severity levels under the same fault type is defined to be the lowest, the cost of misclassification among fault types derived from different parent nodes to be higher, and the cost of recognizing fault as normal to be the highest. Especially, one can assign a cost that is proportional to the node distance in the tree depicted in Figure 4.5. For instance, the cost of misclassification among leaf nodes $26-29$ is 1; and the cost among leaf nodes $26-29$ and $18-21$ is 3. Putting all the defined costs in a cost matrix Δ, the misclassification cost can

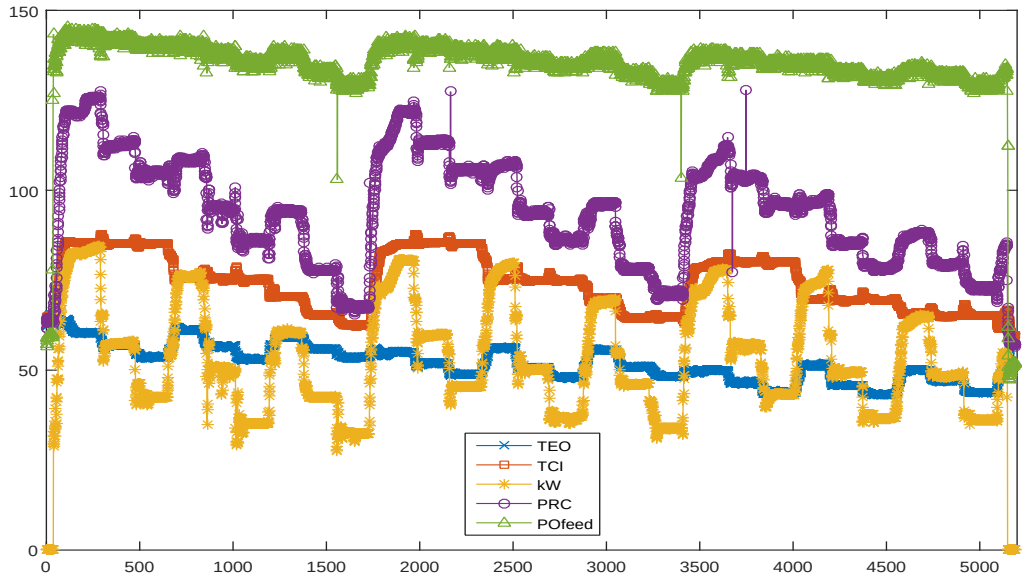


Figure 4.6: Raw data of five variables under normal condition collected by sensors mounted in the cooling system.

be characterized by a loss function as

$$F_{cost}(f) = \frac{\sum_{i=1}^{q}\sum_{j=1}^{q}\Delta_{ij}g\left(y_j, f\left(x_i\right)\right)}{\sum\sum\Delta_{ij}g'\left(y_j, f\left(x_i\right)\right)} \tag{4.42}$$

where $g\left(y_j, f\left(x_i\right)\right)$ is in fact a confusion matrix in which each row represents the number of samples in predicted class while each column represents the samples in actual (true) class, $g'\left(y_j, f\left(x_i\right)\right)$ represents how many testing data points will be classified to category j if testing data from category i is averagely classified to other categories, and Δ_{ij} is the cost of classifying test data point from category i to j ($\Delta_{ij} = 0$ if $i = j$), and here $F_{\cos t}(f)$ is the absolute cost value for the classifier f.

Notice that the misclassification cost is considered as one optimization constraint in Eq. (4.18).

4.4.2 Pre-processing Methods for Chiller Sensor Data

Following the ASHRAE RP-1043, the cooling system studied in this chapter is a typical centrifugal water-cooled chiller system with motor-driven compressor [33]. The experimental chiller fault data has been introduced in Chapter 3, which will not be repeated here. Note that since the sensor measurements are time series, they present periodic patterns mostly due to the on/off (open/close) states of some components in the system. Figure 4.6 shows the raw data of five variables, where periodic patterns and obvious outliers can be viewed. Before applying the tree-structure FDD of the cooling system, raw data is pre-processed by removing periodic patterns and outliers so as to avoid non-negligible side effects caused by confusing patterns and outliers [110].

Outlier Removing

Sensor data for each variable is treated as a column vector and thus obvious outliers can be detected by the Modified Thompson's Tau method [111], which is based on the absolute deviation of each record from the mean of the entire vector. The strength of this method lies in the fact that it takes into account the dataset's standard deviation and average, and provides a statistically determined rejection zone; thus providing an objective method to determine whether a data point is an outlier. The rejection zone is given by

$$\tau = \frac{\mathrm{t}_{\alpha/2}\,(N-1)}{\sqrt{N}\sqrt{N-2+t_{\alpha/2}^2}} \tag{4.43}$$

where $\mathrm{t}_{\alpha/2}$ is the critical value from the Student's t distribution [112], and N is the sample size. The absolute deviation of the data set is

$$\sigma = |(X - mean\,(X))/S| \tag{4.44}$$

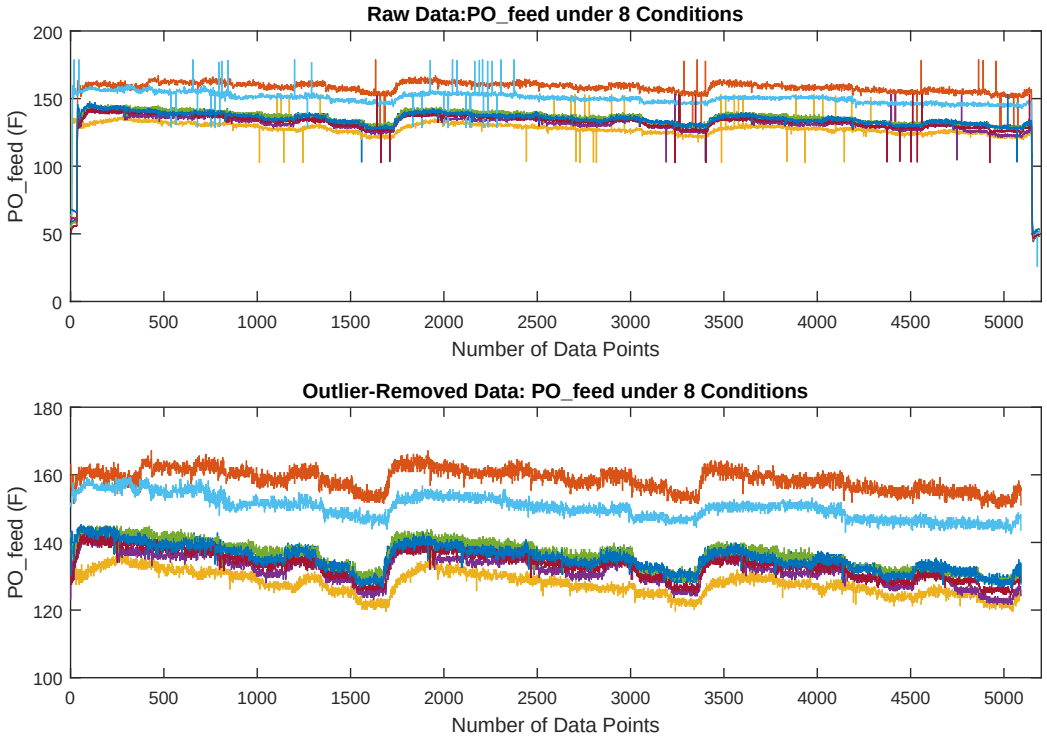


Figure 4.7: Outliers of "pressure of oil feed (PO_feed)" are removed by Thompson Tau method; the raw data is collected under eight working conditions (one normal condition and seven fault conditions).

where S is the sample standard deviation. If $\sigma > \tau$, the data point is an outlier. As shown in Figure 4.7, the variable "pressure of oil feed (PO_feed)" is measured by corresponding sensors under 8 conditions (normal condition and 7 faulty conditions), and the time series sensor measurements are smoother without outliers compared with the raw data.

Periodic Pattern Removing

In this study, Wavelet-based de-noising is utilized to remove the confusing periodic patterns. Wavelet Transform is an infinite set of various transforms $\psi(t)$, and is obtained from a single orthonormal wavelet, called mother wavelet or basic function,

by scaling and shifting (translation). The wavelet series can be defined as

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{4.45}$$

where a and b represent the scale and translation parameters respectively. In the discrete case where the Wavelet Transform can be used for de-nosing, the scale and translation parameters are discretized as $a = 2^m$ and $b = n2^m$. The dilated and translated version of the mother wavelet $\psi(t)$ can be written as:

$$\psi_{a,b}(t) = \psi_{m,n}(t) = 2^{-m/2}\psi\left(2^{-m}t - n\right) \tag{4.46}$$

where m and n denote the scale and translation parameters respectively. Given an original signal $f(t)$, its wavelet coefficients are obtained through the inner product operation:

$$\mathbf{W}(a,b) = \int_{-\infty}^{\infty} \psi^*_{a,b}(t)\, f(t) dt \tag{4.47}$$

where $*$ is the complex conjugate symbol and ψ is the basic function, which can be chosen according to the properties of the given function f. The choice of mother wavelet (e.g. Haar, Daubechies, Coiflets, Symlet, Biorthogonal, etc.) determines the final waveform shape, and here Symlet is chosen as the basic function.

The essence of de-noising using Discrete Wavelet Transform (DWT) is to reduce the noise in the wavelet transform domain [113]. Define the noisy observations as $\mathbf{W} = [w_1, w_2, ..., w_N]$, satisfying

$$\mathbf{W} = \mathbf{f} + \varepsilon \tag{4.48}$$

where $\mathbf{f} = [f_1, f_2, ..., f_N]$ is the desired noise-free signal, and $\varepsilon = [\varepsilon_1, \varepsilon_2, ..., \varepsilon_N]$ is the observation noise. Firstly, DWT is applied to the noisy signal to produce the noisy wavelet coefficients to the level where one can properly distinguish the signal pattern. Then the inversed wavelet transform of filtered wavelet coefficients is applied to

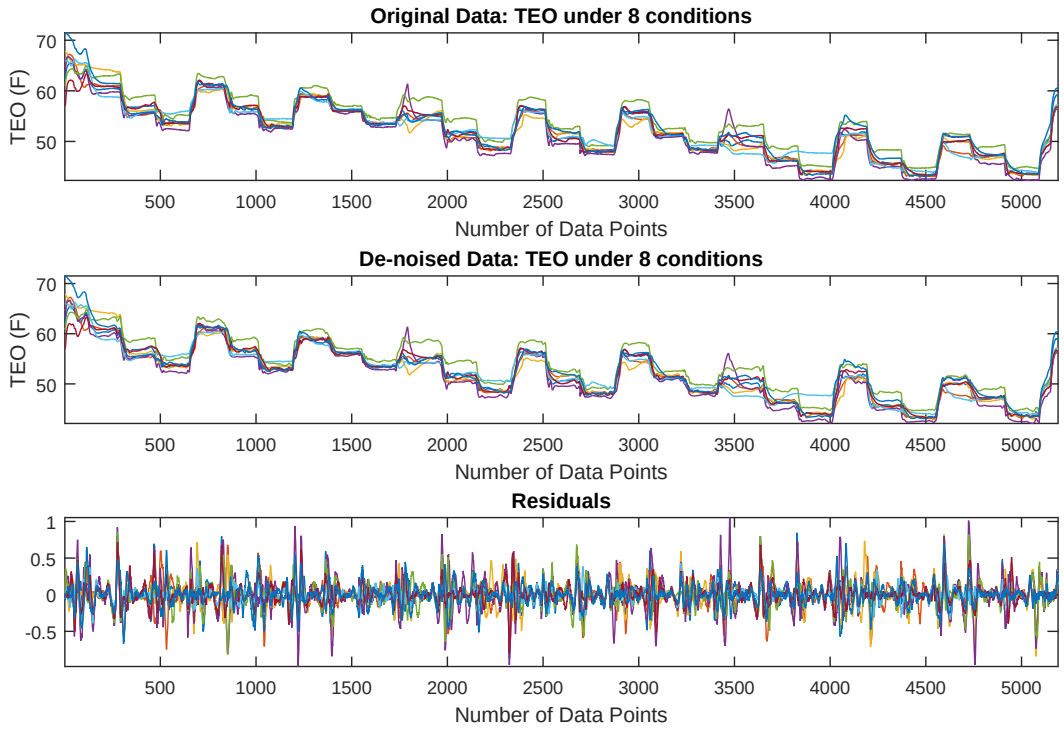


Figure 4.8: Temperature of leaving evaporator water pre-processed by Wavelet de-noising (level=5); the raw data is collected under eight working conditions (one normal condition and seven fault conditions).

obtain a de-noised signal. As shown in Figures 4.8-4.10, the raw data of leaving evaporator water temperature is pre-processed by Wavelet-based De-noising with increasing levels of wavelet decomposing. One can see from those figures that the periodic patterns can be removed from the original raw data when the wavelet decomposition level is relatively high.

4.5 FDD Results and Comparison

4.5.1 Experiment Set-up

A classical "training-testing" procedure is adopted to justify the statistical performance of the proposed FDD framework. The chiller fault data is randomly divided into two parts, one for fitting the TFDK model and the other one for testing the

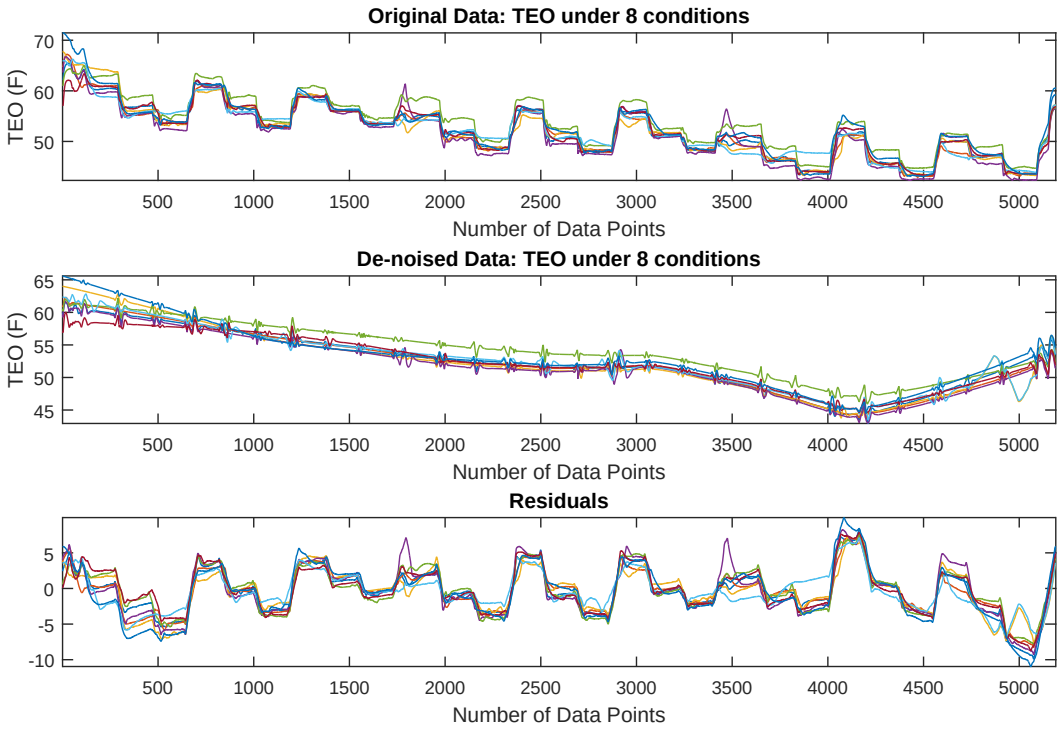


Figure 4.9: Temperature of leaving evaporator water pre-processed by Wavelet de-noising (level=10); the raw data is collected under eight working conditions (one normal condition and seven fault conditions).

attained model on unseen data set. Since labeled data usually has limited availability in practice, the TFDK classifier is trained with various sample sizes to analyze its impact on testing accuracy. Given that the raw data of ASHRAE RP-1043 is collected every 10 seconds, sample size also represents the time duration spent on data collection. For example, within 10 minutes, sensors can collect 60 data samples, each with 24 channels. In this chapter, the classifier is trained with 8 different sample sizes (i.e. 6, 12, 18, 30, 48, 90, 120 and 180). For each configuration, the testing data is randomly chosen from the pre-processed testing data set and testing sample size is 1600 for each fault type (400 for each severity level) and 400 for the normal condition.

Moreover, considering the availability of sensor measurements in a more practical situation, 24 most accessible variables listed in Table 4.1 are chosen as algorithm

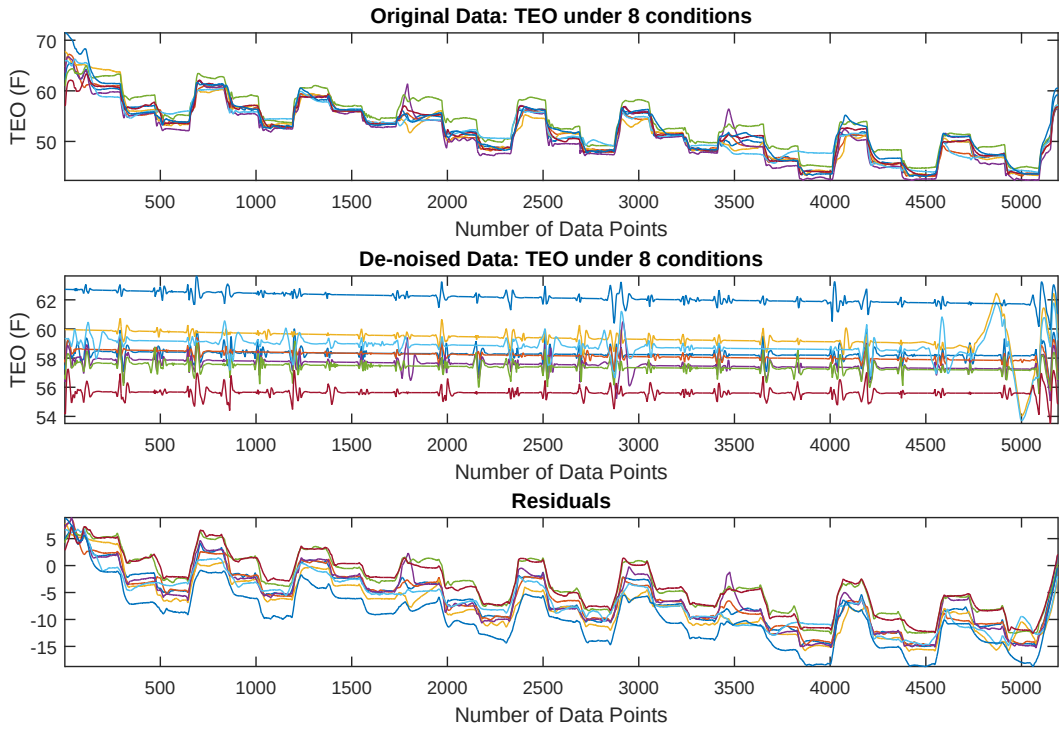


Figure 4.10: Temperature of leaving evaporator water pre-processed by and Wavelet de-noising (level=15); the raw data is collected under eight working conditions (one normal condition and seven fault conditions). It can be viewed that periodic patterns can be removed when the Wavelet decomposition level is relatively high.

input features. If a fault is detected, the desired output by TFDK includes not only normal/fault types, but also 4 severity levels. Data groups of all fault types with four severity levels as well as data collected under normal condition are defined as the training data sets, i.e. 29 categories in total. Those categories are encoded with tree-structured labels as depicted in Figure 4.4.

4.5.2 Comparison of Accuracy and Cost Among Different Methods

TFDK is compared with the state-of-the-art methods, including Multi-class SVM (MSVM) with RBF kernel, Decision Tree (DT), Neural Network (NN), AdaBoost (AB), Quadratic Discriminant Analysis (QDA), and Logistic Regression (LR). Fig-

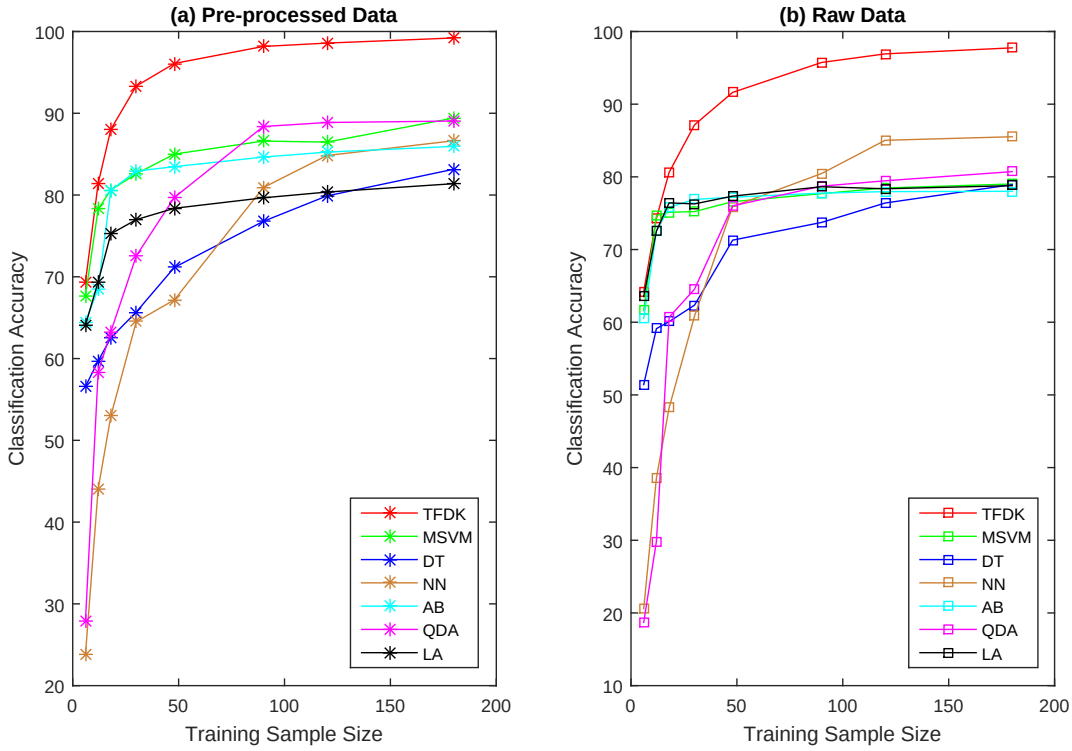


Figure 4.11: Classification accuracy as a function of training sample size by different methods; TFDK generates the highest accuracy. Figure (a) shows the classification accuracy of different methods by data that is pre-processed by de-nosing and outlier removing; figure(b) is the classification accuracy directly with raw data.

ure 4.11 shows the classification accuracy as a function of training sample size for all the methods. In order to demonstrate the effectiveness of de-trending and de-noising, two sets of experiments were conducted with (Figure 4.11 (a)) and without (Figure 4.11(b)) the proposed pre-processing technique. Similarly the results of testing cost as a function of training sample size for all the methods are shown in Figure 4.13. Figures 4.12 and 4.14 also illustrate the accuracy improvement and cost decrease for each considered classification methods (TFDK and other "plain" methods) when the training sample size is 30. More specifically, the advantages of TFDK over the state-of-the-art methods are inspected from the following three aspect:

1. It is seen that TFDK outperforms all the other methods in terms of testing

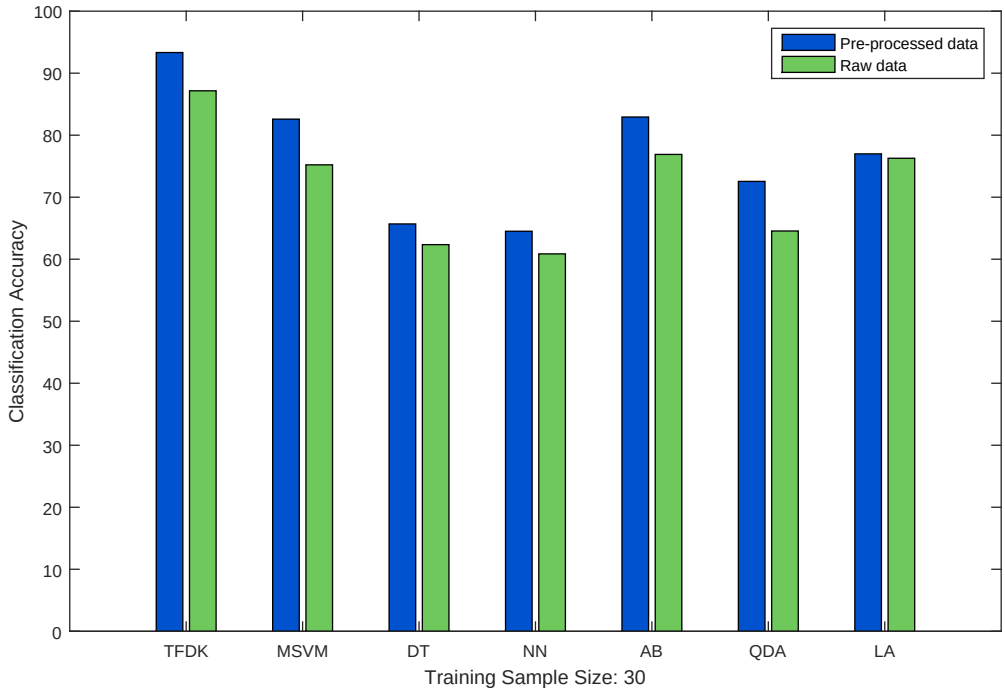


Figure 4.12: Comparison between classification accuracy by pre-processed data and raw data. Data pre-processing helps to improve the classification accuracy.

accuracy and cost under different training configurations. More specifically, TFDK achieves 1.49% to 9.19% accuracy improvement and 10.69% to 75% cost decrease compared to the runner-up method. Note that the enhancement is more significant when the sample size is larger. Although it appears that the improvement is not obvious under small sample size ($\leq$ 12), TFDK has extra advantage of being robust to inter fault type misclassification, as will be revealed later with confusion matrix.

2. As expected, the testing accuracy/cost increases/decreases accordingly with the increment of training samples. For instance, the testing accuracy of TFDK has boosted from 69.64% (6 training samples) to 99.12% (180 training samples); similar trends can be observed for the other methods, which reaffirms the intuition that more training data is beneficial to data-driven FDD.

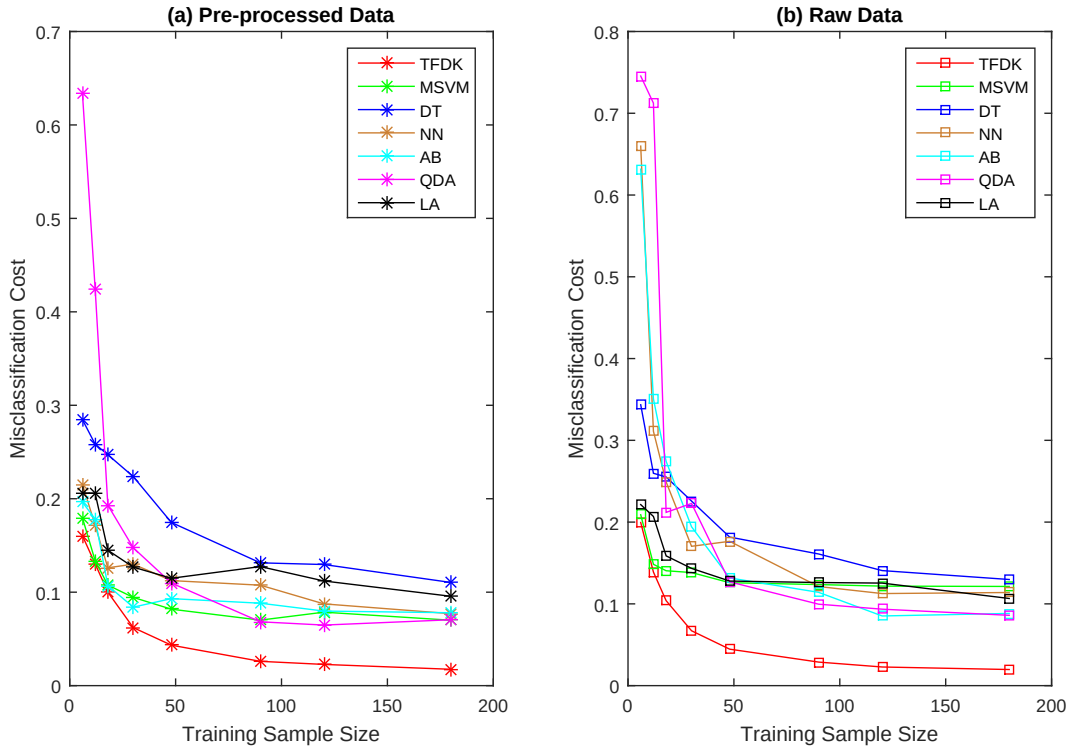


Figure 4.13: Misclassification cost as a function of training sample size by different methods; TFDK generates the lowest cost. Figure (a) shows the misclassification cost of different methods by data that is pre-processed by de-nosing and outlier removing; figure (b) is the misclassification cost directly with raw data.

3. Comparing the two sub-plots (a) and (b) of Figure 4.11 and Figure 4.13, it is viewed that those methods with pre-processed data present better results in general. Specifically, by looking into the case when the training sample size is 30 and comparing the testing accuracy and cost for different methods in Figures 4.12 and 4.14, it is seen that the proposed pre-processing techniques greatly improve the performance of several methods such as TFDK, MSVM and AB.

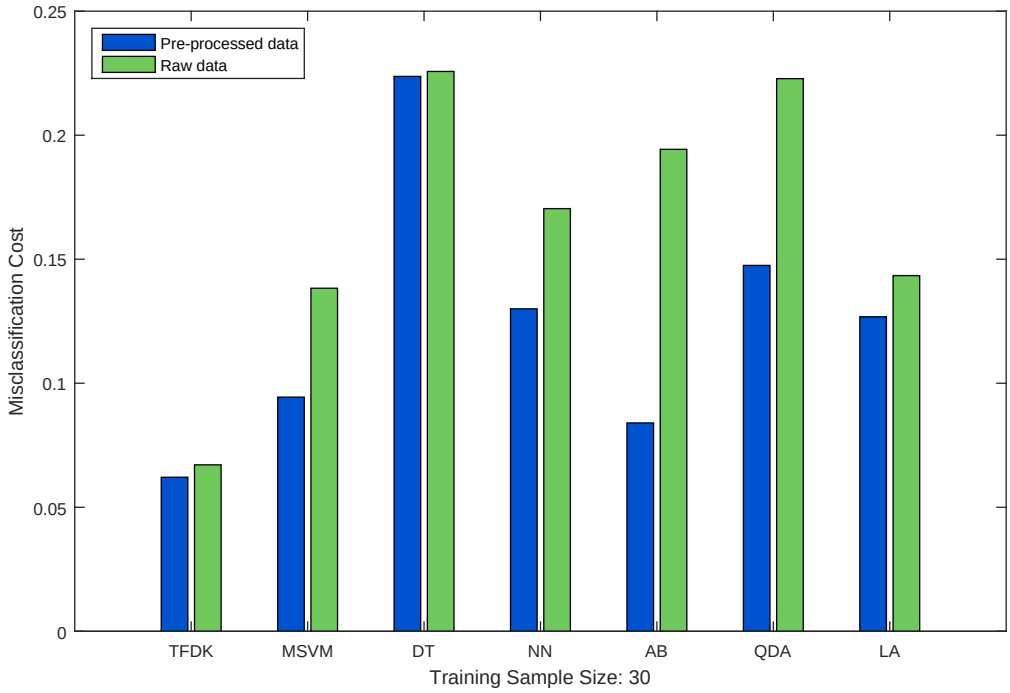


Figure 4.14: Comparison between misclassification cost by pre-processed data and raw data. Data pre-processing helps to reduce the misclassification cost.

4.5.3 Advantages of Incorporating Fault Dependence Tree

To further investigate the benefit of including the prior knowledge of fault dependence, detailed classification results are compared for TFDK and MSVM. The comparative results reflect the effect of tree-structured fault dependence information because TFDK can be viewed as a hierarchical variation of the traditional large margin SVM. Sub-plots (a) and (c) of Figure 4.15 are the confusion matrixes of MSVM and TFDK respectively when the training sample size is 6, which is the smallest training sample size in our test; and sub-plots (b) and (d) of Figure 4.15 are the confusion matrixes for MSVM and TFDK accordingly under the largest training sample size of our test, which is 180.

As mentioned earlier, in the case of small training sample size, TFDK does not bear notable improvement in accuracy compared to MSVM. However, close scrutiny of Figure 4.15 (a) vs. (c) and Figure 4.16 (a) vs. (b) reveals that TFDK presents

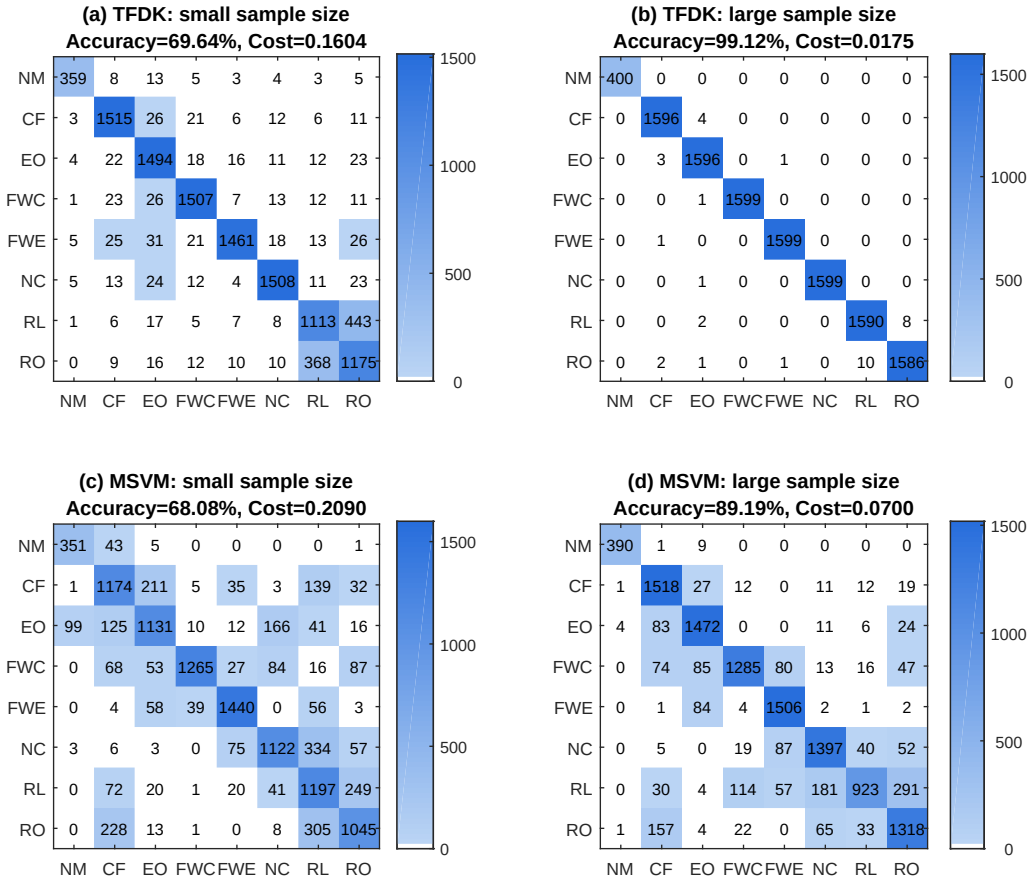


Figure 4.15: Confusion matrix of TFDK and MSVM among fault types under small training sample size and large training sample size, respectively. In (a) and (c), both TFDK and MSVM are trained with small training sample size, where they generate similar classification accuracy, 69.64% and 68.08%. However, TFDK presents little misclassification among fault types. In (b) and (d), TFDK and MSM are trained with relatively large training sample size. TFDK presents very high classification accuracy, while MSVM still presents apparent misclassification.

much lower misclassification rate among fault types. Figure 4.16 (a) and (b) show that the detailed prediction assignment for EO fault by TFDK and MSVM. Indeed, the errors of TFDK mainly occur among severity levels while the correct fault types have already been assigned (Figure 4.16 (a)). On the other hand, quite a few errors committed by MSVM occur among different fault types (Figure 4.16 (b)).

In the case of larger training sample size, the classification accuracy of MSVM is 89.19% which appears relatively high from the FDD perspective, nevertheless Figure 4.15 (d) presents that MSVM still generates significant misclassification rate among

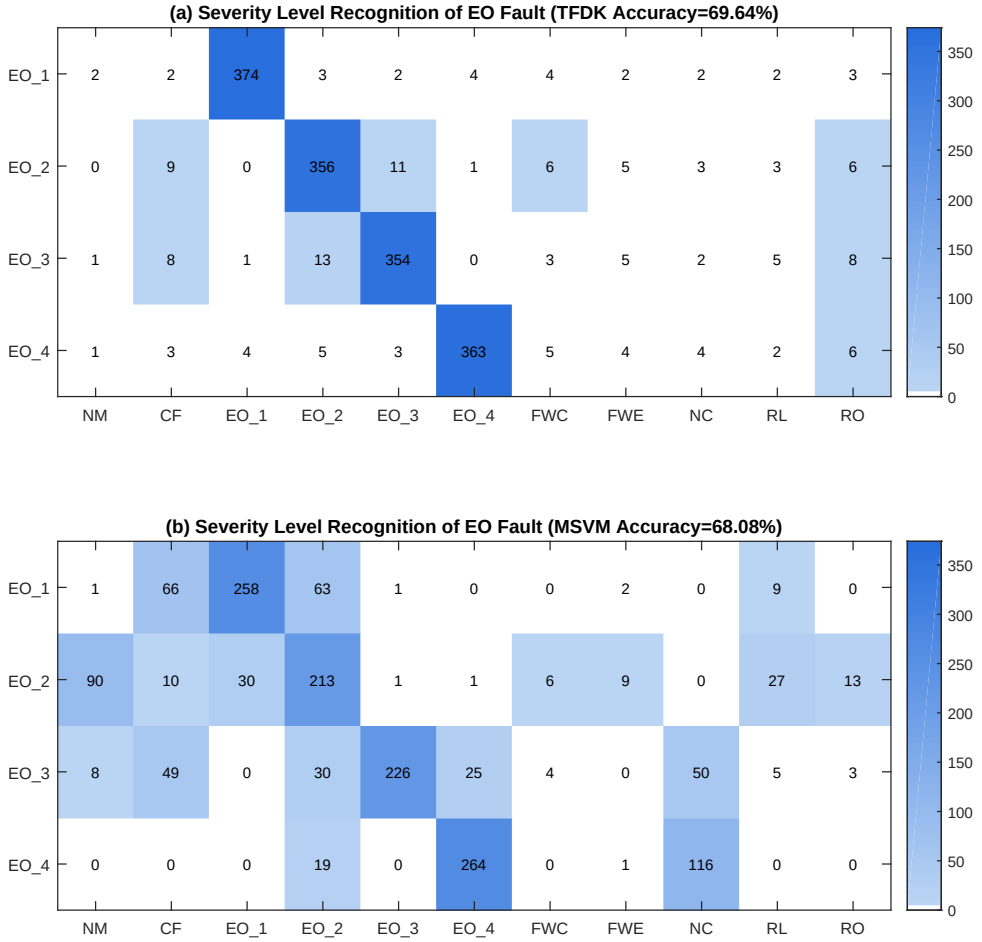


Figure 4.16: Confusion matrix of TFDK and MSVM for the severity levels of the EO fault under small training sample size. To inspect the severity level identification rates of EO fault under small training sample size, (a) shows that most of TFDK's misclassification occurs among its own four severity levels; while (b) shows that MSVM presents misclassification to both its own four severity levels and other fault types.

fault types under the large training sample size situation. Among all the methods, when the training sample size is 180 the proposed TFDK behaves with extremely high classification accuracy (99.12%) and very low misclassification cost, which is shown in Figure 4.15 (b).

4.6 Conclusion

In this chapter, a novel data-driven FDD method is proposed and a corresponding on-line learning algorithm for streaming data is developed. Unlike traditional classification methods which give each category plain labels and ignore the relationship among different faults, the hierarchical kernel learning method assigns tree-structured labels to the faults. To be specific, the fault dependence information is encoded as a "tree" and the severity levels are described as child nodes of each fault type rather than treating them as independent classes. With that, the prior knowledge of the system and the task of identifying fault severity levels are dealt with in a unified framework. Verified by detecting and diagnosing typical chiller faults, TFDK is proven to be superior to the state-of-the-art methods.

Chapter 5

FDD with Feature Selection Method

5.1 Introduction

Through last two chapters, it is known that direct utilization of raw measurements, which have noisy and non-informative variables, may lead to degraded classification performance [88,89]. In Chapter 3, LDA algorithm has been applied to tackle the "curse of dimensionality". In Chapter 4, 24 primary variables are chosen according to their availability and the control requirement in real systems. However, the LDA-projected space (which is a linear combination of the original variables) does not clarify the essential features. Moreover, there is no theoretical guarantee for the empirically chosen variables to be the optimal ones (that can help to improve the FDD accuracy to the largest extent). Hence, rigorous feature selection on theoretical level needs to be studied.

In this chapter, a novel Information Greedy Feature Filter (IGFF) method is proposed. IGFF selects the optimal subset of features by maximizing the mutual information between candidate variables and fault labels. The selected features

can not only guide future experimenters and operators to deploy sensors, but also increase the building FDD accuracy.

Related works on traditional feature selection methods can be divided into two categories, the filter method and the wrapper method [114]. The filter method selects informative features and suppresses the least interesting ones regardless of the underlying model assumption. Usually some dependence metrics e.g., the correlation between variables and the target, are adopted as the selection objective [115,116]. However, previous filter methods are still heuristics that lack theoretical justification. The wrapper method blends in a classifier with the straightforward goal to minimize the classification error [117]. Usually, features selected by the wrapper method can yield high accuracy only for the particular classifier. To incorporate different scenarios, the selected variables are expected to be method-independent. Thus, the filter method is more suitable for building FDD and will be the focus of this chapter.

The remaining part of this paper is arranged as follows. Section 5.2 introduces the theoretical background. Section 5.3 describes the AHU system and formulates the optimal variable selection problem as a mutual information maximization problem. Section 5.4 is devoted to developing IGFF and analyze its performance. Case study and experimental results are given in Section 5.5. Section 5.6 summarizes the paper.

5.2 Preliminaries

This section firstly reviews the notions of mutual information and submodular function. Then, it formulates the optimal sensor configuration and feature selection problem for AHU into cardinality constrained mutual information maximization. Lastly, it analyzes the property of the objective and motivate approximate solution methods.

5.2.1 Mutual Information

Consider two random variables X and Y. The mutual information between X and Y, denoted as $\boldsymbol{I}(X;Y)$, can be defined in various ways:

$$\boldsymbol{I}(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{5.1}$$

$$= H(X) + H(Y) - H(X,Y) \tag{5.2}$$

$$= \mathbb{E}_{P(X,Y)} \log \left(\frac{P(X,Y)}{P(X)P(Y)} \right) \tag{5.3}$$

$$= D_{KL}\left(P(X,Y) \| P(X)P(Y)\right) \tag{5.4}$$

where $H(.)$ represents entropy, $P(.)$ presents probability, and $D_{KL}(.)$ represents the $K-L$ divergence.

The first line of the definition shows an intuition: since entropy is a measure of uncertainty in bits, the mutual information is the difference between the uncertainty of the random variable X and the uncertainty of X given additional information contained in Y. The definition is symmetric, as can be seen from the second and third line of equivalence. The last two lines reveal another intuition: the mutual information is the K-L divergence between the joint probability $P(X,Y)$ and the product of the marginals $P(X)P(Y)$, which measures how "far" the two random variables are from being independent.

The estimation of mutual information can be done by first estimating the joint probability from data and then plugging into the definitions. In this chapter, [118] is utilized for mutual information estimation.

5.2.2 Submodular Function

Submodularity is a natural diminishing returns property which widely exists in economics, game theory, and network systems. There are three equivalent defini-

tions of submodular functions that reveal distinct interpretations of submodularity correspondingly.

Definition 1. *Submodular Set Function*

A submodular funciton is a set function $f : 2^{\Omega} \to \mathbb{R}$, which satisfies one of the three equivalent definitions:

1. *For every $S, T \subseteq \Omega$ with $S \subseteq T$, and every $x \in \Omega \setminus T$,*

$$f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T) \tag{5.5}$$

2. *For every $S, T \subseteq \Omega$,*

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T) \tag{5.6}$$

3. *For every $S \subseteq \Omega$, and $x_1, x_2 \in \Omega \setminus S$,*

$$\begin{aligned} f(S \cup \{x_1\}) + f(S \cup \{x_2\}) \geq \\ f(S \cup \{x_1, x_2\}) + f(S) \end{aligned} \tag{5.7}$$

A set function f is called *supermodular* if $-f$ is submodular. Definition 1 has direct correction with the diminishing return property: the two sides of (5.5) can be considered as marginal returns of the set function f at S versus the return at T, by adding additional element x. Definition 2 is better comprehended in the classic max k-cover problem [119]. Definition 3 demonstrates that the contribution of two elements is maximized by adding them individually to the base set. Note that this property can be easily extended to the case with general k elements, which will be used later to define submodularity index.

View $f(S \cup \{x\}) - f(S)$ as a "first order derivative" of f at base set S, the first definition in fact requires non-increasing derivative. Consequently, submodularity

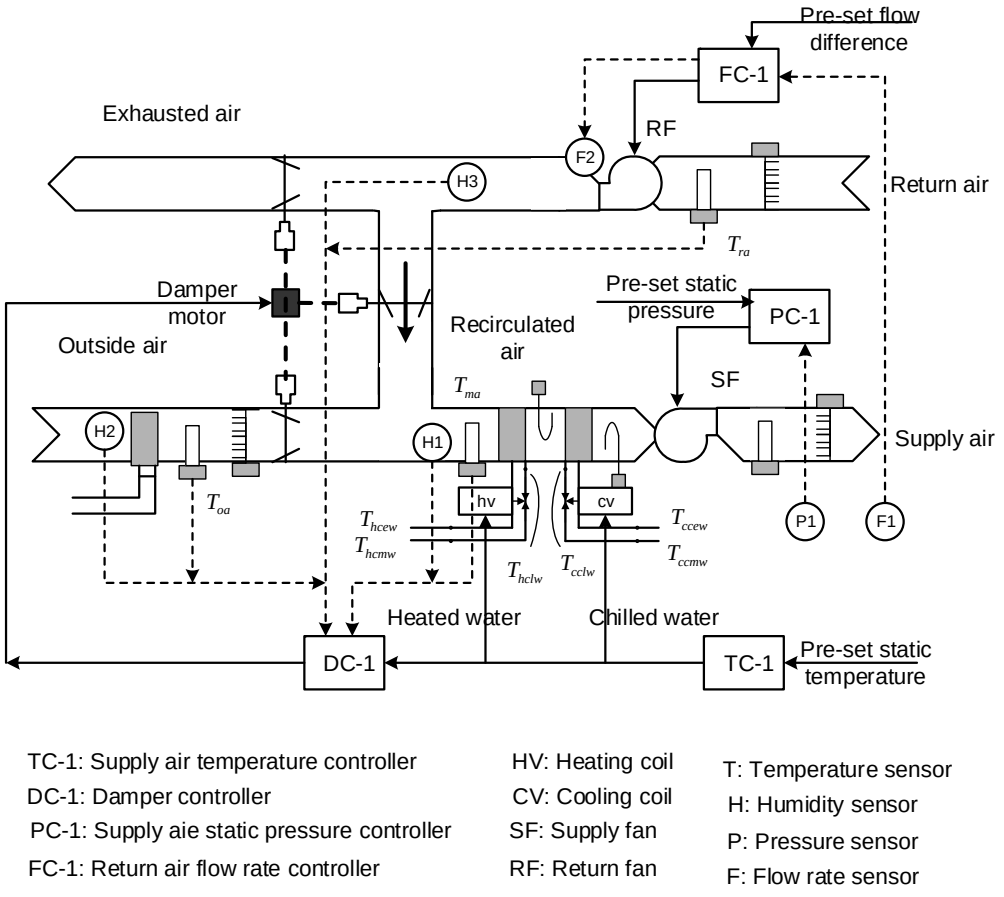


Figure 5.1: A typical single-duct VAV AHU system. The VAV system maintains the supply air temperature (T_{sa}), which is measured and compared with pre-set temperature by TC-1. TC-1 is linked to DC-1 to automatically adjust outside/return air dampers for appropriate mixing air temperature (T_{ma}) before entering the coil.

appears to be similar to "concavity" for set functions. Throughout this paper, $f_X(S) \triangleq f(S \cup X) - f(S)$ is denoted for further analysis. The "concavity" intuition coincides with the well-known fact that, despite being NP-hard, maximizing submodular function with simple greedy heuristic has near optimal performance guarantees [120]. On the other hand, it is also worth pointing out that submodular function is also closely related to "convexity" due to its convex Lovász extension, with which polynomial time algorithms, such as $O(n^5\alpha+n^6)$ in [121], can be designed for unconstrained minimization.

5.3 Problem Formulation

5.3.1 AHU and Faults

Modern building ACMV systems are equipped with maintenance routine and are capable of identifying some obvious faulty situations. To further improve the maintenance and reduce the cost, specialized FDD strategy with delicate design and high sensitivity is still of great importance. AHU is one of the most extensively operated system in large commercial buildings. Typically, AHU is exceedingly customized and is usually composed of sub-systems [122,123]. There is a high chance for AHU to encounter hardware failures and control errors due to improper system design, configuration and operation. Thus, unlike regular system FDD, AHU FDD is relatively more complicated.

A common AHU is configured with Constant Air Volume system (CAV) or Variable Air Volume system (VAV). In a VAV system, the supply fan is equipped with a Variable Frequency Drive (VFD) which modulates the air flow according to different building load conditions. Whereas, a CAV system supplies air flow to a zone constantly despite of building load variations. Fig. 5.1 depicts a typical single-duct VAV system, which includes four subsystem controllers: the supply air Temperature Controller (TC-1), the Damper Controller (DC-1), the supply air static Pressure Controller (PC-1) and the return air Flow-rate Controller (FC-1).

AHU operating modes change in agreement with the seasonal outdoor air temperature and humidity. There are four different modes as shown in Fig. 5.2. In the mechanical heating mode (Mode 1), the outdoor air damper is maintained at its minimum position. The supply air temperature is kept at its set-point by controlling the heating coil valve position. In the free cooling mode (Mode 2), both heating and cooling coil valves are closed. The supply air temperature is maintained at its set-point by modulating the outdoor air dampers only. In the mechanical and econ-

omizer cooling mode (Mode 3), the outdoor air damper is at the maximum position. The supply air temperature is kept at the cooling set-point by adjusting the cooling coil valve. In the mechanical cooling mode (Mode 4), the outdoor air damper is fixed at the minimum position since the outdoor air temperature can not meet the economizer set-point. The cooling coil valve is modulated to maintain the supply air temperature at the cooling set-point.

According to their causes and locations, there are four categories of faults, i.e. failures in AHU equipment, actuators, sensors and feedback controllers [26]. Sensors and controller faults are similar since feedback controllers are typically operated in accordance with sensor measurements. In experimental works (such as [41] and [34]), sensor faults are emulated by manually changing the sensor calibration in the control system. Twenty-five AHU faults that are commonly encountered in three seasons (11 typical faults occur in spring, 8 typical faults occur in summer and 6 typical faults occur in winter) are studied in this chapter. More information about AHU faults can be found in Section 5.5.1.

5.3.2 Sensor Configuration and Feature Selection

The data-driven FDD which formulates the AHU FDD as a multiple classification problem is the focus of this chapter. IGFF is applied to select relevant variables regarding maximum mutual information in the first step. Then, the selected optimal subset of variables is fed to different classification algorithms for FDD. More information about experiment set-up is in Section 5.5.2. In this subsection, the optimal sensor configuration and feature selection problem for AHU is formulated as the cardinality constrained mutual information maximization.

To formulate the feature selection problem, the goal is to select a subset of features, or variables measured by the AHU sensor network, that has maximal dependence with the target random variable, i.e., the fault label Y. With mutual information

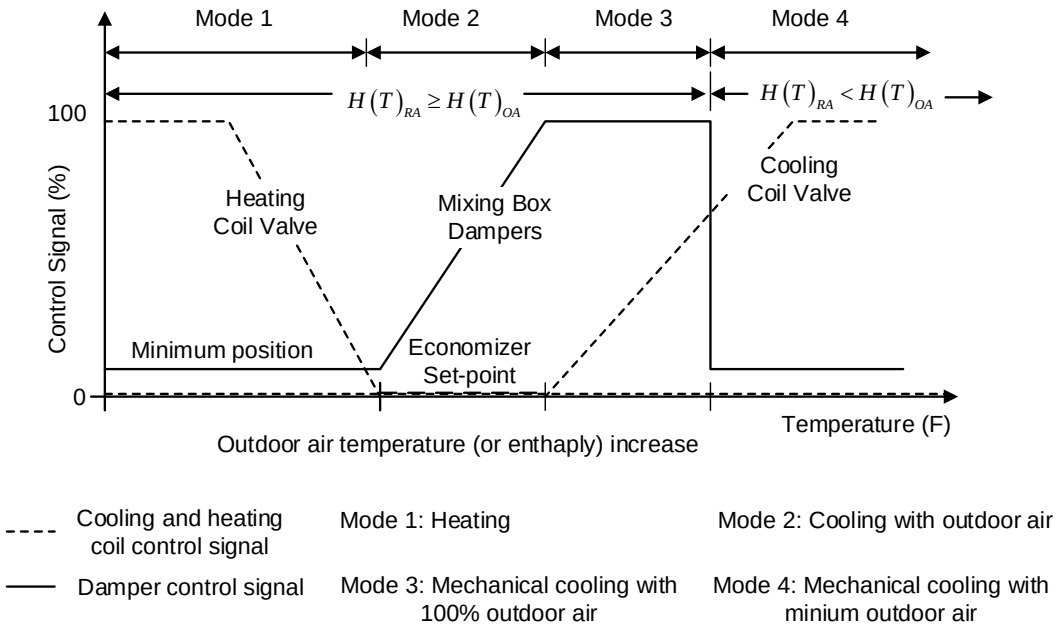


Figure 5.2: Operating modes of AHU. An economizer set-point can be the outdoor temperature set-point, the combination of outdoor temperature and humidity set-points or the outdoor enthalpy set-point. When the outdoor temperature (and humidity) are above the economizer set-point, the outdoor air intake will be a minimum quantity just to satisfy the ventilation requirement.

used as the dependence metric, the problem can be formulated as finding $S \subseteq V$, which has maximal $\boldsymbol{I}(S, Y)$. To leverage sparsity, the cardinality constraints $|S| \leq k$ is imposed on the number of selected features. Hence, the subset selection problem for feature selection reads

$$\underset{S \subseteq V, |S| \leq k}{\operatorname{argmax}} \boldsymbol{I}(S; Y) \tag{OPT}$$

Hardness of the problem

The above problem is shown to be NP-hard even in the most special case: consider a collection of independent Gaussian variables; then the above problems can be reduced to the D-optimal design problem [124].

Proposition 1. *The mutual information maximization problem (OPT) is NP-complete.*

Unless "P = NP", it is unlikely to find any polynomial algorithm for the maximization. Thus, finding a resort to approximation algorithms is necessary. Given the consecutively successful application of greedy algorithms for the subset selection of submodular functions, this study is motivated to apply the same technique. The key issue, after all, is the submodularity of the above objective function, which is discussed in the following subsection. The following subsection will analyze the property of the objective and motivate approximate solution methods.

Solution Strategy

Firstly, to analyze the submodularity of the objective (OPT).For any $x, Y, S \subseteq V$ with $f(S) \triangleq \boldsymbol{I}(S; Y)$, the derivative $f_x(S)$ at S for "direction" x has a more compact form regarding conditioned mutual information: $f_x(S) = \boldsymbol{I}(x; Y|S)$.

According to the first definition of submodularity, if the derivative is decreasing in S, i.e. if $f_x(S) \geq f_x(T)$ for any $S \subseteq T \subseteq V$ and $x \subseteq V \setminus T$, the objective $\boldsymbol{I}(S; Y)$ is a submodular function. By intuition, it seems that the dependence of two considered phenomena decreases with a larger conditioning set (knowing more information). Notwithstanding, it is generally not correct, and a counterexample could be constructed by having "explaining away" variables in graphic models. Therefore, the difficulty for solving (OPT) is in general the objective $\boldsymbol{I}(S; Y)$ is not submodular.

Note that with some extra conditional independence assumptions one can justify the submodularity as stated in [125]. More specifically, if for any two random variables $s_1, s_2 \in S$, the naive Bayesian model is satisfied, i.e., $(s_1 \perp\!\!\!\perp s_2 \mid Y)$, then $\boldsymbol{I}(S; Y)$ is a monotonic submodular function of set S [125].

In practice, the above assumption is rarely satisfied. If the conditional dependence is weak or sparse, possibly the submodularity is not severely deteriorated. This observation suggests that one can define a measure for the degree of submodularity, instead of treating it as a yes-or-no property of set functions.

A novel metric, namely Submodularity Index (SmI), is proposed in [126] to deal with the lack of submodularity. In this chapter, the author follows the similar idea and show that the performance of greedy algorithms is continuously determined by this index. Hence, theoretically, one can apply greedy heuristics to the maximization of a much broader class of set functions.

5.4 IGFF Algorithm and Performance Guarantee

This section first introduces the key term, Submodularity Index (SmI) and demonstrates its interesting properties. Then, it applies the SmI to study the theoretical performance guarantee of the proposed IGFF algorithm.

5.4.1 Approximate Submodularity with SmI

In literature, there are other works that characterize approximate submodularity, such as the ε relaxation of definition (5.5) proposed in [127] for a dictionary selection objective, and the submodular ratio proposed in [128] for R^2 score. The SmI of [126] is inspired by the third definition (5.7) of submodular function, and it is parallel to the submodular ratio defined in [128]. Compared with existing works, 1) SmI is more generally defined for all set functions. 2) it does not presume monotonicity. 3) In terms of computational convenience, SmI is suitable for functions involving information, influence, and coverage metrics.

Firstly, following [126], the *local submodular index* for function f at location A for candidate set S is defined as

$$\varphi_f(S, A) \triangleq \sum_{x \in S} f_x(A) - f_S(A) \tag{5.8}$$

This definition can be considered as an extension of (5.7) for submodular functions.

Essentially, it captures the difference between the sum of individual effect and aggregated effect on the first derivative of the function. Moreover, for a given submodular function f, the local submodular index $\varphi_f(S, A)$ is super-modular of S.

Then, SmI is defined by minimizing over candidate variables, i.e., for a set function $f : 2^V \to \mathbb{R}$, the submodularity index (SmI) for location set L and cardinality k, which is denoted by $\lambda_f(L, k)$, is defined as

$$\lambda_f(L,k) \triangleq \min_{\substack{A \subseteq L \\ S \cap A = \emptyset,\ |S| \leq k}} \varphi_f(S, A) \tag{5.9}$$

Thus, SmI is the smallest possible value of local submodularity index subject to $|S| \leq k$. Note that if implicitly assumes $|S| \geq 2$ in the above definition, as in the cases $|S| = \{0, 1\}$, SmI reduces trivially to 0. In addition, a set function f is submodular if and only if $\lambda_f(L, k) \geq 0 \quad \forall,\ L \subseteq V$ and k.

For functions that are already submodular, SmI scales how strong the submodularity is. A function is *super-submodular* if its SmI is strictly larger than zero. For functions that are not submodular, SmI provides an indicator of how close the function is to submodularity. A function is *quasi-submodular* if it has a negative but close to zero SmI.

Directly computing SmI by solving (5.9) is hard. In order to obtain performance guarantee, a lower bound of SmI is sufficient and is much easier to compute. First, observe the following transformation

$$\sum_{x \in S} \boldsymbol{I}(x; Y|A) - \boldsymbol{I}(S; Y|A) = \mathcal{G}\left(S, \{A, Y\}\right) - \mathcal{G}\left(S, \{A\}\right)$$

where $\mathcal{G}(W, Z) \triangleq H(W|Z) - \sum_{w \in W} H(w|Z)$ defined in terms of entropy is a submodular function of W. By further investigating the properties of the function $\mathcal{G}$, one gets a lower bound for the SmI of the objective of (OPT). For any location sets

$L \subseteq V$, cardinality k, and target process set Y, one has

$$\begin{aligned}&\lambda_{\boldsymbol{I}(\{\bullet\}^n;Y^n)}(L,k)\\ &\geq \min_{\substack{W\subseteq V\\ |W|\leq|L|+k}} \sum_{t=1}^{n} \left\{\mathcal{G}_{|L|+k}\left(W^t, Y^{t-1}\right) - \mathcal{G}_{|L|+k}\left(W^t, Y^t\right)\right\}\end{aligned} \tag{5.10}$$

Since Eq. (5.10) is in fact optimizing the difference of two submodular (supermodular) functions, existing approximate or exact algorithms [129] [130] could be used to compute the lower bound.

5.4.2 IGFF Algorithm and Performance Bound

This subsection analyzes the performance of IGFF for maximizing non-monotonic, quasi or super-submodular function in a unified framework. This general treatment is emphasized as it enables a much richer class of functions to have access to submodularity with theoretical guarantee.

With the knowledge of SmI, the IGFF algorithm (Algorithm 5.1) proposed in this chapter is a variant of the classic greedy algorithm for maximizing cardinality constrained monotonic submodular functions. Note that the IGFF algorithm has the $O(k|V|)$ complexity (number of calls of the oracle mutual information function), making it suitable for large-scale problems.

In order to analyze the performance of the algorithm despite of its lack of submodularity, more properties of SmI will be revealed. Given a set function $f : V \to \mathbb{R}$ and the corresponding SmI $\lambda_f(L,k)$ defined in (5.9), letting $B = A \cup \{y_1, ..., y_M\}$ and $x \in \overline{B}$, and for $\{j_1, ..., j_M\}$, defining $B_m = A \cup \{y_{j_1}, ..., y_{j_m}\}$, $B_0 = A$, $B_M = B$, then

$$f_x(A) - f_x(B) \geq \max_{\{j_1,...,j_M\}} \sum_{m=0}^{M-1} \lambda_f(B_m, 2) \geq M\lambda_f(B, 2)$$

Essentially, the above result implies that, for functions lacking strict submodular-

Algorithm 5.1 IGFF Algorithm for Subset Selection

Input Feature candidate set V, cadinality k
$S_0 \leftarrow \phi$
for $i = 1, ..., k$ **do**
  $u_i = \text{argmax}_{u \subseteq V \setminus S_{i-1}} \boldsymbol{I}(u; Y|S_i)$
  $S_i \leftarrow S_{i-1} \cup \{u_i\}$
end for
Output S_k

ity, as long as the second order SmI can be lower bounded by some small negative number, the increasing derivative property (hence the submodularity as defined in (5.5) is not severely degraded.

Theorem 5.1. *The IGFF algorithm achieves*

$$f(S^g) \geq \left(1 - \frac{1}{e} + \frac{\lambda'_f(S^g, k)}{f(S^g)}\right) f(S^*), \quad \textit{where}$$

$$\lambda'_f(S^g, k) = \begin{cases} \lambda_f(S^g, k) & \textit{if} \quad \lambda_f(S^g, k) < 0 \\ (1 - 1/e)^2 \lambda_f(S^g, k) & \textit{if} \quad \lambda_f(S^g, k) \geq 0 \end{cases}$$

Proof. Let S^* denote the true optimal and x_{i+1} denote the greedy selection at base set S_i. Consider the following inequalities:

$$\begin{aligned} f_{x_{i+1}}(S_i) &\geq \max_{\substack{M_{i+1} \subseteq V \setminus S_i \\ |M_{i+1}| \leq k}} \frac{1}{k} \sum_{x \in M_{i+1}} f_x(S_i) \geq \frac{1}{k} \sum_{x \in S^* \setminus S_i} f_x(S_i) \\ &\geq \frac{1}{k} \left[\lambda_{S_i,k} + f(S^* \cup S_i) - f(S_i)\right] \\ &\geq \frac{1}{k} \left[\lambda_{S_i,k} + f(S^*) - f(S_i)\right] \end{aligned}$$

where the first inequality is valid because x_{i+1} is the maximal greedy selection, the second one is valid because $S^* \setminus S_i \subseteq V \setminus S_i$ and $|S^* \setminus S_i| \leq k$, the third one is from the definition of SmI, and the last is valid because f is monotonic. Rearranging the

inequality yields

$$f(S_{i+1}) \geq \left(1 - \frac{1}{k}\right) f(S_i) + \frac{1}{k} f(S^*) + \frac{\lambda_{S_g,k}}{k} \tag{5.11}$$

Induction is adopted to prove the rest. Assume

$$f(S_i) \geq \left[1 - \left(1 - \frac{1}{k}\right)^i\right] f(S^*) + \frac{\lambda_{S_g,k}}{k} \sum_{j=0}^{i-1} \left(1 - \frac{1}{k}\right)^j$$

It can be verified that this assumption stands for $i = 1$ with the definition and monotonicity of SmI. From i to $i + 1$, plugging the assumption into (5.11)gives

$$f(S_g) \geq \left[1 - \left(1 - \frac{1}{k}\right)^k\right] f(S^*) + \lambda_{S_g,k} \left[1 - \left(1 - \frac{1}{k}\right)^k\right]$$

If the function is submodular, one has $\lambda_{S_g,k} \geq 0$, then

$$f(S_g) \geq \left(1 - \frac{1}{e}\right) f(S^*) + \left(1 - \frac{1}{e}\right) \lambda_{S_g,k} \tag{5.12}$$

$$\geq \left[1 - \frac{1}{e} + \left(1 - \frac{1}{e}\right)^2 \frac{\lambda_{S_g,k}}{f(S_g)}\right] f(S^*) \tag{5.13}$$

which is based on $f(S_g) \geq \left(1 - \frac{1}{e}\right) f(S^*)$. On the other hand, if $\lambda_{S_g,k} \leq 0$, one gets

$$f(S_g) \geq \left(1 - \frac{1}{e}\right) f(S^*) + \lambda_{S_g,k} \tag{5.14}$$

$$\geq \left(1 - \frac{1}{e} + \frac{\lambda_{S_g,k}}{f(S_g)}\right) f(S^*) \tag{5.15}$$

□

Compared to the classic $1 - 1/e \approx 0.6321$ guarantee, it is seen that for general monotonic functions, a stronger bound is obtained when the function is submodular. More importantly, for functions that are not submodular, such as the mutual infor-

mation considered here, it has been shown that the guarantee is degraded but not too much if the negative value of SmI is close to 0. The SmI dependent performance bound implies that submodularity is better characterized by a continuous indicator than being used as a "yes or no" property, which extends the application of greedy algorithms to a much broader class of problems.

Another useful observation is that the performance bound is only related with the ratio $\lambda/f(S_g)$. In fact, in the proofing, a stronger result in terms of $\lambda/f(S^*)$ is shown for all cases. Also, a measure of submodularity that is comparable across different set functions would be preferable. These considerations lead us to define Normalized Submodularity index (NSmI) as

$$\Lambda_f(L,k) \triangleq \frac{\lambda_f(L,k)}{f(L^*)} \tag{5.16}$$

5.5 Experimental Results and Comparison

5.5.1 Data Description

In this chapter, the proposed feature selection framework for AHU FDD is tested with the experimental data collected by the ASHRAE Research Project 1312 (RP-1312) [34]. The project was implemented in the test facility at the Energy Resource Station (ERS). As a brief introduction, RP-1312 conducted several on-site experiments to emulate the dynamic behaviors of a single duct dual fan VAV AHU system serving four building zones under various seasonal conditions. Authors of RP-1312 also archived the experimental data under normal and typical faulty status that could be used in future research. Interested readers can refer to [42] for the details about the test facility provided by Price and Smith.

As shown in Fig. 5.3, the experiment involved two AHUs, i.e., AHU-A and AHU-B, which served as treatment and control groups, respectively. The testing space

included rooms Inner A & B, West A & B, South A & B, and East A & B. Faults were manually introduced into the air-mixing box, coils, and fan sections of AHU-A, while AHU-B was operated at nominal states. During each experiment, the system operation was scheduled "ON" during occupied period from 6 : 00 to 18 : 00 and "OFF" during unoccupied period from 18 : 00 to 6 : 00. All the experiments were conducted under the real weather and building load conditions. Tables 5.1-5.3 list all the typical AHU faults considered in this chapter, which are emulated by RP-1312 during spring, summer and winter, respectively. Details about how those faults are implemented can be found in [34].

5.5.2 Experiment Set-up

With the RP-1312 data, IGFF is applied to select the optimal subset of variables for AHU FDD at first stage. IGFF algorithm selects optimal variables by maximizing mutual information between the feature vector $x_i \in \Re^{1\times n}, i = 1, ..., m$ and the class label vector $Y \in \Re^{1\times n}$. In the case of this paper, there are $n = 720$ samples, and $m = 107$ features for each fault (control signals are beyond the scope of consideration). For comparison purposes, IGFF chooses a subset of k most related variables for each fault, where $k = 1, ..., 15$.

Once top k features for each fault are selected, they are then fused together as the input of some multi-class classifiers. To achieve detection and diagnosis simultaneously, the fault types as well as the nominal condition are combined as the class labels. Classification accuracy, which is defined as the ratio of correct prediction to total number as defined in Eqs. (4.40) and (4.41), is used to measure the FDD performance.

During the FDD procedure, the experimental data of RP-1312 is randomly shuffled to two groups, one for training and the other one for testing. The randomized training-testing round is repeated 20 times to obtain confidence intervals.

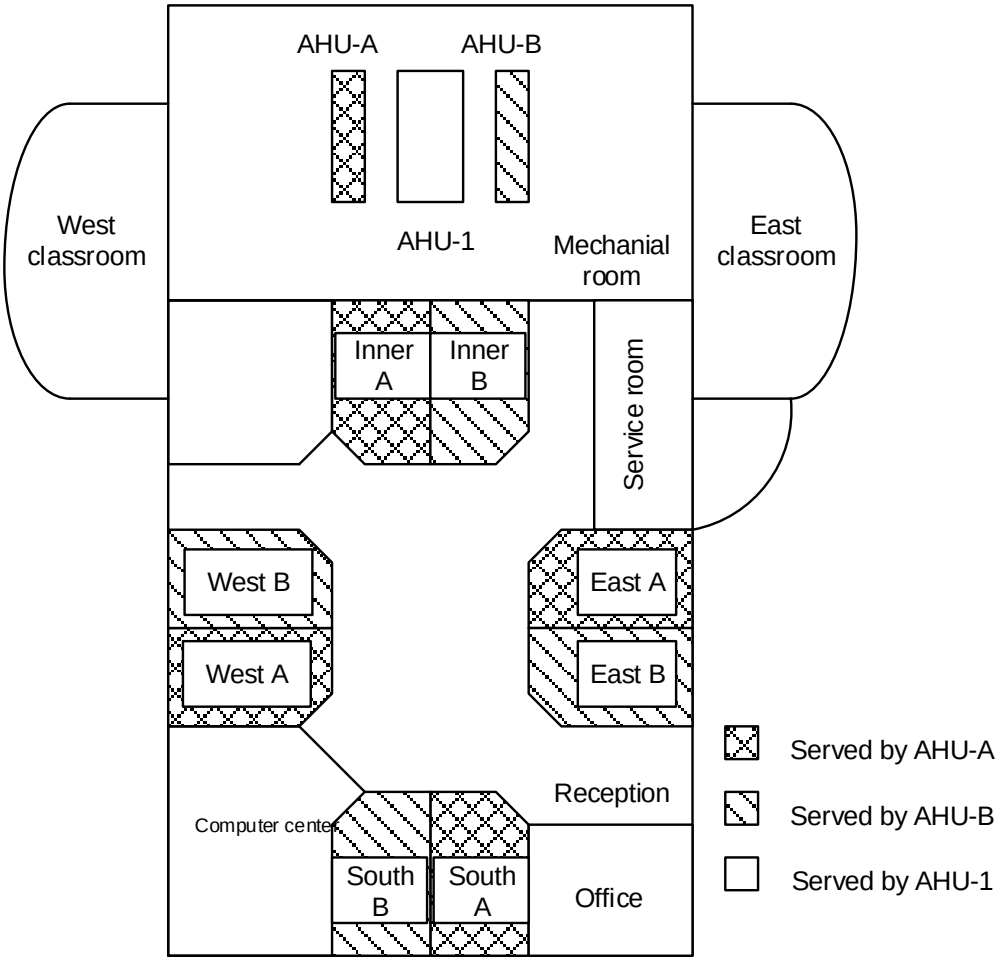


Figure 5.3: Layout of Energy Resource Station (ERS). AHU-A and AHU-B are identical, and each AHU serves four zones. Three of the four zones have external exposures and one only gets internal conditions. The A and B zones are mirror images with identical external thermal loads.

Multiple classification based FDD techniques considered in this chapter include Quadratic Discriminant Analysis (QDA), Logistics Regression (LR), Neural Networks (NN), and Multiple Support Vector Machine (MSVM) [94]. Since AHU operation modes are different among seasons, the FDD framework is formulated according to the seasonal distinctions. To be specific, this chapter focuses on 11, 8, and 6 typical faults emulated in spring, summer, and winter, respectively. Consequently, the FDD of AHU is a 12-class classification problem in the spring case, a 9-class

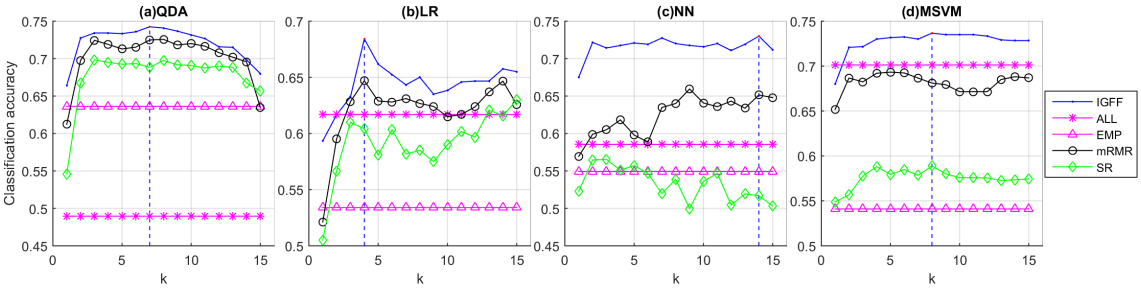


Figure 5.4: Spring test: FDD accuracy as a function of the number of selected features for 11 faults. Lines "IGFF", "ALL", "EMP,', "mRMR" and "SR" are the FDD accuracy generated with IGFF-selected features, all features, empirically-selected features, mRMR selected features, and SR selected features, respectively.

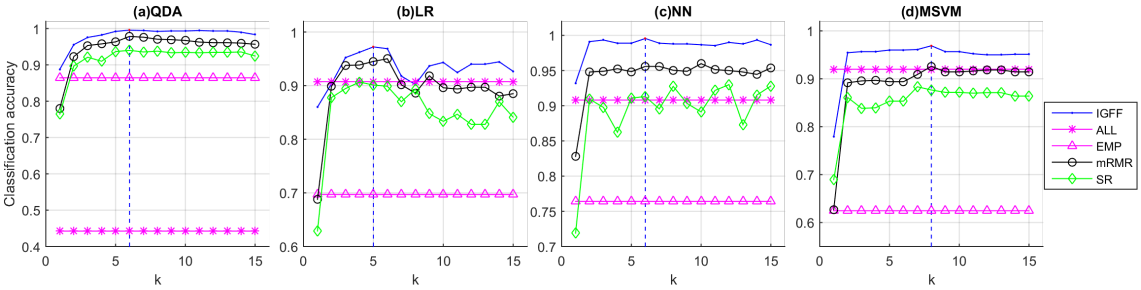


Figure 5.5: Summer test: FDD accuracy as a function of the number of selected features for 8 faults. Lines "IGFF", "ALL", "EMP,', "mRMR" and "SR" are the FDD accuracy generated with IGFF-selected features, all features, empirically-selected features, mRMR selected features, and SR selected features, respectively.

classification problem in the summer case, and a 7-class classification problem in the winter case correspondingly. In terms of the classification (FDD) performance of the aforementioned methods, IGFF selection is compared to four baselines, including all features, empirical features (12 common variables listed in Table 5.7) [131], features selected by the maximum Relevance Minimum Redundancy (mRMR) method [115] and the Sparse Regularization (SR) based method [132].

5.5.3 Feature Selection Results

Tables 5.1-5.3 list the selected variables for each fault with $k = 1, 2, ..., 15$ under three seasonal cases. Each variable is associated with a digit ID that represents its position in the archived RP-1312 data file. Tables 5.4 and 5.5 list names of corresponding variables.

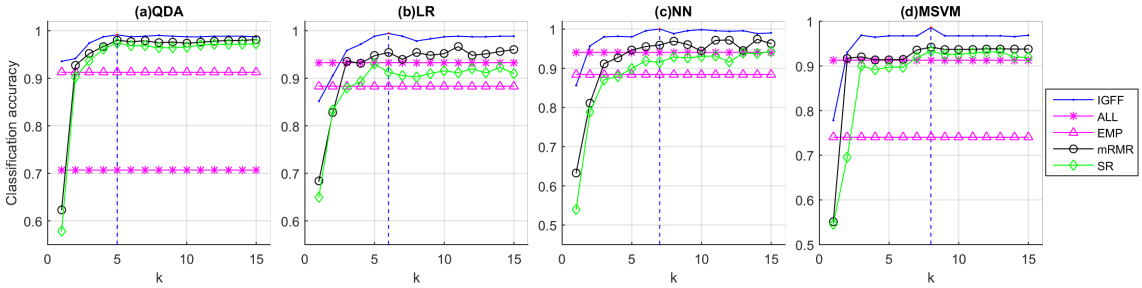


Figure 5.6: Winter test: FDD accuracy as a function of the number of selected features for 6 faults. Lines "IGFF", "ALL", "EMP,', "mRMR" and "SR" are the FDD accuracy generated with IGFF-selected features, all features, empirically-selected features, mRMR selected features, and SR selected features, respectively.

Results shown in Tables 5.1-5.3 review that the optimal sensor variables chosen by IGFF are not the same for different faults in different seasonal cases. Interestingly, even under the same seasonal condition optimal variables chosen by IGFF are disparate for similar faults. As shown in Table 5.1, the variable "cooling coil valve position" is the most relevant feature for detecting the Cooling Coil Valve Stuck (CCVS) fault. While for the Outside Air Damper Stuck (OADS) fault and the Exhaust Air Damper Stuck (EADS) fault, the most related features are not the damper positions but the "room air flow rate" and the "return air flow rate", respectively. Furthermore, optimal variables vary from season to season. Take the OADS fault as an example, the most relevant feature is "room air flow rate" for spring, "outside air damper position" for summer, and "inner room VAV heating coil entering water temperature" for winter.

It is worth mentioning that in real building ACMV infrastructures, sensor locations are also determined by AHU types. Hence, when choosing optimal variables that distinguish faults from normal working condition, particular circumstances of fault situations should be taken into consideration. As a result, the selection results obtained in this section are only applicable to similar AHU systems as the ASHRAE RP-1312. However, the proposed IGFF method is a general feature selection algorithm that can be applied to different FDD scenarios. Moreover from an experimental design perspective, IGFF could guide experimenters and building

operators to deploy sensors while certain budget constraint, e.g., number of available sensors, has to be satisfied.

5.5.4 FDD Performance in Terms of Selected Features

In this section, IGFF is compared to several baselines to verify that it benefits various classification methods (QDA, LR, NN, and MSVM). The results show that IGFF outperforms the state-of-the-art feature selection methods.

IGFF V.S. Baselines

In Figs. 5.4-5.6, the FDD accuracy generated by the aforementioned four classification methods is shown as a function of the number of selected variables (k). Baselines marked by "ALL", "EMP", 'mRMR', 'SR' are the FDD results produced with all features, empirically-selected features, mRMR-selected features, and SR-selected features [1]. As a whole, it is seen that IGFF is consistently better than baselines.

To be specific, it is obvious that the FDD using IGFF-selected features outperforms those using all features or empirically-selected features. For example, in the spring case, it is observed that as much as 25.29%(QDA), 6.73%(LR), 14.46%(NN), and 3.54%(MSVM) accuracy improvement as IGFF is applied instead of using all features. This observation justifies the argument that the redundant or noisy information in the raw data would in effect impair FDD performance. Hence feature selection for FDD is not only favourable when computation and resource cost is a concern, but also can help eliminate irrelevant information for better FDD performance. Besides, the accuracy improvement by IGFF compared to empirical features is as much as 10.63%(QDA), 15.01%(LR), 18.07%(NN), and 19.57%(MSVM). This reaffirms the benefit of a systematic feature selection method.

[1]Features selected by mRMR are listed in Tables 5.9-5.11. Features selected by SR are listed in Tables 5.12-5.14.

Compared to state-of-the-art feature selection methods such as mRMR and SR, IGFF also produces the best FDD performance in terms of accuracy. Again, in the spring case, the FDD accuracy with IGFF-selected features is higher than that with mRMR selection by as much as 5.18%, 7.22%, 13.06%, 6.35% for QDA, LR, NN, and MSVM, respectively. Also, IGFF-based FDD generates better accuracy than SR-based FDD by as much as 11.83%, 8.83%, 21.79%, and 16.41%, respectively. The enhancement appears since IGFF directly maximize mutual information and has theoretical guarantee. On the other hand, mRMR can be viewed as a first order approximation of information maximization [115], which is still a heuristic without guarantees. SR can only incorporate linear dependence while the dependence of the target on features often exhibits non-linearity in real-world FDD applications.

Detailed improvement by IGFF compared with the four baselines is listed in Table 5.8. Note that there are negative values in Table 5.8 since classification performance with one selected variable ($k = 1$) is worse than that with all features or the empirically-selected features in some cases. Besides, it is shown that the FDD accuracy is higher (by about 25%-30%) in summer and winter than that in spring. More fault types are the direct cause for the lower classification accuracy in the spring case. It's worth pointing out that the classification accuracy, although not perfect, is satisfactory as the number of class is relatively large (a random guess would only generate 8.33% accuracy).

The Impact of Feature Selection on Classification Methods

Next, the performance of classification as a function of selected features will be discussed.

1. As for the FDD accuracy generated with selected features (shown by line "IGFF" and baselines "mRMR" & "SR"), the accuracy value starts with an increasing trend along with the number of selected variables, then reaches the

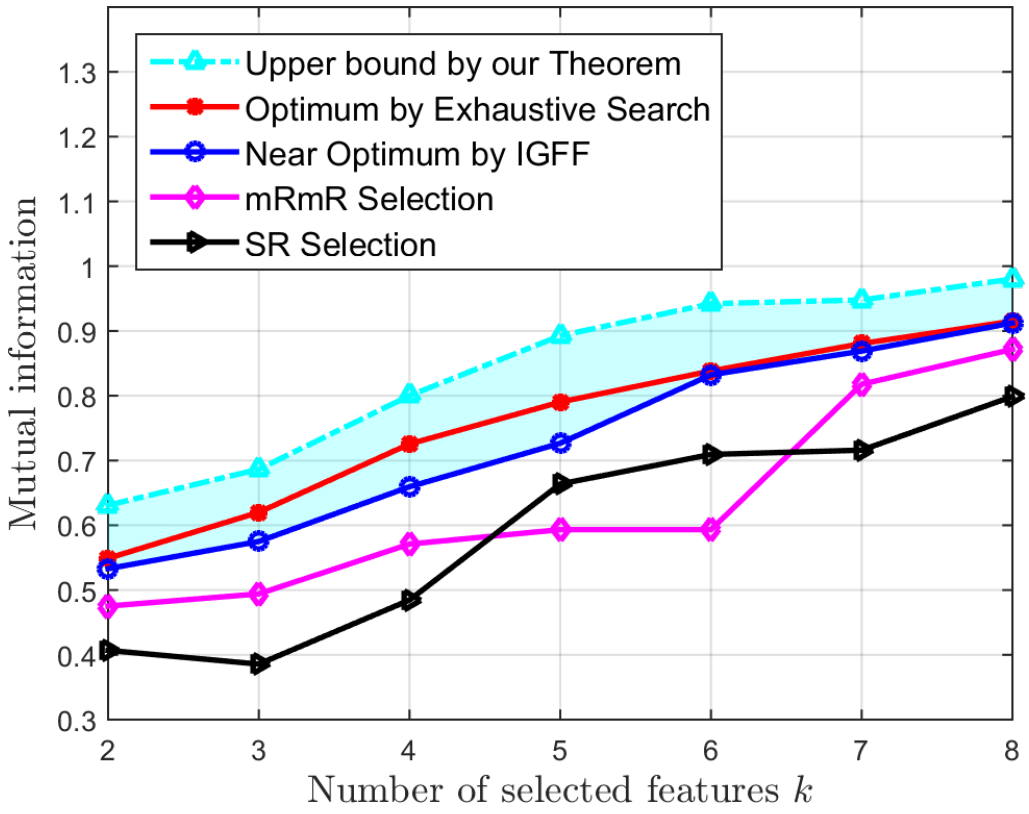


Figure 5.7: Greedy selection (IGFF) v.s. Exhaustive search (true optimum), with bound (shaded area) provided by Theorem I. Note that the exhaustive search takes 23.7 hours to run on a cluster having 16 Xeon E5687 CPUs and requires 128G memory, while IGFF only takes about 2 minutes on a laptop with a i7 3740qm CPU and 4G memory.

best performance and decreases afterwards. This can be intuitively explained as follows. Initially, incorporating more feature would enrich the information coverage while this coverage tends to saturate after certain peak value. After that, instead of enhancing FDD performance, non-informative features would introduce extra noise and lead to lower accuracy.

2. Although as a generic feature selection method, IGFF improves FDD accuracy for all classification methods, it is worth noticing that methods respond differently to the selected features under various conditions:

(a) Theoretically, QDA and LR are more sensitive to the number of selected features due to their conditional independence assumptions. Nevertheless, those assumptions are more vulnerable when the number of features is larger. This is particularly the case for sub-plots (a)(b) in Fig. 5.4, sub-plot (b) in Fig. 5.5, and sub-plot (b) in Fig. 5.6.

(b) Given a relatively small training and testing sample size, theoretically, NN would not be as stable as other methods since its performance is usually guaranteed by massive data. This can be observed in Table 5.8, where variances of the accuracy by NN are larger than that by other methods.

(c) − MSVM is the most stable method regarding the number of selected features k (except for $k = 1$) since SVM is established on the basis of selecting the most related variables. This can be observed in the four sub-plots (d) of Figs. 5.4-5.6.

IGFF Performance for Information Maximization

Feature selection result in terms of maximizing the mutual information is shown in Fig. 5.7. IGFF, mRMR and SR are applied to select the most related variables that contribute to distinguishing fault from normal for each of the 25 faults accordingly. The cardinality constraint is imposed from $k = 1$ to $k = 8$ under all the three cases. For comparison purpose, an exhaustive search is also conducted to obtain the true optimal solution and the corresponding objective values. It can be observed that IGFF consistently presents a near optimal performance and outperforms mRMR and SR regarding mutual information values. The performance bound given in

Theorem 5.1 is also calculated through Eq. (5.10) and plotted in Fig. 5.7 (shown as the shaded area). The bound seems loose at the first glance; however, it should be noted that Theorem 5.1 covers all the possible combinations of optimality. To the best of the authors' konwledge, this is the first time that theoretical guarantee is provided for FDD feature selection with mutual information.

Provided with the mutual information, the FDD performance can be consequently guaranteed via the Shannon Coding Analysis [133,134]. As Shown in Fig. 5.8, with Fano's inequality, the theoretical bound by IGFF can be used to generate bounds for the prediction accuracy. The confidence intervals (shaded areas between upper and lower bounds) can be obtained for the hypothesis that the predicted label $\hat{Y}$ equals to true label Y, i.e., the probabilistic range for correct detection. Hence, the shaded area illustrates possible degrees of confidence/certainty for the detection. As seen in Fig. 5.8, theoretically, the prediction can be 100% correct (zero false alarm for detection) with assumptions such as sample size is large enough, training data and testing data sets are from the same distribution, etc. Moreover, this prediction confidence is applicable to all the classification methods with FDD purpose.

5.6 Conclusion

In this chapter, the problem of optimally configuring sensors and selecting features for building FDD is addressed. A case study on AHU FDD is conducted. The proposed IGFF method is able to efficiently identify the most informative features for FDD. Moreover, IGFF is justified theoretically since it maximizes mutual information with guaranteed bound. To empirically verify the advantages of the proposed method, firstly, IGFF is applied to select essential features based on ASHRAE RP-1312 dataset. Then, the chosen sensor variables are fused together as the input of several classification techniques(QDA, LR, AB, NN, and MSVM). Compared to

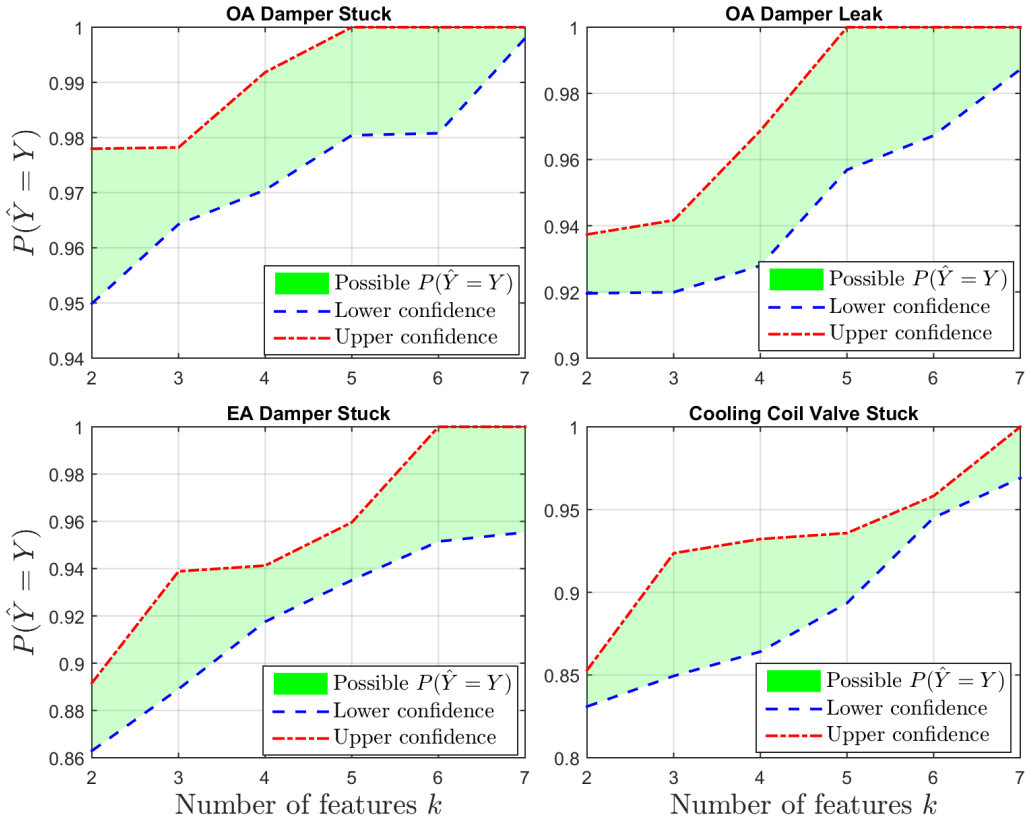


Figure 5.8: Prediction bounds by IGFF for the four actuator faults in the winter case as a function of the number of selected features (k). $P(\hat{Y} = Y)$ is the probability of the prediction $\hat{Y} = Y$ is true.

FDD results with all features, empirically-selected features, and features selected by state-of-the-art feature selection methods, it is observed that IGFF outperforms the other alternatives.

Table 5.1: Faults implemented in AHU-A during spring experiment and the optimal sensors for FDD selected by TFDK.

Category	Device	Fault Description	Optimal Variables (k)
Equipment Faults	Filter	Air filter blockage (25%)	30/43/53/8/22/9/3/12/79/31/13/29/14/69/6
Actuator Faults	Damper	OA damper stuck (fully close)	11/47/40/88/12/41/14/86/13/5/7/23/73/43/19
	Damper	EA damper stuck (fully close)	13/69/43/3/30/31/29/7/49/12/6/11/14/39/47
	Valve	Cooling coil valve stuck (fully open)	2/103/3/4/5/8/14/43/1/104/102/13/18/105/69
Sensor Faults	Temperature sensor	OA temperature sensor bias +3F	69/65/43/30/74/17/68/8/3/4/14/13/29/31/12
	Temperature sensor	OA temperature sensor bias -3F	43/30/39/74/8/68/22/3/13/6/69/14/29/31/12
Controller Faults	Fan controller	Return fan at fixed speed (80%)	7/30/24/107/8/43/22/13/69/6/14/29/31/82/12
	Damper controller	Mixed air damper unstable	43/18/30/16/8/31/64/24/29/14/13/69/6/82/12
	Damper controller	Mixed air damper unstable (Cooling Coil Control Unstable)	43/53/76/1/8/104/10/7/107/14/69/30/13/6/31
	Valve controller	Sequence of Heating and cooling unstable	43/30/75/8/9/104/91/1/10/3/12/29/69/33/13
	Fan controller	Supply fan control unstable	43/30/74/75/68/51/8/3/4/6/13/69/31/29/14

Table 5.2: Faults implemented in AHU-A during summer experiment and the optimal sensors for FDD selected by TFDK.

Category	Device	Fault Description	Optimal Variables (k)
Equipment Faults	Duct	AHU duct leak after supply fan	30/28/88/9/8/68/3/4/12/69/13/82/6/29/11
	Duct	AHU duct leak before supply fan	30/74/88/106/8/3/13/14/7/29/9/31/6/69/12
Actuator Faults	Damper	OA damper stuck (fully close)	5/31/29/49/5/14/69/48/86/6/12/25/13/11/7
	Damper	OA damper leak (55%)	5/9/61/81/107/3/4/12/13/69/14/29/30/31/6
	Damper	EA damper stuck (fully close)	49/3/88/65/9/107/68/7/6/31/13/30/48/29/12
	Valve	Cooling coil valve stuck (fully open)	14/30/1/81/59/3/42/6/60/13/2/18/86/103/34
Controller Faults	Valve controller	Cooling coil valve control unstable	31/29/30/88/87/9/68/22/13/12/69/48/49/6/11
	Valve controller	Cooling coil valve reverse action	104/22/3/33/14/8/4/5/1/102/103/2/18/105/88

Table 5.3: Faults implemented in AHU-A during winter experiment and the optimal sensors for FDD selected by TFDK.

Category	Device	Fault Description	Optimal Variables (k)
Equipment Faults	Coil	Heating coil fouling	48/49/6/13/7/23/69/5/82/14/43/24/21/106/16
	Coil	Heating coil reduced capacity	48/49/78/8/104/102/39/13/1/82/3/4/5/31/52
Actuator Faults	Damper	OA damper stuck (fully close)	48/49/78/5/38/13/12/14/82/43/26/11/4/3/73
	Damper	OA damper leak (62%)	78/106/9/8/46/3/82/92/13/4/48/39/7/6/5
	Damper	EA damper stuck (fully close)	48/49/13/69/7/82/3/4/14/5/16/43/12/104/6
	Valve	Cooling coil valve stuck (fully open)	78/34/2/13/39/7/69/6/33/43/14/3/82/4/48

Table 5.4: Sensor number and the corresponding variable names (I).

Sensor Number	Variable Name
1	Heating coil valve position
2	Cooling coil valve position
3	Exhaust air damper position
4	Return air damper position
5	Outside air damper position
6	Supply Fan power
7	Return fan power
8	Heating water pump water flow rate
9	Chilled water pump water flow rate
10	Supply fan speed
11	Room air flow rate
12	Supply air flow rate
13	Return air flow rate
14	Outside air flow rate
16	Supply air temperature
17	Mixed air temperature
18	Heating coil differential air temperature
19	Cooling coil differential air temperature
20	Static pressure
21	Supply fan differential pressure
22	Return fan differential pressure
23	Supply fan speed
24	Return fan speed
25	Supply air related humidity
26	Return air related humidity
27	Outside air temperature
28	Outside air damper temperature
29	Heating coil entering water temperature
30	Heating coil leaving water temperature
31	Heating coil mixed water temperature
32	Cooling coil entering water temperature
33	Cooling coil leaving water temperature
34	Cooling coil mixed water temperature
35	Heating water pump differential pressure (PSI)
36	cooling coil entering air humidity
37	cooling coil leaving air humidity
38	Inner room temperature
39	Inner room plenum temperature
40	Inner room VAV entering air temperature
41	Inner room VAV differential air temperature
42	Inner room VAV heating coil valve position

Table 5.5: Sensor number and the corresponding variable names (II).

Sensor Number	Variable Name
43	Inner room VAV damper position
44	Inner room VAV maximum flow
45	Inner room VAV minimum flow
46	Inner room VAV differential pressure
47	Inner room VAV air flow rate
48	Inner room VAV heating coil entering water temperature
49	Inner room VAV heating coil leaving water temperature
50	Inner room VAV water flow rate
51	West room temperature
52	West room plenum temperature
53	West room VAV entering air temperature
54	West room VAV differential air temperature
55	West room VAV heating coil valve position
56	West room VAV damper position
57	West room VAV maximum flow
58	West room VAV minimum flow
59	West room VAV differential pressure
60	West room VAV air flow rate
61	West room VAV heating coil entering water temperature
62	West room VAV heating coil leaving WT
63	West room VAV water flow rate
64	South room temperature
65	South room plenum temperature
66	South room VAV entering air temperature
67	South room VAV differential air temperature
68	South room VAV heating coil valve position
69	South room VAV damper position
70	South room VAV maximum flow
71	South room VAV minimum flow
72	South room VAV differential pressure
73	South room VAV air flow rate
74	South room VAV heating coil entering water temperature
75	South room VAV heating coil leaving water temoerature
76	South room VAV heating coil water flow rate
77	East room temperature
78	East room plenum temperature
79	East room VAV entering air temperature
80	East room VAV differential air temperature
81	East room VAV heating coil valve position
82	East room VAV damper position
83	East room VAV maximum flow

Table 5.6: Sensor number and the corresponding variable names (III).

Sensor Number	Variable Name
84	East room VAV minimum flow
85	East room VAV differential pressure
86	East room VAV air flow rate
87	East room VAV heating coil entering water temperature
88	East room VAV heating coil leaving water temoerature
89	East room VAV water flow rate
90	Inner room base board current Amps
91	West room base board current
92	South room base board current
93	East room base board current Amps
94	Inner room A lighting power
95	West room A lighting power
96	South room A lighting power
97	East room A lighting power
98	Inner room B lighting power
99	West room B lighting power
100	South room B lighting power
101	East room B lighting power
102	Heating coil water flow rate
103	Cooling coil water flow rate
104	Heating coil energy
105	Cooling coil energy
106	Supply fan energy
107	Return fan energy

Table 5.7: Empirical variables for AHU FDD research

Sensor Number	Variable Name	Sensor Number	Variable Name
1	Heating coil valve position	15	Supply air temperature
2	Cooling coil valve position	17	Mixed air temperature
4	Return air damper position	20	Supply air static pressure
6	Supply fan fan power	23	Supply fan speed
7	Return fan power	24	Return fan speed
12	Supply air flow rate	27	outside air temperature

Table 5.8: FDD accuracy values(%) of IGFF outperforms baselines.

Season	Method	IGFF Accu(mean$\pm 3\sigma$)	IGFF$-$ALL	IGFF$-$EMP	IGFF$-$mRMR	IGFF$-$SR	Optimal k of IGFF
Spring	QDA	73.3 $\pm$ 0.18	[17.5, 25.3]	[2.8, 10.6]	[0.3, 5.2]	[2.3, 11.8]	7
	LR	63.1$\pm$ 0.23	[-2.4, 6.7]	[5.9, 15.0]	[0.4, 7.2]	[2.3, 8.8]	4
	NN	69.4$\pm$ 0.52	[9.0, 14.5]	[12.6, 18.1]	[5.8, 13.1]	[14.9, 21.8]	14
	MSVM	72.1$\pm$ 0.211	[-2.1, 3.5]	[13.9, 19.6]	[2.8, 6.4]	[13.1, 16.4]	8
Summer	QDA	98.0$\pm$ 0.28	[44.5, 55.3]	[2.3, 13.1]	[1.7, 10.7]	[5.4, 12.3]	6
	LR	89.8$\pm$ 0.305	[-4.7, 6.5]	[16.3, 27.5]	[0.5, 17.2]	[0.9, 23.1]	5
	NN	92.3$\pm$ 5.785	[2.4, 8.8]	[16.8, 23.2]	[2.7, 10.4]	[5.9, 21.3]	6
	MSVM	94.4$\pm$ 0.32	[-14.1, 4.9]	[15.4, 34.4]	[3.2, 15.2]	[7.8, 11.8]	8
Winter	QDA	97.1$\pm$ 0.37	[22.9, 28.5]	[2.3, 7.9]	[0.6, 31.2]	[1.4, 35.7]	5
	LR	97.2$\pm$ 0.243	[-8.1, 6.1]	[-3.2, 11.1]	[2.2, 16.8]	[5.9, 20.1]	6
	NN	96.8$\pm$ 1.17	[-8.5, 6.0]	[-2.8, 11.6]	[1.5, 22.3]	[4.7, 31.6]	7
	MSVM	92.7$\pm$ 0.636	[-13.4, 7.3]	[3.8, 24.5]	[1.3, 22.7]	[3.6, 23.4]	8

*Note: IGFF-ALL, IGFF-EMP, IGFF-mRMR, and IGFF-SR are max/min improvements of IGFF v.s. ALL, EMP, mRMR, and SR, respectively.

Table 5.9: Faults implemented in AHU-A during spring experiment and the optimal sensors for FDD selected by mRMR.

Category	Device	Fault Description	Optimal Variables (k)
Equipment Faults	Filter	Air filter blockage (25%)	30/1/43/8/22/10/35/69/44/45/46/29/50/57/58
Actuator Faults	Damper	OA damper stuck (fully close)	11/1/5/22/20/107/21/40/10/75/9/19/35/43/33
	Damper	EA damper stuck (fully close)	13/1/3/43/10/9/69/22/35/8/39/44/40/7/45
	Valve	Cooling coil valve stuck (fully open)	2/89/8/3/4/104/43/5/10/1/9/22/103/35/102
Sensor Faults	Temperature sensor	OA temperature sensor bias +3F	69/1/43/10/8/30/22/35/53/79/44/59/65/45/40
	Temperature sensor	OA temperature sensor bias -3F	43/72/30/1/8/10/69/35/44/45/46/82/50/57/68
Controller Faults	Fan controller	Return fan at fixed speed (80%)	7/1/107/22/43/10/24/9/85/30/35/8/69/44/45
	Damper controller	Mixed air damper unstable	43/1/69/9/30/10/22/8/33/79/35/82/44/45/31
	Damper controller	Mixed air damper unstable (Cooling Coil Control Unstable)	43/1/30/8/10/69/22/9/76/19/35/33/44/45/79
	Valve controller	Sequence of Heating and cooling unstable	43/50/33/8/9/69/1/78/30/10/22/40/35/44/39
	Fan controller	Supply fan control unstable	43/1/69/9/8/74/89/40/30/10/22/82/35/44/67

Table 5.10: Faults implemented in AHU-A during summer experiment and the optimal sensors for FDD selected by mRMR.

Category	Device	Fault Description	Optimal Variables (k)
Equipment Faults	Duct	AHU duct leak after supply fan	30/9/1/8/88/10/79/74/22/65/35/69/44/45/46
	Duct	AHU duct leak before supply fan	30/1/9/8/21/80/74/10/22/39/35/43/44/79/65
Actuator Faults	Damper	OA damper stuck (fully close)	30/1/5/49/22/9/107/31/21/88/48/10/29/35/62
	Damper	OA damper leak (55%)	5/81/30/9/8/75/1/69/10/62/65/22/88/59/74
	Damper	EA damper stuck (fully close)	49/3/9/30/48/10/22/107/39/8/35/29/44/45
	Valve	Cooling coil valve stuck (fully open)	14/10/104/22/3/35/8/72/79/103/44/4/45/2/46
Controller Faults	Valve controller	Cooling coil valve control unstable	31/10/88/9/48/35/87/8/69/44/45/49/46/30/50
	Valve controller	Cooling coil valve reverse action	31/10/88/9/48/35/87/8/69/44/45/49/46/30/50

Table 5.11: Faults implemented in AHU-A during winter experiment and the optimal sensors for FDD selected by mRMR.

Category	Device	Fault Description	Optimal Variables (k)
Equipment Faults	Coil	Heating coil fouling	48/10/49/69/107/22/102/82/9/106/5/38/21/8/32
	Coil	Heating coil reduced capacity	48/2/49/82/8/78/3/43/38/4/39/33/5/69/10
Actuator Faults	Damper	OA damper stuck (fully close)	48/10/5/38/49/107/78/82/8/26/22/39/20/102/43
	Damper	OA damper leak (62%)	78/2/3/82/4/39/8/5/9/43/106/48/10/69/22
	Damper	EA damper stuck (fully close)	48/2/69/49/3/43/82/102/4/10/7/5/22/8/104
	Valve	Cooling coil valve stuck (fully open)	78/10/82/3/43/39/2/4/48/69/5/22/103/9/35

Table 5.12: Faults implemented in AHU-A during spring experiment and the optimal sensors for FDD selected by SR.

Category	Device	Fault Description	Optimal Variables (k)
Equipment Faults	Filter	Air filter blockage (25%)	35/9/20/43/30/102/104/22/77/8/79/25/72/31/46
Actuator Faults	Damper	OA damper stuck (fully close)	35/22/43/5/25/102/18/20/30/16/62/49/79/80/88
	Damper	EA damper stuck (fully close)	35/9/43/22/3/69/30/56/107/51/25/63/53/76/102
	Valve	Cooling coil valve stuck (fully open)	35/8/43/2/103/9/33/19/63/89/15/53/69/105/34
Sensor Faults	Temperature sensor	OA temperature sensor bias +3F	35/22/9/20/53/40/43/63/38/30/72/79/89/33/55
	Temperature sensor	OA temperature sensor bias -3F	35/22/20/43/9/63/80/50/30/79/72/64/51/77/28
Controller Faults	Fan controller	Return fan at fixed speed (80%)	35/22/43/9/24/107/20/30/104/23/79/80/66/69/8
	Damper controller	Mixed air damper unstable	35/102/9/43/59/72/104/20/53/69/66/50/40/89/30
	Damper controller	Mixed air damper unstable (Cooling Coil Control Unstable)	35/20/43/107/69/76/53/63/17/9/26/66/22/40/15
	Valve controller	Sequence of Heating and cooling unstable	35/53/20/43/22/78/9/40/28/16/18/63/46/30/76
	Fan controller	Supply fan control unstable	35/43/9/22/50/20/69/79/78/102/30/72/64/53/104

Table 5.13: Faults implemented in AHU-A during summer experiment and the optimal sensors for FDD selected by SR.

Category	Device	Fault Description	Optimal Variables (k)
Equipment Faults	Duct	AHU duct leak after supply fan	76/35/63/89/9/22/65/46/88/26/30/80/20/79/18
	Duct	AHU duct leak before supply fan	35/89/63/76/9/22/88/20/65/53/54/102/107/8/30
Actuator Faults	Damper	OA damper stuck (fully close)	50/89/63/76/35/49/22/75/62/48/5/21/31/29/30
	Damper	OA damper leak (55%)	35/5/63/22/76/9/65/20/46/51/77/53/26/30/89
	Damper	EA damper stuck (fully close)	50/63/89/76/35/22/49/48/104/9/87/74/88/61/8
	Valve	Cooling coil valve stuck (fully open)	35/89/76/22/87/46/88/64/53/54/20/9/65/66/78
Controller Faults	Valve controller	Cooling coil valve control unstable	50/89/76/63/35/88/75/62/48/22/9/29/46/20/310
	Valve controller	Cooling coil valve reverse action	35/33/76/9/103/19/65/53/89/2/66/54/34/67/20

Table 5.14: Faults implemented in AHU-A during winter experiment and the optimal sensors for FDD selected by SR.

Category	Device	Fault Description	Optimal Variables (k)
Equipment Faults	Coil	Heating coil fouling	50/35/9/48/49/103/22/20/107/85/38/21/26/46/32
	Coil	Heating coil reduced capacity	50/35/9/103/48/49/78/39/59/93/33/77/85/46/38
Actuator Faults	Damper	OA damper stuck (fully close)	50/35/9/48/38/78/49/103/46/39/17/41/53/65/25
	Damper	OA damper leak (62%)	35/9/103/78/33/39/50/52/21/20/38/65/17/53/49
	Damper	EA damper stuck (fully close)	50/35/9/103/48/49/22/77/21/64/20/46/33/3/59
	Valve	Cooling coil valve stuck (fully open)	35/78/39/46/59/65/85/25/43/21/52/77/90/20/17

Chapter 6

Conclusions and Future Works

6.1 Conclusions

Fault detection and diagnosis is quite essential in terms of reducing building energy consumption and improving indoor occupancy comfort level. Seven typical chiller faults and twenty-five AHU faults have been studied. Basically, the proposed FDD algorithms are designed with the assumption that all the sensor measurements are correct and reliable. However, as a common sense, there is high chance for sensor malfunctions such as sensor drafting, which is categorized as sensor fault. Sensor faults are not included in the seven typical chiller faults, but they are considered as one single category of AHU faults in Chapter 5. Although sensor faults are different form device/equipment faults as well as controller faults (as discussed in in Chapter 5), learning algorithms can also revel the hidden difference among those faults and fault free condition so as to identify them.

As a summary, this dissertation studies the data-driven techniques and proposes several effective strategies to solve the FDD problem for building chillers and AHUs. The proposed DAFC can identify the fault severity level at a second stage after fault types have been recognized. Then, the system structured information is incorpo-

rated to the learning algorithm such that fault types and the corresponding severity levels can be diagnosed in a uniformed framework. The proposed TFDK methods outperforms the state-of-the-art classification techniques in terms of building FDD. Besides, by selecting features that are more correlated with faults with the proposed IGFF algorithm, not only the FDD accuracy is improved, but also the FDD application becomes more convenient and practical. All in all, the FDD for building chillers and AHUs is studied from the following aspects.

1. A two-stage strategy for detection and diagnosis of typical chiller faults using distance-based classifier;

2. Fault detection and diagnosis for building cooling system with a tree-structured learning method;

3. Optimal sensor configuration and feature selection for AHU fault detection and diagnosis.

Firstly, the proposed two-stage data-driven FDD strategy formulates the chiller FDD task directly as a multiple classification problem. The Discriminant Analysis-based Fault Classification (DAFC) method is driven due to its intuition and simplicity. To tackle the "curse of dimensionality", LDA is firstly applied to project the high dimensional data into lower dimension so as to achieve maximum class separation and optimal information maintenance. After the LDA projection, fault types are detected and diagnosed by DAFC classifier in the lower dimensional space at the first stage. Similarly, at the second stage, the severity level of the diagnosed fault is identified by DAFC. The proposed strategy is validated by the experimental data of the ASHRAE Research Project 1043 (RP-1043). Results show that the two-stage FDD strategy using LDA and distance-based classifier can detect and diagnose chiller faults effectively.

Secondly, a tree-structured learning method is considered to identify fault types as well the corresponding fault severity level in a unified framework. The proposed TFDK method adopts structured labeling to include the inter-class fault dependence information and describe the severity levels. Thus, TFDK can identify a fault and recognize its severity level at one step. Also, TFDK is validated by the chiller fault data of RP-1043. Numerical results show that compared to previous data-driven methods, TFDK can greatly improve the FDD performance as well as recognizing the fault severity levels with high accuracy.

Lastly, the problem of optimally configuring necessary sensors and selecting essential features for building FDD is addressed. The proposed IGFF method can efficiently identify the most informative features, and it is justified theoretically since it maximizes mutual information with guaranteed bound. To verify the advantages of IGFF, the AHU FDD following the ASHRAE Research Project 1312 (RP-1312) is studied. Experimental results show that the FDD performance is improved by fusing IGFF-selected features to several common classification methods other than using all features, empirically-selected features, or features selected by state-of-the-art methods.

6.2 Future Work

The following problems would be considered in the future work.

6.2.1 On-site Experiments

In this dissertation, experimental data collected by ASHRAE Research Projects 1043 and 1312 are utilized to validate the proposed FDD strategies and algorithms. However, since different working conditions, such as outdoor environment, indoor electricity schedules, or indoor occupancy levels, will result in different building per-

formances, comprehensive experiments are needed so as to explore FDD strategies and algorithms that could be applied to building systems under as many working conditions as possible. On the one hand, there are many references about how to introduce soft faults to ACMV components, such as reducing water flow through a device to simulate tube fouling and adding foreign gas to change the inner pressure so as to simulate system overcharge. On the other hand, hard faults like high-power electrical appliances and abrupt device failure can be easily simulated by experimental methods. In addition, building simulation tools, such as HVACSIM+, EnergyPlus and TRNSYS, can be applied to provide a reference calibration for experimental data.

6.2.2 FDD with Incomplete Data

Usually, in commercial buildings, sensors are deployed to collect data periodically according to fixed protocols. The raw data collected from WSNs is rarely satisfactory for direct data analysis. Incomplete data, namely missing and corrupted values exist in the raw dataset, is one of the most frequently encountered situations for building researchers. Thus, reconstructing missing values before conducting FDD is an interesting topic. In pattern classification field, naive data interpolation methods have been extensively applied to handle missing values. However, the existing techniques merely utilize information extracted from either time series adjacency or channel adjacency of sensor streams to infer the expected data. Also, most of the them rely on stationary or independent and identically distributed assumptions that are rarely true in real FDD datasets. Its worthy of developing a method that could take both the time series and channel adjacency information into consideration. The reconstructed data will help with the improvement of FDD accuracy.

6.2.3 FDD with Unsupervised Method

When considering FDD for real buildings, supervised FDD is hardly possible due to the following reasons.

1. The BMS merely stores normal data while well labeled fault data is rare. What's worse, on-site experiments are hard to emulate in commercial buildings since experiments are supposed to perturb the ACMV working conditions and physical damage might be caused.

2. Granted that experiments are possible in laboratorial test-beds, currently available datasets are limited in terms of sample size and fault types. For example, as introduced in Chapter 3, only seven out of hundreds chillers faults were emulated in the ASHRAE RP-1043.

3. Since the building performance is influenced greatly by the outside/inside conditions, fault data collected in test-beds cannot be directly used as training datasets for detecting and diagnosing faults for commercial buildings.

Thus, from the aspect of real application, it is interesting to develop unsupervised or semi-supervised methods to detect building abnormal conditions based on limited labeled data and try to identify possible reasons.

6.2.4 Identifying Unseen Faults

Although typical faults introduced in Chapters 3 and 5 are common faults that occur most frequently, there are still high chances that other faults might happen in a real building. It is impossible to emulate every possible fault in the experiment. As a result, the FDD method needs to be capable of recognizing new faults. Other than the unsupervised method mentioned in last subsection, unseen (new) faults can

be identified by incorporating expert knowledge into the learning algorithm. This work would be an extension of the work described in Chapter 4.

6.2.5 Identifying Concurrent Faults

To the best of the author's knowledge, most of the existing building FDD studies are focusing on detecting and diagnosing single fault at one time. However, in real applications, both chiller plant and AHU may have multiple faults. If two faults occur at the same time, the fault characteristics should not be the simple superposition of their characteristics when occur alone. What's worse, in real systems, some faults usually might cause the occurrence of other faults. For example, a stuck fan may cause malfunctions of damper since the controllers are trying to maintain the system's balance, and the casual relationships are different from system to system. Given those aforementioned situations, it is impossible for researchers to emulate and collect data for all kinds of fault combinations and possibles situations. As a result, the traditional supervised classification techniques and the proposed methods in this dissertation seem to be not powerful enough. New algorithms should be designed specifically aiming at solving the concurrent and casual relationships among multiple faults.

AUTHOR'S PUBLICATIONS

Journal papers:

1. D. Li, Y. Zhou, G. Hu, and C. J. Spanos, "Optimal Sensor Configuration and Feature Selection for AHU Fault Detection and Diagnosis," *IEEE Transactions on Industrial Informatics*, vol.PP, no.99, pp.1-1, 2016.

2. D. Li, Y. Zhou, G. Hu, and C. J. Spanos, "Fault detection and diagnosis for building cooling system with a tree-structured learning method," *Energy and Buildings*, vol. 127, pp. 540-551, 2016.

3. D. Li, G. Hu, and C. J. Spanos, "A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis," *Energy and Buildings*, vol. 128, pp. 519529, 2016.

Conference papers:

1. D. Li, Y. Zhou, G. Hu, and C. J. Spanos, "Fusing System Configuration Information for Building Cooling Plant Fault Detection and Severity Level Identification," *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 1319-1325, 2016.

2. Y. Zhou, D. Li, and C. J. Spanos, "Learning optimization friendly comfort model for hvac model predictive control", 2015 IEEE International Conference on Data Mining Workshop (ICDMW), pp. 430-439, IEEE, 2015.

Bibliography

[1] G. Mantovani and L. Ferrarini, "Temperature control of a commercial building with model predictive control techniques," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2651–2660, 2015.

[2] J. Yao, G. T. Costanzo, G. Zhu, and B. Wen, "Power admission control with predictive thermal management in smart buildings," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 4, pp. 2642–2650, 2015.

[3] S. Katipamula and M. R. Brambley, "Review article: methods for fault detection, diagnostics, and prognostics for building systemsa review, part i," *HVAC&R Research*, vol. 11, no. 1, pp. 3–25, 2005.

[4] A. Schumann, J. Hayes, P. Pompey, and O. Verscheure, "Adaptable fault identification for smart buildings," in *Artificial Intelligence and Smarter Living*. AAAI Workshop, 2011.

[5] X. Li, C. P. Bowers, and T. Schnier, "Classification of energy consumption in buildings with outlier detection," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3639–3644, 2010.

[6] Z. O'Neill, X. Pang, M. Shashanka, P. Haves, and T. Bailey, "Model-based real-time whole building energy performance monitoring and diagnostics," *Journal of Building Performance Simulation*, vol. 7, no. 2, pp. 83–99, 2014.

[7] D. Sellers, H. Friedman, T. Haasl, N. Bourassa, and M. A. Piette, "High performance commercial building systems," 2003.

[8] A. Handbook, "Hvac applications," *ASHRAE Handbook, Fundamentals*, 2015.

[9] H. Dibowski, J. Ploennigs, and K. Kabitzsch, "Automated design of building automation systems," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3606–3613, 2010.

[10] T. Novak and A. Gerstinger, "Safety-and security-critical services in building automation and control systems," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3614–3621, 2010.

[11] M. Comstock, J. Braun, and E. Groll, "The sensitivity of chiller performance to common faults," *HVAC&R Research*, vol. 7, no. 3, pp. 263–279, 2001.

[12] B. Sun, P. B. Luh, Z. O'Neill, and F. Song, "Building energy doctors: Spc and kalman filter-based fault detection," in *IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, 2011, pp. 333–340.

[13] B. Sun, P. B. Luh, Q.-S. Jia, Z. O'Neill, and F. Song, "Building energy doctors: An spc and kalman filter-based method for system-level fault detection in hvac systems," *IEEE Transactions on Automation Science and Engineering.*, vol. 11, no. 1, pp. 215–229, 2014.

[14] D. Li, G. Hu, and C. J. Spanos, "A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis," *Energy and Buildings*, vol. 128, pp. 519–529, 2016.

[15] D. Li, Y. Zhou, G. Hu, and C. J. Spanos, "Fault detection and diagnosis for building cooling system with a tree-structured learning method," *Energy and Buildings*, vol. 127, pp. 540–551, 2016.

[16] ——, "Fusing system configuration information for building cooling plant fault detection and severity level identification," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, Conference Proceedings, pp. 1319–1325.

[17] ——, "Optimal sensor configuration and feature selection for ahu fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 127, pp. 540–551, 2016.

[18] Q. Zhou, S. Wang, and Z. Ma, "A modelbased fault detection and diagnosis strategy for hvac systems," *International Journal of Energy Research*, vol. 33, no. 10, pp. 903–918, 2009.

[19] Z. Du and X. Jin, "Detection and diagnosis for sensor fault in hvac systems," *Energy Conversion and Management*, vol. 48, no. 3, pp. 693–702, 2007.

[20] S. Wang, Q. Zhou, and F. Xiao, "A system-level fault detection and diagnosis strategy for hvac systems involving sensor faults," *Energy and Buildings*, vol. 42, no. 4, pp. 477–490, 2010.

[21] S. Wang and J. Cui, "A robust fault detection and diagnosis strategy for centrifugal chillers," *HVAC&R Research*, vol. 12, no. 3, pp. 407–428, 2006.

[22] H. Chilton, "Performance of natural-draught water-cooling towers," *Proceedings of the IEE-Part II: Power Engineering*, vol. 99, no. 71, pp. 440–452, 1952.

[23] Y. Zhao, S. Wang, and F. Xiao, "A system-level incipient fault-detection method for hvac systems," *HVAC&R Research*, vol. 19, no. 5, pp. 593–601, 2013.

[24] S. Katipamula and M. R. Brambley, "Review article: Methods for fault detection, diagnostics, and prognostics for building systemsa review, part ii," *HVAC&R Research*, vol. 11, no. 2, pp. 169–187, 2005.

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

[26] Y. Yu, D. Woradechjumroen, and D. Yu, "A review of fault detection and diagnosis methodologies on air-handling units," *Energy and Buildings*, vol. 82, pp. 550–562, 2014.

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

[28] Z. Gao, C. Cecati, and S. Ding, "A survey of fault diagnosis and fault-tolerant techniques part ii: Fault diagnosis with knowledge-based and hybrid/active approaches," *IEEE Transactions on Industrial Electronics*, 2015.

[29] C. Li, F. Meggers, M. Li, J. Sundaravaradan, F. Xue, H. Lim, and A. Schlueter, "Bubblesense: wireless sensor network based intelligent building monitoring," *Proceedings of the ICT4S*, pp. 159–166, 2013.

[30] D. J. Cook and S. K. Das, "How smart are our environments? an updated look at the state of the art," *Pervasive and mobile computing*, vol. 3, no. 2, pp. 53–73, 2007.

[31] A. Purarjomandlangrudi, A. H. Ghapanchi, and M. Esmalifalak, "A data mining approach for fault diagnosis: An application of anomaly detection algorithm," *Measurement*, vol. 55, pp. 343–352, 2014.

[32] M. Bruelisauer, K. W. Chen, R. Iyengar, H. Leibundgut, C. Li, M. Li, M. Mast, F. Meggers, C. Miller, and D. Rossi, "Bubblezerodesign, construction and operation of a transportable research laboratory for low exergy building system evaluation in the tropics," *Energies*, vol. 6, no. 9, pp. 4551–4571, 2013.

[33] M. Comstock and J. Braun, "Fault detection and diagnostic (fdd) requirements and evaluation tools for chillers," *West Lafayette, IN: ASHRAE*, 2002.

[34] J. Wen and S. Li, "Tools for evaluating fault detection and diagnostic methods for air-handling units," *Philadelphia, PA: Drexel University. ASHRAE*, 2011.

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

[36] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part i: Quantitative model-based methods," *Computers & chemical engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[37] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies," *Computers & Chemical Engineering*, vol. 27, no. 3, pp. 313–326, 2003.

[38] J. M. House and G. E. Kelly, "An overview of building diagnostics," in *National Conference on Building Commissioning, Kansas City, MO*, Conference Proceedings.

[39] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*.

[40] A. Dexter and J. Pakanen, *Demonstrating automated fault detection and diagnosis methods in real buildings*. Technical Research Centre of Finland (VTT), 2001.

[41] L. K. Norford, J. A. Wright, R. A. Buswell, D. Luo, C. J. Klaassen, and A. Suby, "Demonstration of fault detection and diagnosis methods for air-handling units," *HVAC&R Research*, vol. 8, no. 1, pp. 41–71, 2002.

[42] B. Price and T. Smith, "Development and validation of optimal strategies for building hvac systems," Technical Report: ME-TEF-03-001, Department of Mechanical Engineering, The University of Iowa, Iowa City, Iowa, Tech. Rep., 2003.

[43] J. J. Gertler, "Survey of model-based failure detection and isolation in complex plants," *IEEE Control systems magazine*, vol. 8, no. 6, pp. 3–11, 1988.

[44] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, 1990.

[45] ——, "Analytical and qualitative model-based fault diagnosisa survey and some new results," *European Journal of control*, vol. 2, no. 1, pp. 6–28, 1996.

[46] R. Isermann and P. Ball, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control engineering practice*, vol. 5, no. 5, pp. 709–719, 1997.

[47] R. Isermann, "Model-based fault-detection and diagnosisstatus and applications," *Annual Reviews in control*, vol. 29, no. 1, pp. 71–85, 2005.

[48] S. Ding, *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.

[49] Z. Hou, M. Noori, and R. S. Amand, "Wavelet-based approach for structural damage detection," *Journal of Engineering Mechanics*, vol. 126, no. 7, pp. 677–683', year =.

[50] Z. Peng and F. Chu, "Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography," *Mechanical systems and signal processing*, vol. 18, no. 2, pp. 199–221, 2004.

[51] S. Yin, G. Wang, and H. R. Karimi, "Data-driven design of robust fault detection system for wind turbines," *Mechatronics*, vol. 24, no. 4, pp. 298–306, 2014.

[52] E. L. Russell, L. H. Chiang, and R. D. Braatz, *Data-driven methods for fault detection and diagnosis in chemical processes*. Springer Science & Business Media, 2012.

[53] H. Yoon and J.-H. Jang, "Ongoing energy fault detection using a data-driven chiller performance prediction model," in *7th International Conference on Computing and Convergence Technology (ICCCT), 2012*. IEEE, Conference Proceedings, pp. 866–869.

[54] S. Wang and J. Cui, "Sensor-fault detection, diagnosis and estimation for centrifugal chiller systems using principal-component analysis method," *Applied Energy*, vol. 82, no. 3, pp. 197–213, 2005.

[55] Y. Chen and L. Lan, "A fault detection technique for air-source heat pump water chiller/heaters," *Energy and Buildings*, vol. 41, no. 8, pp. 881–887, 2009.

[56] Y. Hu, H. Chen, J. Xie, X. Yang, and C. Zhou, "Chiller sensor fault detection using a self-adaptive principal component analysis method," *Energy and buildings*, vol. 54, pp. 252–258, 2012.

[57] S. Wang and F. Xiao, "Ahu sensor fault diagnosis using principal component analysis method," *Energy and Buildings*, vol. 36, no. 2, pp. 147–160, 2004.

[58] Z. Du and X. Jin, "Multiple faults diagnosis for sensors in air handling unit using fisher discriminant analysis," *Energy Conversion and Management.*, vol. 49, no. 12, pp. 3654–3665, 2008.

[59] Z. Du, X. Jin, and L. Wu, "Fault detection and diagnosis based on improved pca with jaa method in vav systems," *Building and Environment*, vol. 42, no. 9, pp. 3221–3232, 2007.

[60] Z. Du, X. Jin, and Y. Yang, "Fault diagnosis for temperature, flow rate and pressure sensors in vav systems using wavelet neural network," *Applied Energy*, vol. 86, no. 9, pp. 1624–1631, 2009.

[61] Z. Du, X. Jin, and X. Yang, "A robot fault diagnostic tool for flow rate sensors in air dampers and vav terminals," *Energy and Buildings*, vol. 41, no. 3, pp. 279–286, 2009.

[62] S. Wu and J. Sun, "A top-down strategy with temporal and spatial partition for fault detection and diagnosis of building hvac systems," *Energy and Buildings*, vol. 43, no. 9, pp. 2134–2139, 2011.

[63] S. Wu and J.-Q. Sun, "Cross-level fault detection and diagnosis of building hvac systems," *Building and Environment*, vol. 46, no. 8, pp. 1558–1566, 2011.

[64] S. Li and J. Wen, "A model-based fault detection and diagnostic methodology based on pca method and wavelet transform," *Energy and Buildings*, vol. 68, pp. 63–71, 2014.

[65] H. Wang, Y. Chen, C. W. Chan, and J. Qin, "An online fault diagnosis tool of vav terminals for building management and control systems," *Automation in Construction*, vol. 22, pp. 203–211, 2012.

[66] G. Mustafaraj, J. Chen, and G. Lowry, "Development of room temperature and relative humidity linear parametric models for an open office using bms data," *Energy and Buildings*, vol. 42, pp. 348–356, Aug. 2010.

[67] D. J. Hill, B. S. Minsker, and E. Amir, "Real-time bayesian anomaly detection for environmental sensor data," in *Proceedings of the Congress-International Association for Hydraulic Research*, vol. 32. Citeseer, 2007, p. 503.

[68] Y. Zhao, F. Xiao, and S. Wang, "An intelligent chiller fault detection and diagnosis methodology using bayesian belief network," *Energy and Buildings*, vol. 57, pp. 278–288, 2013.

[69] F. Xiao, Y. Zhao, J. Wen, and S. Wang, "Bayesian network based fdd strategy for variable air volume terminals," *Automation in Construction*, vol. 41, pp. 106–118, 2014.

[70] B. Fan, Z. Du, X. Jin, X. Yang, and Y. Guo, "A hybrid fdd strategy for local system of ahu based on artificial neural network and wavelet analysis," *Building and environment*, vol. 45, no. 12, pp. 2698–2708, 2010.

[71] Y. Zhu, X. Jin, and Z. Du, "Fault diagnosis for sensors in air handling unit based on neural network pre-processed by wavelet and fractal," *Energy and buildings*, vol. 44, pp. 7–16, 2012.

[72] Z. Du, B. Fan, X. Jin, and J. Chi, "Fault detection and diagnosis for buildings and hvac systems using combined neural networks and subtractive clustering analysis," *Building and Environment*, vol. 73, pp. 1–11, 2014.

[73] P. Jaikumar, A. Gacic, B. Andrews, and M. Dambier, "Detection of anomalous events from unlabeled sensor data in smart building environments," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 2268–2271.

[74] Y. Zhao, S. Wang, and F. Xiao, "Pattern recognition-based chillers fault detection method using support vector data description (svdd)," *Applied Energy.*, vol. 112, pp. 1041–1048, 2013.

[75] Y. Zhao, F. Xiao, J. Wen, Y. Lu, and S. Wang, "A robust pattern recognition-based fault detection and diagnosis (fdd) method for chillers," *HVAC&R Research*, vol. 20, no. 7, pp. 798–809, 2014.

[76] J. Liang and R. Du, "Model-based fault detection and diagnosis of hvac systems using support vector machine method," *International Journal of refrigeration*, vol. 30, no. 6, pp. 1104–1114, 2007.

[77] H. Han, Z. Cao, B. Gu, and N. Ren, "Pca-svm-based automated fault detection and diagnosis (afdd) for vapor-compression refrigeration systems," *HVAC & R Research*, vol. 16, no. 3, pp. 295–313, 2010.

[78] K.-Y. Chen, L.-S. Chen, M.-C. Chen, and C.-L. Lee, "Using svm based method for equipment fault detection in a thermal power plant," *Computers in industry*, vol. 62, no. 1, pp. 42–50, 2011.

[79] K. Yan, W. Shen, T. Mulumba, and A. Afshari, "Arx model based fault detection and diagnosis for chillers using support vector machines," *Energy and Buildings*, vol. 81, pp. 287–295, 2014.

[80] T. Mulumba, A. Afshari, K. Yan, W. Shen, and L. K. Norford, "Robust model-based fault diagnosis for air handling units," *Energy and Buildings.*, vol. 86, pp. 698–707, 2015.

[81] D. Dietrich, D. Bruckner, G. Zucker, and P. Palensky, "Communication and computation in buildings: A short introduction and overview," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3577–3584, 2010.

[82] K. Yoshida, M. Inui, T. Yairi, K. Machida, M. Shioya, and Y. Masukawa, "Identification of causal variables for building energy fault detection by semi-supervised lda and decision boundary analysis," in *2008 IEEE International Conference on Data Mining Workshops*. IEEE, 2008, Conference Proceedings, pp. 164–173.

[83] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 104.

[84] S. Dumais and H. Chen, "Hierarchical classification of web content," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2000, pp. 256–263.

[85] L. Cai and T. Hofmann, "Hierarchical document categorization with support vector machines," in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM, 2004, pp. 78–87.

[86] S. Li, J. Wen, X. Zhou, and C. J. Klaassen, "Development and validation of a dynamic air handling unit model, part 1 (rp-1312)," *ASHRAE Transactions*, vol. 116, no. 1, p. 45, 2010.

[87] ——, "Development and validation of a dynamic air handling unit model, part 2 (rp-1312)," *ASHRAE Transactions*, vol. 116, no. 1, p. 57, 2010.

[88] S. Verron, T. Tiplica, and A. Kobi, "Fault detection and identification with a new feature selection based on mutual information," *Journal of Process Control*, vol. 18, no. 5, pp. 479–490, 2008.

[89] B. Chebel-Morello, S. Malinowski, and H. Senoussi, "Feature selection for fault detection systems: application to the tennessee eastman process," *Applied Intelligence*, vol. 44, no. 1, pp. 111–122, 2016.

[90] F. Xiao, S. Wang, X. Xu, and G. Ge, "An isolation enhanced pca method with expert-based multivariate decoupling for sensor fdd in air-conditioning systems," *Applied Thermal Engineering*, vol. 29, no. 4, pp. 712–722, 2009.

[91] Y. Zhang, W. Du, Y. Fan, and L. Zhang, "Process fault detection using directional kernel partial least squares," *Industrial & Engineering Chemistry Research*, vol. 54, no. 9, pp. 2509–2518, 2015.

[92] S. Wang and J. Cui, "Sensor-fault detection, diagnosis and estimation for centrifugal chiller systems using principal-component analysis method," *Applied Energy*, vol. 82, no. 3, pp. 197–213, 2005.

[93] Q. Zhou, S. Wang, and F. Xiao, "A novel strategy for the fault detection and diagnosis of centrifugal chiller systems," *HVAC&R Research*, vol. 15, no. 1, pp. 57–75, 2009.

[94] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.

[95] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 6.

[96] G. Trunk, "A problem of dimensionality: A simple example," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 3, pp. 306–307, 1979.

[97] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, *On the surprising behavior of distance metrics in high dimensional space*. Springer, 2001.

[98] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, and J. Weng, *Discriminant analysis of principal components for face recognition*. Springer, 1998, pp. 73–85.

[99] M. Welling, "Fisher linear discriminant analysis," *Department of Computer Science, University of Toronto*, vol. 3, 2005.

[100] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[101] J. Fan and Y. Fan, "High dimensional classification using features annealed independence rules," *Annals of statistics*, vol. 36, no. 6, p. 2605, 2008.

[102] Y. Jia, "Model-based generic approaches for automated fault detection, diagnosis, evaluation (fdde) and for accurate control of field-operated centrifugal chillers," Ph.D. dissertation, 2002.

[103] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *The Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2002.

[104] K. Wang, S. Zhou, and S. C. Liew, "Building hierarchical classifiers using class proximity," in *25th International Conference on Very Large Data Bases*, vol.

Proceedings of VLDB-99. Morgan Kaufmann Publishers,San Francisco, US, 1999, pp. 363–374.

[105] R. Bellman, "Dynamic programming and lagrange multipliers," *Proceedings of the National Academy of Sciences*, vol. 42, no. 10, pp. 767–769, 1956.

[106] J. C. Platt, "Using analytic qp and sparseness to speed training of support vector machines," *Advances in neural information processing systems*, pp. 557–563, 1999.

[107] S. Fine and K. Scheinberg, "Efficient svm training using low-rank kernel representations," *The Journal of Machine Learning Research*, vol. 2, pp. 243–264, 2002.

[108] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[109] Y. Zhou, J. Y. Baek, D. Li, and C. J. Spanos, *Optimal Training and Efficient Model Selection for Parameterized Large Margin Learning*. Springer, 2016, pp. 52–64.

[110] X. Li, C. P. Bowers, and T. Schnier, "Classification of energy consumption in buildings with outlier detection," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3639–3644, 2010.

[111] J. M. Cimbala, "Modified thompson tau used for determination of outliers," *Penn State University*, 2011.

[112] D. R. Cox and D. V. Hinkley, *Theoretical statistics*. CRC Press, 1979.

[113] H. Xie, L. E. Pierce, and F. T. Ulaby, "Sar speckle reduction using wavelet denoising and markov random field modeling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 10, pp. 2196–2212, 2002.

[114] Y. Saeys, I. Inza, and P. Larraaga, "A review of feature selection techniques in bioinformatics," *bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[115] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[116] P. Estvez, M. Tesmer, C. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *Neural Networks, IEEE Transactions on*, vol. 20, no. 2, pp. 189–201, 2009.

[117] E. Yu and S. Cho, "Ensemble based on ga wrapper feature selection," *Computers & Industrial Engineering*, vol. 51, no. 1, pp. 111–116, 2006.

[118] L. Paninski, "Estimation of entropy and mutual information," *Neural computation*, vol. 15, no. 6, pp. 1191–1253, 2003.

[119] Z. Abrams, A. Goel, and S. Plotkin, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 424–432.

[120] A. Krause and D. Golovin, "Submodular function maximization," *Tractability: Practical Approaches to Hard Problems*, vol. 3, p. 19, 2012.

[121] A. Schrijver, "A combinatorial algorithm minimizing submodular functions in strongly polynomial time," *Journal of Combinatorial Theory, Series B*, vol. 80, no. 2, pp. 346–355, 2000.

[122] Y. Zhao, J. Wen, and S. Wang, "Diagnostic bayesian networks for diagnosing air handling units faultspart ii: Faults in coils and sensors," *Applied Thermal Engineering*, vol. 90, pp. 145–157, 2015.

[123] S. Li and J. Wen, "Application of pattern matching method for detecting faults in air handling unit system," *Automation in Construction*, vol. 43, pp. 49–58, 2014.

[124] C.-W. Ko, J. Lee, and M. Queyranne, "An exact algorithm for maximum entropy sampling," *Operations Research*, vol. 43, no. 4, pp. 684–691, 1995.

[125] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *Journal of Artificial Intelligence Research*, vol. 42, pp. 427–486, 2011.

[126] Y. Zhou, D. Li, and C. J. Spanos, "Causal meets submodular: Subset selection with directed information," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[127] V. Cevher and A. Krause, "Greedy dictionary selection for sparse representation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 979–988, 2011.

[128] D. K. Abhimanyu Das, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection." *Proc. of ICML 2011, Seattle, WA*, 2011.

[129] Y. Kawahara and T. Washio, "Prismatic algorithm for discrete dc programming problem," in *Advances in Neural Information Processing Systems*, 2011, pp. 2106–2114.

[130] M. Narasimhan and J. A. Bilmes, "A submodular-supermodular procedure with applications to discriminative structure learning," *arXiv preprint arXiv:1207.1404*, 2012.

[131] S. Li and J. Wen, "Application of pattern matching method for detecting faults in air handling unit system," *Automation in Construction*, vol. 43, pp. 49–58, 2014.

[132] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint 2, 1-norms minimization." Advances in neural information processing systems, 2010, pp. 1813–1821.

[133] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[134] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2012.