ABSTRACT

Title of Dissertation: MACHINERY ANOMALY DETECTION UNDER INDETERMINATE OPERATING CONDITIONS

Jing Tian, Doctor of Philosophy, 2018

Dissertation directed by: George E. Dieter Professor, Michael Pecht, Department of Mechanical Engineering

Anomaly detection is a critical task in system health monitoring. Current practice of anomaly detection in machinery systems is still unsatisfactory. One issue is with the use of features. Some features are insensitive to the change of health, and some are redundant with each other. These insensitive and redundant features in the data mislead the detection. Another issue is from the influence of operating conditions, where a change in operating conditions can be mistakenly detected as an anomalous state of the system. Operating conditions are usually changing, and they may not be readily identified. They contribute to false positive detection either from non-predictive features driven by operating conditions, or from influencing predictive features. This dissertation contributes to the reduction of false detection by developing methods to select predictive features and use them to span a space for anomaly detection under indeterminate operating conditions.

Available feature selection methods fail to provide consistent results when some features are correlated. A method was developed in this dissertation to explore the correlation structure of features and group correlated features into the same clusters. A representative feature from each cluster is selected to form a non-correlated set of features, where an optimized subset of predictive features is selected. After feature selection, the influence of operating conditions through non-predictive variables are removed. To remove the influence on predictive features, a clustering-based anomaly detection method is developed. Observations are collected when the system is healthy, and these observations are grouped into clusters corresponding to the states of operating conditions with automatic estimation of clustering parameters. Anomalies are detected if the test data are not members of the clusters. Correct partitioning of clusters is an open challenge due to the lack of research on the clustering of the machinery health monitoring data. This dissertation uses unimodality of the data as a

criterion for clustering validation, and a unimodality-based clustering method is developed.

Methods of this dissertation were evaluated by simulated data, benchmark data, experimental study and field data. These methods provide consistent results and outperform representatives of available methods. Although the focus of this dissertation is on the application of machinery systems, the methods developed in this dissertation can be adapted for other application scenarios for anomaly detection, feature selection, and clustering.

MACHINERY ANOMALY DETECTION UNDER INDETERMINATE OPERATING CONDITIONS

by

Jing Tian

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Professor Michael Pecht, Chair
Dr. Michael H. Azarian
Professor Abhijit Dasgupta
Professor F. Patrick McCluskey
Professor Peter Sandborn
Professor Yunfeng Zhang

Acknowledgements

I would like to thank my advisor Prof. Michael Pecht. His guidance and training let me grow to a critical thinker in research and an enlightened person in life. I would like to express my gratitude to my co-advisor Dr. Michael H. Azarian. Without his sacrifice of time and patient guidance I would never master the skills and techniques to complete my research. I am very grateful to my committee members, Prof. Abhijit Dasgupta, Prof. F. Patrick McCluskey, Prof. Peter Sandborn, and Prof. Yunfeng Zhang for their sacrifice of time to help me to complete my dissertation. I would also like to express my appreciation to Dr. Michael Krein and Janet Wedgewood from Lockheed Martin, Dr. William Hardman and Dr. Jason Hines from NAVAIR for their support and inspiration. I would like to thank my colleagues at CALCE. Especially, Dr. Carlos Morillo and Dr. Myeongsu Kang have given me constant help in solving problems.

I owe a lot to my family. Thank you, dear mother, for your understanding. Thank you, dear father. The faulty bearings and rolling elements you tested were the earliest seeds of this dissertation. A special note to Petermao. Thank you for your heroism in the earthquake.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

1-SVM	One-Class Support Vector Machine
ANN	Artificial Neural Network
BPNN	Back Propagation Neural Networks
CALCE	Center for Advanced Life Cycle Engineering
CBM	Condition-Based Maintenance
cdf	Cumulative Distribution Function
CTC	Correlation Tree-Cut
DBN	Deep Belief Networks
EEMD	Ensemble Empirical Mode Decomposition
EMD	Empirical Mode Decomposition
FMMEA	Failure Mode Mechanisms and Effect Analysis
GA	Genetic Algorithm
GC	Gap-Based Clustering
GMM	Gaussian Mixture Model
HDN	Hierarchical Diagnosis Network
HMM	Hidden Markov Model
ISVM-BT	Improved Support Vector Machine-based Binary Tree
KNN	K-Nearest Neighbor
KS	Kolmogorov–Smirnov
LDA	Linear Discriminant Analysis
LR	Logistic Regression

MD	Mahalanobis Distance
MLP	Multi-Layer Perceptron
mRMR	Max-Relevancy Min-Redundancy
PCA	Principal Component Analysis
pdf	Probability Density Function
PHM	Prognostics and Health Management
PoF	Physics-of-Failure
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machines
RCF	Rolling Contact Fatigue
RF	Random Forest
RFE	Recursive Feature Elimination
rms	Root-Mean-Square
RUL	Remaining Useful Life
SC	Silhouette-Based Clustering
SI	Silhouette Index
SOMs	Self-Organizing Maps
SS	Stability Selection
std	Standard Deviation
SVM	Support Vector Machine
TR-LDA	Trace Ratio Linear Discriminant Analysis
UC	Unimodality-Based Clustering

UCAD	Unimodality Clustering Anomaly Detection
WPNE	Wavelet Packet Node Energy
WPT	Wavelet Packet Transform

Chapter 1: Introduction

Machinery such as bearings, gears, and shafts are widely used in electromechanical systems including electric motors, generators, pumps, turbines, and fans. They usually play critical roles in the systems. For example, in a typical wind turbine, the kinetic energy of wind is captured by blades to rotate the main shaft, which is constrained to rotate in the desired direction by a main bearing. The rotation of the main shafts converted by a gearbox to get the desired rotation speed and torque to drive the generator, completing the conversion of kinetic energy of wind to the electrical energy. The drivetrain of a typical wind turbine is shown in Figure 1. A pump works on the opposite direction that the electrical energy is converted to the rotation of the pump shaft by an electric motor. The rotation of the shaft is constrained to the desired direction by pump bearings and it is converted to the desired rotation speed and torque by a gearbox. The output rotation drives the impeller to lift the liquid. In both examples, the failure of any bearing, shaft, or gear breaks the required energy transmission, resulting in system failures. These machinery components play similar critical roles in other systems as well, such as cooling fans, gas turbines, hydro turbines, steam turbines, vehicle powertrains, and machine tools.

Although machinery components have been used in various industrial sectors since the first industrial revolution, their reliability has remained an issue of research focus due to multiple factors. One is the criticality of machinery failures, as machinery components failures often lead to system failures. Another factor is the frequency of

machinery failures, which are widespread in some applications. For example, bearing failures account for more than 40% of the system failures of induction motors [2]. A third factor is the system downtime caused by the machinery failures. In wind turbines, gearbox failure is the top contributor of system downtime [3] due to the time spent on the diagnosis and replacement of the failed gearbox.

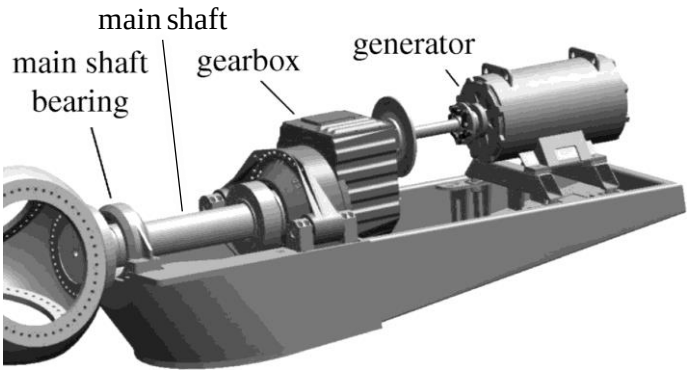

Figure 1: Wind Turbine Drivetrain [1]

1.1 Background

When a machinery component fails, corrective maintenance is carried out to detect the failed component and restore its reliability. However, because the maintenance is carried out after the occurrence of failure, it cannot avoid some failure consequences. Because the maintenance time is unexpected, logistics may not be ready. Since a machinery component usually takes time to degrade from a healthy state to failure, taking maintenance measures before the occurrence of failure is a desired strategy to avoid failures. Because the time of the failure is unknown, maintenance is planned to

perform regularly, leading to the development of scheduled maintenance, which is also called planned preventive maintenance. Scheduled maintenance has become a standard strategy to maintain the machinery reliability in various industry sectors because of several advantages over corrective maintenance. First, it restores the reliability of machinery components before failures and thus the occurrence of disastrous failure consequences can be reduced. Second, it avoids the unscheduled downtime from failures. Third, because the maintenance is scheduled, logistics can be prepared in advance. However, there are several shortcomings of scheduled maintenance. First, if a fault initiated and developed to failure between two maintenances, scheduled maintenance cannot detect it. Second, compared with maintenance plans depend solely on corrective maintenance, scheduled maintenance avoids unscheduled downtime, but it requires more maintenances, which increase maintenance cost and require more downtime. Third, when a component is replaced, it may have a significant portion of remaining useful life (RUL) left, and thus increases the cost on unnecessary component replacement. Finally, scheduled maintenance often involves intrusive measurements, increasing the risk of causing damage during the maintenance.

To overcome the shortcomings of the scheduled maintenance, the advent of failure should be predicted so the maintenance can be optimized accordingly. The prediction of failure needs in-situ monitoring, which becomes possible with the development of technologies such as sensing and computation. Based on in-situ monitoring, several maintenance strategies have been developed, including predictive maintenance and condition-based maintenance (CBM). They are improved further to form prognostics

and health management (PHM), which is an enabling discipline consisting of technologies and methods to assess the reliability of a product in its actual life cycle conditions to determine the advent of failure and mitigate system risk [4]. PHM aims to detect, diagnose, and predict the onset and source of system degradation as well as the time to system failure. The goal is to make intelligent decisions about the system health and to arrive at strategic and business case decisions [5].The implementation of PHM has two major approaches: physics-of-failure (PoF) approach, and data driven approach.

The PoF approach utilizes knowledge of hardware configurations and life cycle loading to predict the reliability and remaining useful life of the components [27]. The major inputs in respect of hardware configurations include material properties and machinery structure. The life cycle loading includes operational loads such as rotation speed, torque, duty cycles, and environmental loads include temperature, relative humidity, and pressure. PoF PHM involves conducting a failure mode mechanisms and effect analysis (FMMEA) to identify major failure mechanisms. The physics of failure models are developed for the identified failure mechanisms to establish functional relationships between the time to failure and various physical and stresses parameters. The application PoF PHM has following challenges. First, the development of physics of failure models is still an active research area that many failure mechanisms lack applicable models. Second, it is common that multiple failure mechanisms contribute to the failure together and the interaction between these failure mechanisms is often stochastic and is not modeled.

Data-driven PHM is an alternative to avoid the challenges in PoF PHM. Data-driven PHM does not require physics of failure models for any failure mechanisms, and it does not need detailed understanding of failure mechanisms. However, compared with PoF PHM, it requires larger amount of data and higher capability of computation. With the development and popularization of data acquisition and computation technologies, these limits on the application of data-driven PHM has been alleviated. In data-driven PHM, data are acquired in-situ using a network of sensors that monitors the system. Features carrying health information of the system are extracted from the sensor signals through a series of procedures like noise reduction, normalization, and transformation. Health states of the system are then estimated based on the extracted features via decision making using methods such as machine learning techniques. Based on the use of historic data, machine learning techniques can be classified as supervised learning techniques, and unsupervised learning techniques. In supervised learning, historic data with system health labels are used to train an algorithm to establish regions of different health states of the system. Current health state of the system is determined by classifying the current data to one of the regions. Widely used supervised learning techniques include support vector machine (SVM), *k*-nearest neighbor, artificial neural network, deep learning classifiers and regressors, decision tree, and random forest. Unsupervised learning techniques do not need data with health labels. They explore the nature of the data from different aspects. For example, unsupervised learning uses clustering techniques to partition the health monitoring data to multiple clusters according to the density of the data,

and the fault was detected by judging the density of the data. Widely used unsupervised learning techniques include *k*-means clustering, Gaussian mixture model, and self-organizing maps.

In both PoF PHM and data-driven PHM, fault detection is a major task across a wide range of applications, such as induction motor rotor bars [6], bearings [7], and gears[8]. A machinery fault is an abnormal condition that leads to the failure of the machinery, which is a state in which the machinery cannot perform its required function under stated conditions. Commonly observed faults in machinery include pits, indents, and wear in bearings, pits, root crack, wear, and missing teeth in gears, and misalignment, wear, and bent in shafts. In most cases, when a fault emerges, the machine can still perform its required function until the fault develops to a certain degree. For example, a main failure mechanism of rolling element bearing is rolling contact fatigue (RCF), which happens even if the bearing is working under stated conditions and is lubricated and maintained properly. In RCF, cracks initiate beneath or on the contacting surfaces because of stress concentration around deficiencies or material impurities and the cyclic loading from the rolling elements. As the cracks propagate, some material is removed, forming pits on the surface. At an early stage, the bearing can still work as required at the presence of these faults, which are cracks and pits. With the development of the cracks and pits, more material is lost, and the working profile of the bearing is changed to a degree that the bearing fails to work as required. In general, there is a time gap between the emergence of fault and the occurrence of failure. If the fault can be detected at an incipient stage and if its

development can be monitored, PHM can be performed to reduce the failure frequency and criticality as a result of optimized maintenance.

Machinery fault detection based on PoF PHM monitors the variables of PoF models and compares their monitored value to the calculated value from the model. When the deviation of the monitored value from the variables exceeds a predetermined threshold, the fault is detected. PoF fault detection requires in-depth knowledge of machinery failure mechanisms or fault characteristic signal-generating mechanisms [9] to construct PoF models, limiting the application range. Therefore, a lot of methods were developed for data-driven fault detection, which avoids the limitations of the PoF approach.

In data-driven PHM, fault detection is achieved by anomaly detection through learning the rules of detection from historical data [10]. Anomalies are patterns in the data that do not conform to a defined notion of normal behavior [11]. In machinery fault detection, the normal behavior is usually the distribution of healthy reference data, which are the health monitoring data collected when the machinery is healthy. When a system becomes faulty, the health monitoring data no longer conform to the normal behavior defined by the healthy reference data, and thus the behavior is considered anomalous. In sum, a fault is a physical state of the machinery, anomalies are the representations of the fault in the data, and anomaly detection is the process to identify the existence of fault through the detection of anomalies.

1.2 Motivation

Current practice of fault detection is not satisfactory that false detections, including both false positive detection and false negative detection, are causing losses. False positive detection is also termed type I error. It mistakenly regards healthy data as faulty, leading to unnecessary downtime and maintenance cost. False negative detection is also termed type II error. It mistakenly regards faulty data as healthy, and therefore it may leave catastrophic failures undetected. For example, in 2016, undetected gearbox bearing pitting and gear fatigue cracking led to the crash of an Airbus Helicopters H225, resulting in 13 deaths [12]. Therefore, it is necessary to investigate the problem and improve over available methods.

Two issues contribute to the false detection of anomalies: first, the health monitoring data usually contain insensitive and redundant features, and they mislead the detection. Features are often extracted using engineering experience about the system. Some of them are not sensitive to the faults of specific systems, and thus insensitive features exist. Some features are driven by the same underlying factor and they are redundant with each other. Useless and redundant information from insensitive and redundant features can mislead the detection. Especially, some features are more sensitive to the change of operating conditions than the change of system health. When operating conditions are changing, false positive detection may occur. Second, machinery operating conditions are usually changing due to multiple operation regimes and environmental influence. The changing operating conditions are often indeterminate because the operating conditions are often not monitored, and the

influence of some operating conditions is unknown. As a result, anomalous states can be confused with changed operating states. For example, driving on a road with unknown surface quality is a case of indeterminate operating condition. If vibration amplitude is used to monitor automobile engine health, a car with a faulty engine driving on a smooth road surface can be confused with a healthy engine when the car is driving on a rough road surface, as shown in Table 1. Both issues and available methods to address them need to be investigated.

Table 1: An Example of the Influence of Operating Conditions

Engine health	Operating condition	
	Smooth road surface	Rough road surface
Healthy	Low vibration amplitude	High vibration amplitude
Faulty	High vibration amplitude	High vibration amplitude

1.3 Overview of the Dissertation

The structure of the remaining dissertation is as follows. Chapter 2 discusses the issues related to features and anomaly detection methods and provides a literature review of available methods that address the issues. Based on the analysis of the literature, research gaps were identified, and objectives of the dissertation is proposed to fill the gaps. Chapter 3 introduces the development and evaluation of a feature selection method that works when insensitive and redundant features exist. Chapter 4 introduces the development and evaluation of a clustering method and an anomaly detection method based on it that works under indeterminate operating conditions. Chapter 5 presents the contributions of the dissertation and suggested future work.

Chapter 2: Literature Review

Inappropriate set of features and incompetent anomaly detection methods are two major issues of failed anomaly detection. Therefore, these two issues were investigated and available methods to address them were reviewed in this chapter to identify research gaps.

2.1 Issues Related to Features in Anomaly Detection

In-situ monitoring signals themselves are often inadequate for fault detection so features are extracted from them to capture the existence of fault. Ranging from 2012 to 2017, every year more than 20,000 papers about machinery fault features are published, estimated by searching using Google Scholar. Many features are insensitive to faults, and they are influenced by operating conditions. For example, shape factor and crest factor are established features used in rotating machinery fault detection [13]. In Figure 2, they were calculated to detect bearing faults. Rotation speed of the bearing was used as an operating condition. Figure 2 (Left) illustrates the insensitivity of features to a fault: under the same rotation speed, shape factor did not separate the data from the healthy bearing and faulty bearing, leading to false negative detection. Figure 2 (Right) illustrates the influence of operating conditions on features. Crest factor separated the data from a healthy bearing and those from a faulty bearing under 600 RPM. However, when the rotation speed was increased to

1200 RPM, a large portion of the data from the healthy bearing overlaps with the faulty data at 600 RPM, leading to false positive detection.

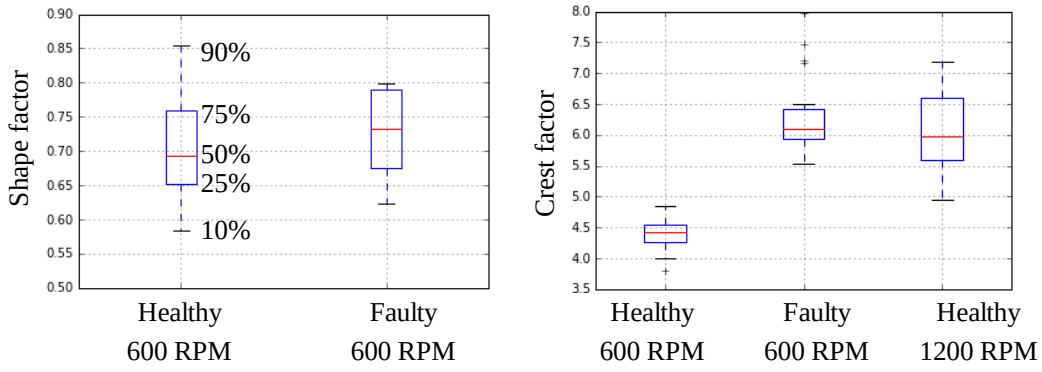

Figure 2: Detecting Bearing Fault Using Shape Factor (Left) and Crest Factor (Right)

Researchers have tried to increase the sensitivity of machinery anomaly detection by using multiple features together to perform multivariate analysis. Some researchers constructed features for specific failure modes and perform multivariate analysis based on them. Tian et al. [14] constructed 5 features, each of which is sensitive to a bearing failure mode. Some researchers used as many features as possible. Xia et al. [15] constructed 21 bearing fault features using signal processing techniques. Oh et al. [16] constructed 1000 bearing features using deep learning techniques. However, for a given failure mode, some features are insensitive to the fault. They behave as noise and have negative influence on the result. In Figure 3 (left), a benchmark dataset Iris Data was used to demonstrate the influence of noise features. Iris data have 4 features and 3 classes. Additional features consist of Gaussian noise were appended to the original data as noise features. Logistic regression was applied to perform classification. The performance of classification was evaluated by 5-fold cross-

validation, and the mean accuracy is used as a performance measure. When the number of noise features increases, the mean accuracy has a decreasing trend. Therefore, feature selection is necessary to select useful features that are capable of separating anomalies from normal data. Besides noise features, redundant features also have negative influence on the analysis. Redundant features are the features linearly correlated with each other. Some features are redundant with each other because they are measuring the same dynamics of the data. For example, the vibration signals collected by two accelerometers on the same surface of a gearbox chassis are redundant features. Peak-to-peak value and rms from vibration signals are likely correlated with each other. In Figure 3 (right), redundant features were appended to the Iris data. The redundant features are linear transforms of the original Iris data features with random constant terms. As the number of redundant features increases, the accuracy decreases.

In multivariate analysis, variables are sometimes termed as attributes, dimensions, or features. One observation of variables is also called a point or an object. In some cases, sensor signals or other forms of raw data are directly used in anomaly detection. To be consistent with academic and industrial conventions, the selection of variables is called feature selection, no matter the variables are attributes of raw data or features.

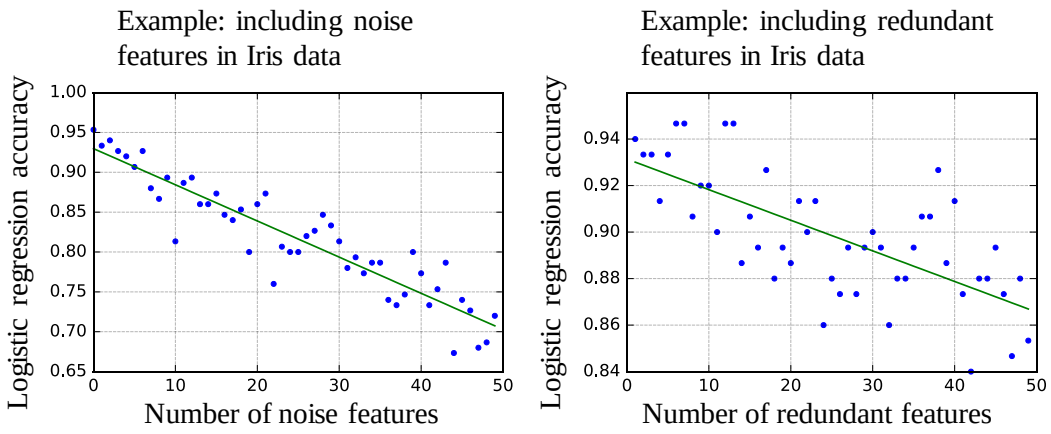

Figure 3: The Influence of Noise and Redundant Features

2.2 Survey of Feature Selection Methods

Available feature selection methods can be categorized as filter methods, wrapper methods, embedded methods, and ensemble methods. Filter methods select features based on the individual features' properties towards the objective of the specified machine learning task. For example, in binary classification, individual features are evaluated that any feature provides a certain degree of separation of the data independently is selected. One approach of filter methods is hypothesis testing. The null hypothesis is, the data from two classes are sampled from the same distribution using a specific feature. If the null hypothesis is rejected, the data from the two classes are not regarded as from the same distribution, and it means the feature is able to separate the two classes. The hypothesis testing methods used in feature selection include *t*-distribution, *F*-distribution, and KS-distribution. Another approach selects features depending on the similarity between individual features and the response. Similarity measures used in feature selection include Pearson's correlation coefficient

[17] and mutual information [18]. The features selected by filter methods can be highly correlated and thus some selected features are redundant [17]. Therefore, a greedy search algorithm was developed in [19] to compensate this drawback. An objective function was used that only the feature maximizes the mutual information between the feature and the response while the mutual information between the selected feature and the subset of the so far selected features is a minimum. This type of optimization problem was summarized as max-relevancy, min-redundancy (mRMR) [20], and different optimization methods have been developed to solve the problem [21]. Filter methods select features based on the performance of individual features and thus they ignore the joint effect of data separation of some features. For example, when the data are only separable by a nonlinear function, all the variables in the nonlinear function should be selected as useful features but filter methods do not realize their relationships because they just evaluate one feature at a time.

Wrapper methods use a search procedure such as forward selection and backward elimination to search the optimal subset. Forward selection starts with an empty subset and it repeatedly includes one feature at a time into a subset that optimizes an objective function [22]. The procedure stops when a threshold on the objective function is reached. Backward elimination starts with all features and repeatedly removes features one at a time until a criterion is satisfied. In a complete search, the approach takes $O(n^2)$ calls of the machine learning algorithm, and thus it is impractical in computation. Therefore, heuristic search algorithms have been implemented. For example, simulated annealing [23], genetic algorithm (GA) [24] and particle swarm

optimization (PSO) [25] have been applied to search the optimized subset of features. These methods are not guaranteed to converge to the global optimum given finite iterations. Therefore, they are often simplified that a predefined order of selection is used to reduce the number of combinations for the feature subset. Representative method is recursive feature elimination (RFE), such as RFE support vector machine (RFE-SVM) [26][27].However, since a predefined order of selection is used, some combinations of features are not evaluated and the optimal subset of features may not be found.

Embedded approach incorporates feature selection as part of the training process of a machine learning algorithm. The idea is to rank the features according to their weights or importance assigned by the algorithm during training. For example, the hyperplane of a linear support vector machine (SVM) is the optimized linear model that maximizes the separation of the data from different classes. The features with larger absolute values of weight are the ones contribute more to the separating hyperplane, and they are the useful features. Representative methods include neural network pruning [28] and decision tree-based feature selection[29]. For example, Krishnakumari et al. [30] used decision tree to select features from a group of vibration features and then applied a fuzzy classifier to diagnose spur gear fault. Some embedded methods utilize regularization to penalize the size of feature set, such as Lasso regression using $L1$ regularization, Ridge regression using $L2$ regularization, and elastic net using both $L1$ and $L2$ regularization [31]. When some features are correlated, regularization-based methods suffer from the inconsistency of

selection, which is the randomly selection of features from correlated features.

Ensemble features selection approach was developed to aggregate the power of different feature selection procedures. In [32], SVM is used as the feature selection algorithm, and it is run on multiple bootstrap samples to generate an ensemble of feature sets and a subset of features is selected from the ensemble. To reduce the influence of the inconsistency of regularization-based methods, stability selection using the idea of ensemble selection was developed, such as randomized Lasso for regression and randomized logistic regression for classification [33]. By aggregating the feature selection power of decision trees, random forest has been implemented in feature selection [34]. In [35], multiple feature selection algorithms from filter, embedded and wrapped approaches were combined. Ensemble approach reduces the influence of correlated features but still suffer from instability. For example, in random forest, correlated variables are used interchangeably in the trees. As a result, the less relevant variables often replace the useful ones as selected features [36]. In randomized lasso and randomized logistic regression, when the size of a group of correlated features increases, the weights of the features in the group decreases, leading to incorrect model interpretation and misleading feature ranking [37]. Feature selection based on feature clustering [38] has been used for removing the correlation bias. However, this method only considers linear model since it depends on Lasso regression. Moreover, correlated features were averaged to make new features that the physical meaning of original features is lost. In [39], *k*-means clustering was improved to automatically estimate the feature groups. Because the clustering is based on an

Euclidean distance-type dissimilarity, the challenge from the correlation of features is not addressed. For features extracted by principal component analysis (PCA) and manifold learning techniques, correlation is not an issue because the extracted features are orthogonal to each other. However, similar to the shortcoming of [38], the features lose their physical meaning. Keeping the physical meaning of the original features is a desired property of feature selection methods because feature selection is often applied in choosing optimal sensor set and interpreting failure physics. These tasks cannot be completed without features with physical meanings.

2.2 Issues Related to Anomaly Detection Methods

When features capable of separating anomalies from normal data are extracted and selected, they are still under the influence of operating conditions. In an experimental study, an accelerometer was mounted on the housing of a healthy bearing to collect vibration signals. The rotation speed was used as an operating condition and it was changed several times, as shown in Figure 4 (Left). The vibration signal was affected, and it changed with the rotation speed, as in Figure 4 (Right). Without knowing the rotation speed, the increase of amplitude and frequency in the vibration signal can be mistakenly regarded as anomalies. Therefore, some researchers used the signals of the operating conditions to normalize the signals used in health monitoring. In the case of vibration analysis, Prof. R. B. Randall at the University of New South Wales advocates the use of order tracking [40], and it has been applied to normalize the frequency of the vibration signals using the rotation speed in wind turbine bearing fault diagnosis [41]. More generally, some researchers develop different healthy data

models for different states of operating conditions, such as multi-regime modeling PHM from the team of Prof. J. Lee at the University of Cincinnati [42]. Using the similar idea, Sammaknejad et al. [43] modeled observations around different process operating modes by different multivariate Student's *t*-distributions to describe different likelihoods of anomalies. Above methods require monitoring of operating conditions and thus they do not work when the operating conditions are indeterminate. Therefore, anomaly detection methods without the information of operating conditions are needed.

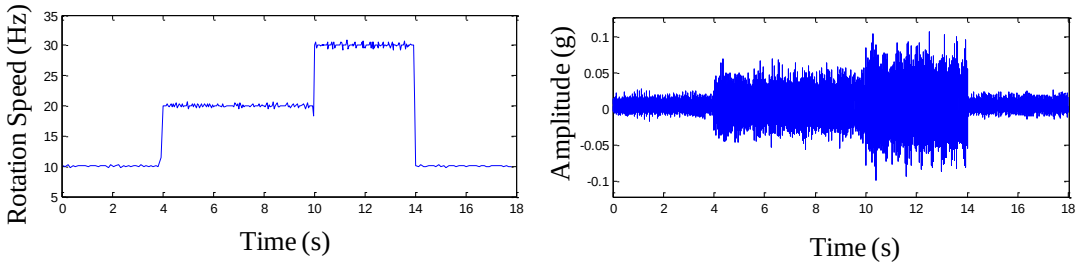

Figure 4: Influence of Rotation Speed (Left) on Vibration Signal (Right)

In the bearing example of Figure 4, a sliding window was applied to the vibration signal, and for each window an observation of features was calculated. In the example, rms (Dim 1) and Kurtosis (Dim 2) were calculated. These features span a feature space and certain patterns in the feature space can be observed, as shown in Figure 5. Therefore, machine learning can be used to detect anomalies without knowing operating conditions by learning these patterns and differentiate them from those of anomalies.

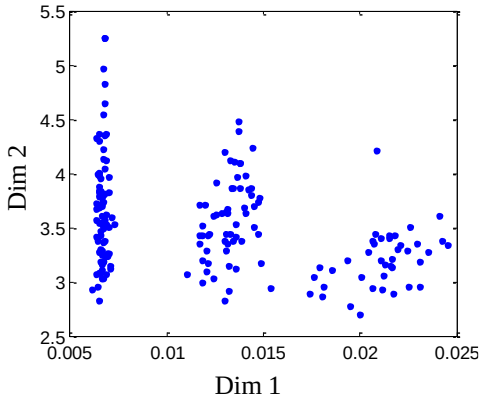

Figure 5: Patterns in the Feature Space

2.4 Survey of Anomaly Detection Methods

Anomaly detection methods using machine learning can be categorized as supervised anomaly detection [44], unsupervised anomaly detection [45][46][47], and semi-supervised anomaly detection [48][49]. Different methods have different requirements on the data and different application ranges.

2.4.1 Survey of Supervised Anomaly Detection

Supervised anomaly detection is based on statistical classification, which identifies to which of a set of classes a test observation belongs by learning the training data of these classes. Besides detecting the occurrence of the anomaly, supervised methods can also identify some properties of the anomaly, such as the type or location of the anomaly. Therefore, supervised methods are usually used for both the detection and diagnosis of anomalies. To implement supervised anomaly detection, training data for both healthy data and anomalies are required to be labeled. Classification techniques are used to set up decision rule from learning the labeled healthy data and anomalies to

determine if testing data are anomalies. The anomaly detection task is a binary classification problem. Multiple classification is implemented to determine the type, location, and severity of anomalies if corresponding labeled training data are available. Supervised methods used in machinery anomaly detection include support vector machine (SVM), *K*-nearest neighbor (KNN), artificial neural network (ANN), linear discriminant analysis (LDA), naïve Bayes classifier, hidden Markov model (HMM), logistic regression, decision tree, random forest, and hybrid ensemble learning classification algorithms.

In support vector machine (SVM), the data space is partitioned into normal regions and abnormal regions by hyperplanes [50]. If testing data fall into the abnormal region, they are detected as anomalies. Du et al.[51] constructed features using wavelet analysis and in the feature space bearing faults from different locations and severities were classified by SVM.As an improvement of SVM, Improved Support Vector Machine-based Binary Tree (ISVM-BT) was applied in [52][53] to classify different fault categories. ISVM-BT takes the advantage of both the efficient computation of the tree architecture and the high accuracy of SVM. However, users need to define the optimal hierarchy of the ISVM-BT, which lacks available rules. SVM methods work on both linear and nonlinear data, and they are robust against outliers. Their results are sensitive to the selection of kernel functions and parameters, but there is no efficient method for the selection. Because SVM is memory intensive and the search of the parameters requires multiple iterations of training and testing, SVM methods are not suitable for large datasets.

K-nearest neighbor (KNN) classifier compares the distance from a testing data observation to its nearest neighbors in the healthy training data and the anomalies training data. If the distance to the neighbors of anomalies training data is closer, it is classified as an anomaly. [54] extracted bearing features using principal component analysis (PCA), and then classifies the bearing health states using KNN. KNN is a nonparametric method that it does not pose any assumptions on the data and therefore it has a wide range of applications. However, KNN is sensitive to outliers. If irrelevant or redundant features are included, KNN may also generate undesired result.

Artificial neural network (ANN) simulate the structure of brain that multiple layers of neurons are interconnected to establish the relationship between the input and output. In the bearing fault detection and diagnosis, Ali et al. [55] used empirical mode decomposition (EMD) to extract features from vibration signals, and then used back propagation neural networks (BPNN) with two hidden layers to classify the anomalies. With the development and successful applications of deep learning techniques based on ANN, researchers began to explore the opportunity of implementing deep learning to machinery anomaly detection. Gan et al. [56] presented a deep learning application architecture for fault detection and fault severity classification using features constructed from the coefficients of wavelet packet decomposition. They developed a Hierarchical Diagnosis Network (HDN) based on Deep Belief Networks (DBN) that are constructed from multiple layers of Restricted Boltzmann Machines (RBM). The HDN consists of two layers with one for fault identification and another for severity

classification. ANN methods can provide the best classification accuracy among available classification algorithms. However, they need large amount of training data, and they are sensitive to outliers.

Discriminant analysis uses a discriminating function to perform classification. The optimal coefficients of the discriminating function are estimated that that the distance between classes is maximized, and the distance within the classes is minimized. The most widely used discriminating function is linear, leading to linear discriminant analysis (LDA). LDA assumes the data follow Gaussian distribution, but actual machinery data are usually non-Gaussian. Therefore, Jin et al. [57] uses trace ratio linear discriminant analysis (TR-LDA), which is a variant of LDA without the same normality requirement of the original LDA, to diagnose the bearing fault.

Bayesian network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph (DAG). Observations are classified according to their probabilities received from the network. A widely used type of Bayesian network is naïve Bayes classifier, which assumes the features are independent. Because the independence assumption is difficult to meet, Zhang et al. [58] used decision tree and selective SVM to select features with low correlations, and then applied naïve Bayesian classifier to perform fault diagnosis. Hidden Markov model (HMM) is another type of Bayesian network. It models the training data with Markov process with one hidden state. In [59] shift-invariant dictionary learning was used to extract features for bearing signals, and then HMM was applied to detect and

diagnose bearing faults. Because Bayesian network methods set strict assumptions on the distributions and the dependencies of the data, which are rarely met, their classification accuracy usually fail to match with other algorithms if they are properly tuned.

Logistic regression trains a logistic function with the labeled training data by optimizing a cost function, such as the negative log-likelihood of the true labels given the predictions. During testing, the membership probabilities of test observations are calculated. Authors in [60] used logistic regression to classify engine health. In [61], logistic regression was applied to the features extracted from wavelet packet decomposition for bearing fault diagnosis. Compared with algorithms such as ANN and SVM, the result from logistic regression has better probabilistic interpretation, and its training process is efficient. However, if the decision boundary is nonlinear, its classification performance fails to be compared with ANN and SVM.

Decision trees classify observations by sorting them based on feature values. Each node in a decision tree represents a feature in an observation to be classified, and each branch represents a value that the node can assume. Observation are classified starting at the root node and sorted based on their feature values. Because decision tree is working based on setting rules on features, its process facilitate intuitive understanding in engineering. In [62], decision tree was set up to detect machinery faults consist of mass imbalance, gear fault, and belt fault. Because classification algorithms such as

decision trees, ANN and KNN are sensitive to outliers, they were often used in an ensemble to make a consensus decision, giving rise to ensemble learning.

Ensemble learning combines the results of multiple classifiers to give a consensus decision. Because different classifiers have different sensitivity to outliers, ensemble learning is robust against overfitting. Random forest is an ensemble learning method that aggregates the estimation of a diverse set of decision trees To detect and diagnose bearing fault, Wang et al. [63] extracted features using wavelet packet transform, and then applied random forest in the feature space to perform classification. In the fault detection and diagnosis of spur gear in [64], an initial set of features from vibration signals were extracted from time domain, frequency domain, and wavelet transform. The classification performance of the subsets of these features were evaluated by a random forest classifier. The optimal subset was identified using genetic algorithm. Based on the selected subset of features, bearing health states were classified using random forest. Because diversity among classifiers is desired in ensemble learning, hybrid classifiers with an ensemble of classifiers induced from different classification algorithms were developed. To detect gear fault, Lei et al. [65] extracted features using envelope analysis, wavelet packet transform, and empirical decomposition, and then used a hybrid classifier consists of multi-layer perceptron (MLP) neural network, radial basis function (RBF) neural network, and KNN to perform the detection. Tian et al. [66] extracted commonly used features for rotating machinery from raw signals to form the base sample, and then bootstrap samples were generated. Each bootstrap sample was used to train a set of different classification algorithms to obtain a group

trained classifier. The final decision was obtained by majority voting from the groups of trained classifiers from all the bootstrap samples. Because this hybrid classifier has two sources of diversity from bootstrapping and different algorithms, it is less likely to provide estimation with high variance. Ensemble learning methods can provide the best classification accuracy, they are robust against outliers, and they do not set restrictions to the statistical properties of the data. However, there is no explicit rule to determine the setup of the ensemble, and the training process is usually not as efficient as other algorithms.

The application of supervised anomaly detection in machinery anomaly detection is limited by the requirement on the data. First, the requirement on the labeled faulty data as anomalous training data is difficult to meet. The training data from faulty systems are usually unavailable. For example, rotating machinery are critical components in safety-critical systems such as airplanes, and faulty data from these systems are unavailable. Some researchers have tried to fill this gap by simulating faulty data, as in the study of Tian et al. [14], where faulty bearing data are simulated using a faulty bearing signal generating model. In general, to simulate faulty data of a component, data generating models based on the physics of failure of the component are required. However, those models are often unavailable due to lack of modeling or inadequate understanding of the physics of failure. Second, even if faulty training data are collected in some cases, the sample size is not comparable to the data from the healthy systems because the faulty component either develops to failure or is processed by maintenance when it is noticed. Therefore, the training data from the healthy class and

the faulty class are not balanced. Although techniques such as upsampling of the faulty training data or downsampling of the healthy training data can alleviate the biased classification of the unbalanced data, the faulty training data only represent a portion of the faulty population so that the classification boundaries are still biased towards the healthy training data.

2.4.2 Survey of Unsupervised Anomaly Detection

Unsupervised anomaly detection avoids the requirement on the labeled data. In unsupervised anomaly detection, anomalies are detected using unlabeled data based on pre-defined assumptions that differentiate normal data and anomalies. A review of unsupervised anomaly detection is provided in [67]. Unsupervised anomaly detection starts with partitioning health monitoring data into clusters from the assumption that the data from healthy systems and the data from faulty systems are generated by different mechanisms and they form different clusters, as in the example in [68] that fuzzy *c*-means clustering identified multiple clusters in the feature space when the same number of bearing health states exist. In [69], noise Clustering and Density Oriented Fuzzy *c*-Means algorithms were used to eliminate outliers, and then kernel c-means algorithm with optimized parameters was applied to maximize the separation of clusters, which were assumed to be from different health states. An improved artificial ant clustering technique was applied in [70] to automatically group data consist of observations from healthy motor, motor with broken bar, and motor with faulty bearing. In [71], wavelet packet transform (WPT) and ensemble empirical mode decomposition (EEMD) were applied to extract features from bearing vibration signals,

and then an adaptive feature selection technique was developed to remove redundant features. Using these features, affinity propagation clustering was applied to partition the data into clusters, which are assumed to correspond to different bearing health states. Using vibration signals, Li et al. [72] extracted features using empirical wavelet transform and autoregressive model. After dimensionality reduction using locality preserving projection, fuzzy *c*-means was applied to cluster the data in the feature space. Observations from different health states were found to concentrate around certain centroids.

Above papers did not talk how to determine if certain clusters correspond to fault conditions to complete anomaly detection. To identify the clusters of anomalies, the properties of the clusters need to be evaluated based on pre-defined assumptions. One assumption is based on the densities of clusters. For a given observation, the radius of a hyper-sphere centered at the observation, which contains a defined number of other observations, is an estimate of the inverse of the density in the neighborhood of the observation. The mean value of the densities for all the observations in a cluster is used to represent the density of the cluster. Healthy systems are working in equilibrium states, and thus the health monitoring data form clusters with high densities. When the systems become faulty, they do not stay in the equilibrium states and thus the data are scattered more widely, forming clusters with lower densities. These clusters of different densities can be identified using density-based clustering. Tian et al. [73] applied a density-based clustering algorithm to partition bearing health monitoring data, and used the rate of change of the density as a threshold to detect

anomalies. A challenge to this approach is, under different operating conditions, the cluster densities of a healthy system can be different. For example, under higher rotation speed and higher load, the vibration of a bearing has higher variance that the features extracted from the vibration signal form clusters with lower densities. This kind of lower density clusters can be confused with anomaly clusters. A third assumption assumes the data from a healthy system and a faulty system are sampled from different statistical distributions. For example, if the data in the clusters from the healthy system are Gaussian, the non-Gaussian clusters consist of anomalies. There are some difficulties to implement this approach because machinery health monitoring data are usually high dimensional. First, the evaluation of the high-dimensional multivariate distributions, such as multivariate hypothesis testing, is difficult. Second, the data in a given cluster may not follow any known multivariate distribution. A widely used assumption is based on the sizes of clusters. The data from healthy systems are usually abundant, and anomalies are scarce. Because they are generated by different data generation mechanisms, they form different clusters, and the clusters of anomalies have smaller sizes. Clusters with sizes smaller than a certain threshold are regarded as consist of anomalies. This approach can be unreliable because smaller clusters can be generated by situations other than anomalies: if the system is working briefly in a state of operating conditions, a small cluster is formed even if the system is healthy. One assumption assumes anomalies are distributed in a manner that they do not form clusters by themselves. Therefore, only the data from healthy systems form clusters and the observations do not form any cluster are regarded as anomalies. A difficult to implement this approach is from the challenge of setting up threshold to

determine the cluster membership of observations. Moreover, it is common that anomalies form clusters by themselves, and thus violating the assumption.

Some researchers borrow the knowledge from the faulty data generating mechanisms in PoF for the identification of faulty clusters. *K*-means clustering was used to partition bearing data according to the bearing's health states in [74]. Initial centers of the *k*-means clustering are produced by simulated data. The initial cluster centers for the faulty bearing data were simulated using known faulty bearing data generating models, and the initial cluster centers for the healthy bearing data were simulated as white noise. The observations in the clusters developed from the initial centers of the simulated faulty data are regarded as faulty. Compared with simulating training data in supervised anomaly detection, this approach does not have strict requirement on the accuracy of the data generating models, because the initial cluster centers are only required to be close to the final centers, and *k*-means algorithm updates its value during iterations. However, after the update, the cluster center values may have significant changes that the initial cluster center, which represent certain health state, may not represent the final cluster, and thus false detection occurs. Moreover, the data generating model for many failure modes are not available. Inacio et al. [75] developed a recursive clustering approach for machinery fault diagnosis. At the beginning, only healthy data are available, and after the clustering, a rule was set up to identify the healthy cluster. When faulty occurred, the rule was updated that the faulty clusters were manually identified. Although training data were not formally labeled, knowledge from the actual machine condition was required to identify faulty clusters.

2.4.3 Survey of Semi-Supervised Anomaly Detection

The data from healthy systems are abundant, facilitating the application of semi-supervised anomaly detection, which uses labeled normal data to define the normal behavior and identifies observations deviating from the defined normal behavior as anomalies [76]. Statistical anomaly detection techniques assume that anomalies occur in the low probability regions of a stochastic model of the healthy data. These techniques rely on the assumption that the data follow certain distributions. However, real data may not follow these distributions. Representative work includes nonparametric statistical analysis [8]. Nearest neighbor-based anomaly detection techniques assume that anomalies occur far from the nearest neighbors in the healthy reference data. A representative method is *k*-nearest neighbor (KNN) [7]. Outliers in the healthy reference data may lead to false negative errors since they can be regarded as close neighbors by an anomaly. Also, these techniques do not consider the influence of the distribution of the data on anomaly detection.

Tian et al. [14] used KNN in a semi-supervised manner for motor bearing anomaly detection. At first, bearing fault features are extracted using spectral kurtosis, cross correlation, and principal component analysis. Then, KNN is applied to calculate the distance between the test observation to its nearest neighbors in the training data in the feature space. The KNN distance is used as an anomaly indicator that a hypothesis testing is applied on it to perform anomaly detection.

In [77], an one-class SVM model was trained by healthy bearing data, and during test the faulty bearing data were found outside of the model's decision boundaries. To reduce the influence of outliers in the healthy training data, one-class SVM was improved in [78] for the purpose of anomaly detection. In [79], recurrence time statistics of vibration signals were calculated as features to capture the information of incipient bearing fault, and one-class SVM was trained by healthy bearing data to set up healthy boundaries for anomaly detection.

KNN and one-class SVM methods are sensitive to the choice of hyper-parameters, as shown in Figure 6. Choosing the hyper-parameters requires labeled anomalies, which are often unavailable.

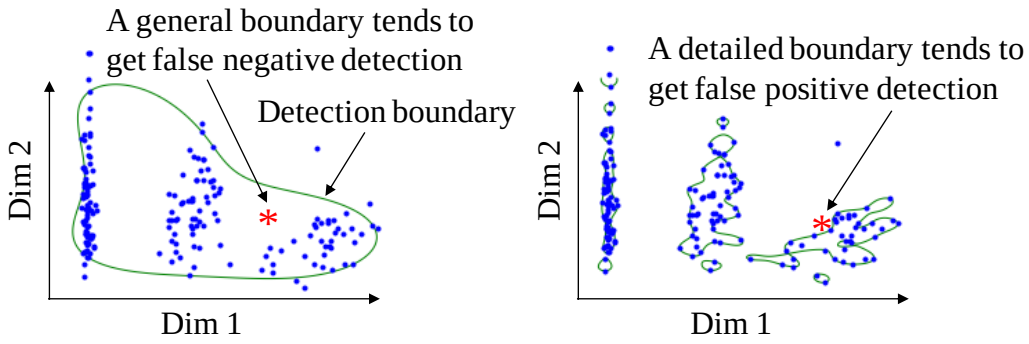

Figure 6: A General Boundary and a Detailed Boundary

Some researchers model the healthy data as a mixture of distributions, and anomalies are detected if test data are not generated by the mixture. In the bearing health assessment work of [80], features were extracted by locality preserving projections as

an improvement of feature extraction by principal component analysis (PCA). In the feature space a Gaussian mixture model (GMM) was trained by healthy bearing data. Kernel density estimation [81] was applied to setup the anomaly threshold. The anomaly score of the test data is the exponentially weighted moving average of the negative log likelihood probability referring to the trained GMM.

Because GMM is not application for many situations, other clustering techniques are used. Pan et al. [82] extracted bearing fault features using wavelet packet transform, and then applied fuzzy *c*-means to the healthy data in the feature space to identify the cluster centroid as healthy reference. The deviation from the reference is used as the degradation indicator. Although no faulty data were used in the calculation of the degradation indicator, the characterization of the values of the indicator still needs the centroid of the faulty data cluster. Huang et al. [83] developed a method based on self-organizing maps (SOMs) that vibration features from healthy bearings are used to train SOMs and anomaly score of a test observation is obtained by calculating the minimum quantization error of the observation referring to the trained maps. The robustness of SOMs in machinery anomaly detection is further improved by Tian et al. [84] that the mean quantization error from the test observation referring to its nearest best matching units in the trained SOMs is calculated as the anomaly score.

Some researchers combine multiple anomaly detection methods together to construct an ensemble. In [85], empirical mode decomposition and Hilbert-Huang Transform were applied to extract features from bearing vibration signals, and then a hybrid

ensemble detector from the majority voting of Gaussian anomaly detector, nearest neighbor anomaly detector, and PCA anomaly detector was developed to detect anomalies in the feature space.

2.5 Problem Statement and Objectives

A typical scenario in PHM is, a number of sensors and features are prepared from different sources of information and feature selection is carried out at an early stage that fault sensitive sensors and features can be selected and installed for health monitoring. At this stage, experimental study can still be carried out to generate both the healthy and faulty data to guide the feature selection. In the health monitoring at a later stage, large amount of healthy data under a wide range of operating conditions are collected, and faulty data are usually unavailable.

Under this scenario, insensitive and redundant features still remain an issue in feature selection. Selecting features without considering feature correlation structure results in overfitting from redundant features. Available methods either provide inconsistent result when features are correlated or are impractical in terms of computation. Even when appropriate features are used, available anomaly detection methods are unsatisfactory when operating conditions are indeterminate. Because faulty data are usually not available at the stage of anomaly detection and healthy data are abundant, semi-supervised learning is an appropriate approach. Modeling healthy data and

setting up hyper-parameters of the model without faulty data is an open issue to be addressed.

To address above research gaps, this dissertation has the following objectives: first, develop a method to select useful features when insensitive features exist and some features are correlated. This objective involves the identification of the correlation structure of the features, selecting representatives from correlated features, and selecting useful features from representative features. Second, develop a semi-supervised anomaly detection method that works without the information from operating conditions. This objective involves investigation of the influence of operating conditions on machinery data, developing semi-supervised anomaly detection method by identifying the influence of operating conditions, and developing methods to automatically set up model parameters. The feature selection method and the anomaly detection method developed in the dissertation have different requirements on the data: the feature selection method is a supervised method that the training data of all health classes are needed. The anomaly detection method is a semi-supervised method that only the training data from the healthy class are required. Typical application scenarios combining the use of the feature selection and anomaly detection methods are machinery health monitoring projects: at the beginning, sensors and their mounting locations are evaluated. An initial set of sensors and their locations is selected based on domain knowledge. To avoid false negative detection, this initial set of sensors tries to include a large number of sensors covering all possible conditions. Then, experiments are conducted to collect labeled data for all health

classes of interest. After that, feature selection is performed to select uncorrelated useful sensors from the initial sensor set. Based on the correlation structure of the sensors identified by the feature selection method, redundant sensors for critical features are kept. These selected sensors are the ones most sensitive to the change of health conditions and they are installed to the machinery system as the final product. During the actual health monitoring, failure data are rare and they have a wider diversity than the failure data collected from experiments. Therefore, only healthy data are collected by the selected sensors as the training data to train the anomaly detection method of this dissertation, which is semi-supervised.

Chapter 3: Feature Selection

A feature selection method is developed in this dissertation to achieve two objectives: first, select useful features. The method should work even when both noise features and redundant features exist, and it should accept both linear and nonlinear data. Second, identify the correlation structure of the features to find useful redundant features. Identifying the useful redundant features is necessary because in some applications, such as the sensor systems of aircraft or nuclear plant, redundancy of data sources is required for safety concerns.

3.1 Development of the Feature Selection Method

The feature selection method searches the optimal subset of useful features using a procedure consists of three steps. In the first step, features are clustered according to their linear correlations. Correlated features are grouped into the same cluster and they

are regarded as redundant with each other. If a feature is not correlated with other features, it is regarded as a cluster containing only itself. Second, from each cluster a representative feature is selected. Although the features within the same cluster are correlated, they are still different from each other that some features contain more useful information, and some contain more noise. The feature that is most capable of separating different classes is selected as the representative. Third, representative features from all the clusters are concatenated to form a new feature space where all the features are noncorrelated. A subset of features is selected by a feature selection algorithm from this preprocessed feature space. The first two steps identify redundant features and the third step removes noise features. Because redundant features are removed in the first two steps, feature selection algorithm will not suffer from the correlation. The procedure is illustrated in Figure 7.

If feature clustering and representative selection are performed after useful feature selection, the whole procedure will have the same shortcoming as other feature selection methods in processing redundant features. For example, if random forest is used to select useful features as the first step, features with lower capability of separability within a group of highly correlated features have the same chance of being selected [36]. If one less useful feature is selected, the features highly correlated with it will be abandoned and these features may be more capable of separating different classes. After clustering and representative selection, the final feature set loses some of the most useful features. Therefore, feature clustering and representative selection should be performed before useful feature selection.

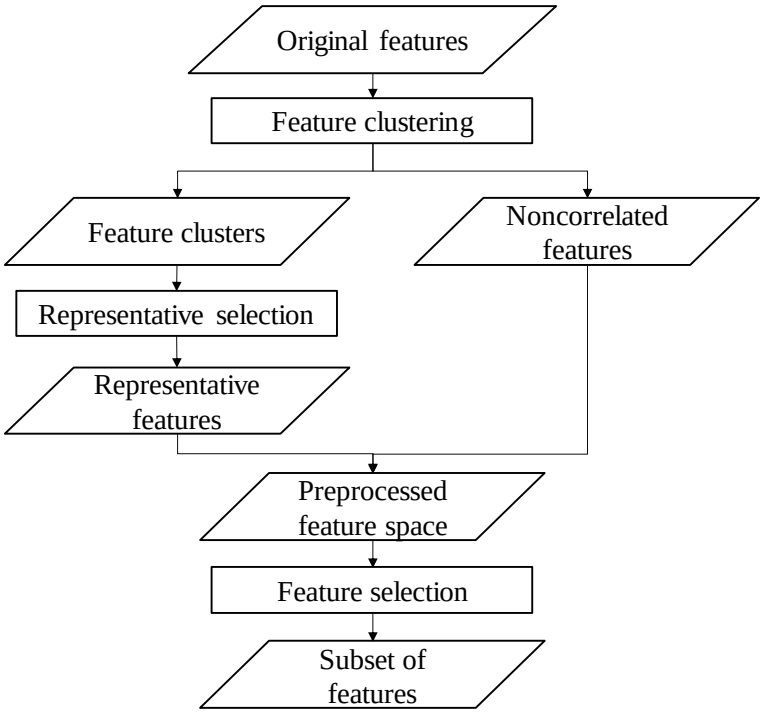

Figure 7: Flowchart of Selection

To implement this procedure, several challenges need to be solved. First, what is the criterion to determine some features are highly correlated and they should be grouped into the same cluster? Second, how to determine if a feature is more capable of separating the classes than the rest within a cluster? Third, what feature selection algorithm should be used in the third step?

To measure if two features are highly correlated, correlation distance is used. Correlation distance $d(u,v)$ between feature u and feature v is defined as in Equation (1).

$$d(u,v) = 1 - \left| \frac{\mathrm{cov}(u,v)}{\sigma_u \sigma_v} \right| \quad (1)$$

where cov is the covariance operator, σ_u and σ_v are the standard deviation of feature u and v, respectively.

The correlation distance is developed based on Pearson's correlation coefficient, where 1 indicates perfect positive linear correlation and -1 indicates perfect negative linear correlation. Since both positive and negative linear correlations have the same influence on feature selection, the strength of linear correlation can be represented by the absolute value of Pearson's correlation coefficient, where 1 indicates perfect linear correlation, and 0 indicates no linear correlation. Correlation distance measures how much two features' relationship deviates from perfect correlation, where 0 indicates no deviation from perfect correlation, and 1 indicates a total deviation. Correlation distance is a dissimilarity measure that can be readily processed by available clustering methods.

Using correlation distance as the dissimilarity metrics, correlated features are grouped into clusters. Compared with *k*-means clustering, density-based clustering [86], and self-organizing maps, agglomerative clustering provides a correlation tree to describe the structure of correlations between the features, which is desirable for intuitive understanding. The agglomerative clustering is used in this research in the following

way: at the beginning of clustering, every feature is a cluster itself. These clusters form a set *F*. Two clusters are merged into one cluster based on the criterion of Equation (2). Then the two original clusters are removed from *F*, and the newly merged cluster is included in *F*. This procedure is performed iteratively until *F* contains only one cluster.

$$\min\{g(a,b) : a,b \in F\} \tag{2}$$

where $g(a, b)$ is the distance between cluster *a* and *b*, as defined in Equation (3).

$$g(a,b) = \min\{d(u,v) : u \in a, v \in b\} \tag{3}$$

The result is a correlation tree that the features in the clusters near the root are less correlated than the features at the furthest branches, as shown in Figure 8, where the cluster tree of the Iris data is plotted. To complete the clustering, a threshold on the correlation distance is needed to cut down the tree. In the Iris example, if the tree is cut below node 1, every feature is a cluster itself. If the tree is cut between node 1 and node 2, feature 2 and 3 form a cluster, and feature 0 and 1 are two separate clusters. If the tree is cut between node 2 and 3, feature 0, 2, and 3 form a single cluster and feature 1 is a cluster itself. If the tree is cut above node 3, all the features form a single cluster.

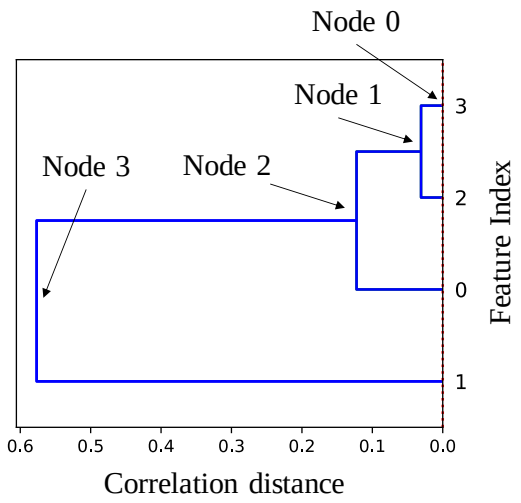

Figure 8: Correlation Tree of the Iris Data

When the correlation tree is cut, one or more clusters are formed. Features in the same clusters are similar and a representative is selected. Selecting only one representative feature from a group of linearly correlated features does not have the same problem of filter methods of ignoring the joint effect of separation using multiple features for the following reasons: when the features are linearly correlated, they carry the same information that contributes to the separation of different classes and therefore including multiple linearly correlated features will not improve the result. However, perfect linear correlation rarely exists and some features may contain a nonlinear relationship that contributes to the separation of classes. Therefore, an optimal value to cut the tree that minimizes loss of joint information among features is needed.

The representative is the feature that is most capable of separating different classes of data within a cluster. Decision tree is used to evaluate the separating capability because of its generosity on the types of data and its advantage of providing intuitive

results. The data from individual features in the same cluster are evaluated by decision tree in turn using cross validation. The feature providing the highest mean accuracy is selected as the representative of the cluster. For clusters with only one feature, the only feature is used as the representative. After representative selection, the correlation among the features that restricts the performance of available feature selection methods is removed.

To select useful features from the representative features, random forest with Gini impurity is used. For each tree in the random forest, the impurity decrease of each feature is calculated. The mean impurity decrease across all the trees in the forest is calculate for every feature, and the features are ranked according to their mean impurity decrease. Compared with feature selection methods based on linear models, random forest has a wider application range that both linear and nonlinear data can be processed. Since random forest is nonparametric, it does not require the data follow any parametric distributions. Being an ensemble learning method, random forest is robust against noise and is more likely to provide stable results.

To determine the optimal value to cut down the correlation tree, the cluster should be defined specifically. Because in this research the purpose of clustering is to find the groups of features that behave similarly in classification, a cluster is defined as a group of features sharing the same piece of useful information that does not exist outside the group. If the threshold value is too high, a smaller number of clusters are formed, and each cluster has more members with higher dissimilarity. Except the information

shared by the members of clusters, some members may specifically contain useful information that contributes to the classification. After representative selection, this useful information is lost, leading to reduced classification accuracy. If the threshold value is too low, a larger number of clusters are formed, and different clusters may share the same information. After representative selection, correlated features still exist, leading to unstable feature selection and reduced classification accuracy. In both scenarios, the definition of clusters of this research is violated. If an optimal threshold is selected, the loss of useful information is minimized because all the features in the same cluster share the same piece of useful information, and after representative selection, this information is kept. The number of redundant features is also minimized because the shared useful information does not exist outside the cluster. As a result, classification accuracy is maximized. The optimal threshold is determined by searching all the tree-cut nodes. In the example of the Iris data, the cluster tree is cut at node 1, node 2, and node 3, generating 3 subsets of features by implementing the procedure described in Figure 7. The tree-cut node value leads to the highest classification accuracy corresponds to the threshold, as shown in Equation (4).

$$d_T = \arg\max_{d \in D} A(d) \tag{4}$$

where d_T is the threshold, d is the correlation distance, A is the model performance metrics, such as accuracy, area under curve, or a transform of generalization error. In this research the mean accuracy of cross validation is used. D is the set of correlation distance values corresponding to the nodes of correlation tree.

In general, if there are *m* features, the correlation tree has *m* nodes, including the root node. The procedure in Figure 7 is evaluated *m* times to get the optimal value. The dimensionality of most systems is in the order of less than 10^3, and an exhaustive searching strategy is practical. In cases of large dataset with higher dimensions, heuristic optimization algorithms such as simulated annealing [87] can be applied for the search. The searching strategy is shown in Figure 9. Moreover, since the searches are independent from each other, parallel computing is implemented, as shown in the dashed box of Figure 9.

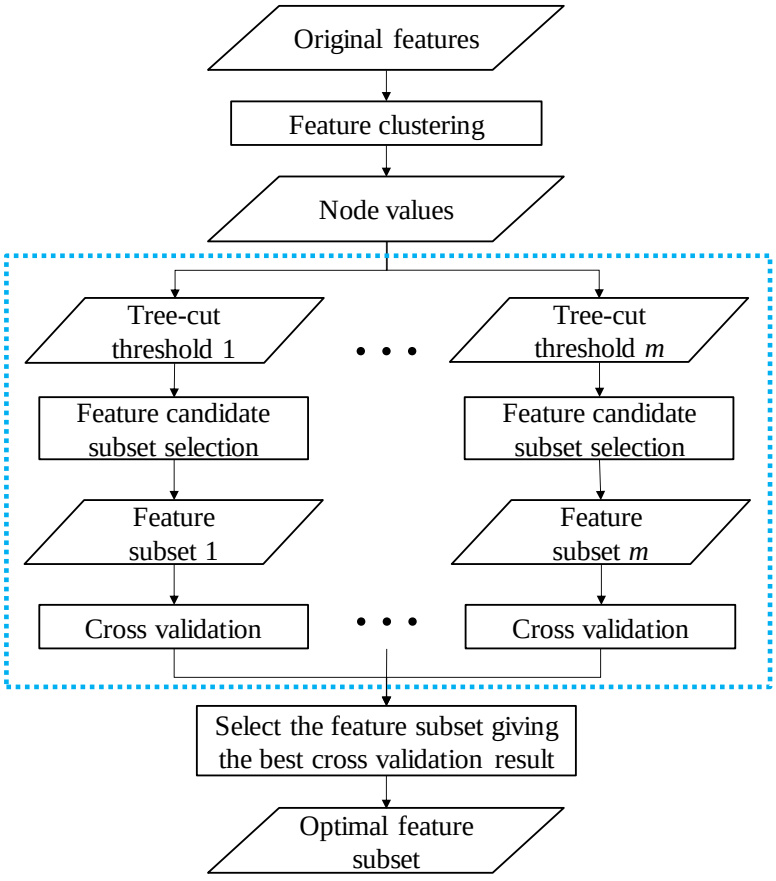

Figure 9: Optimization of the Feature Selection

At first, a correlation tree is obtained by agglomerative feature clustering using correlation distance as dissimilarity measure. From the correlation tree the correlation distance values at the nodes are obtained. These values of the nodes are used as thresholds to cut down the treetop form clusters. For each tree-cut threshold, a feature subset is selected using the procedure of Figure 7. Each feature subset is evaluated by cross-validation to get its performance in classification. The feature subset giving the best cross-validation result is the final selection, and the corresponding threshold value is the optimal value for cluster tree-cut. The classifier used in the cross validation can be determined by the specific problem. In this research, decision tree is used due to its wide application range and intuitive result, which is needed in feature selection. Because the whole procedure is based on the searching of the optimal tree cut, the method is named correlation tree-cut (CTC) feature selection. This procedure also helps to define the concept of correlated features. Measuring by the absolute value of Pearson's correlation coefficient, any pair of features are correlated with a value between 0 to 1. 0 indicates no correlation and 1 indicates perfect correlated. In real life 0 and 1 of the coefficient values are ideal and they are unlikely to occur. Since any value above 0 indicates a degree of correlation, to determine if a value is significant, researchers have selected fixed thresholds, which is a subjective approach. In this research, a flexible threshold is used. Different thresholds lead to different groups of features. The threshold that leads to the selection of the optimal feature subset, which provides the highest classification accuracy, is selected, and the features with correlation coefficients above the threshold are regarded as correlated with each other. The true-cut method is choosing a global threshold to cut down the tree to form

clusters. Using localized thresholds for tree-cut may have possibility of utilizing more detailed information of the correlation tree to generate clusters leading to higher classification accuracy. This will lead to the development of another feature clustering algorithm and can be studied as a future research topic.

3.2 Evaluation of the Feature Selection Method

The correlation tree-cut (CTC) feature selection was compared with widely used feature selection methods including random forest (RF) with Gini impurity, stability selection (SS) using randomized logistic regression with $L1$ penalty, recursive feature elimination (RFE) using decision tree, support vector machine (SVM), and logistic regression (LR) with $L1$ penalty. For each method, the selected subset of features was evaluated by k-fold cross validation. The mean accuracy and the number of selected features were used as performance metrics. Higher mean accuracy and smaller number of selected features indicate better performance of a method. In the k-fold cross validation, decision tree was used to induce classifiers because of its capability of classifying both linearly separable and non-separable data and its insensitivity to the setup of hyperparameters. Moreover, unlike random forest classifier or hybrid classifier [88], decision tree's own feature selection effect is less likely to mask the performance of the feature selection methods under evaluation is set to 5 because 5 values from the testing results are usually regarded as statistically large enough for the calculation of mean and standard deviation (std), and are small enough to give adequate observations for the testing fold. Datasets used in the evaluation consist of simulated data, benchmark data and experimental data.

3.2.1 Feature Selection Method Evaluation Using Simulated Data

Simulated data, experimental data, and field data were applied to evaluate the method to avoid the possibility that any conclusion is made only for specific situations. Simulated data were used because the ground truth of useful features is known. Spiral data with two classes were simulated because they are challenging cases of nonlinear data, and thus the performance of different methods can be distinguished more easily. The data have 12 features and 400 observations. The features include 2 useful features that separate the classes, as shown in Figure 10. Additional features include 6 features of Gaussian noise, 2 features correlated with the 2 useful features, and 2 features correlated with 2 noise features.

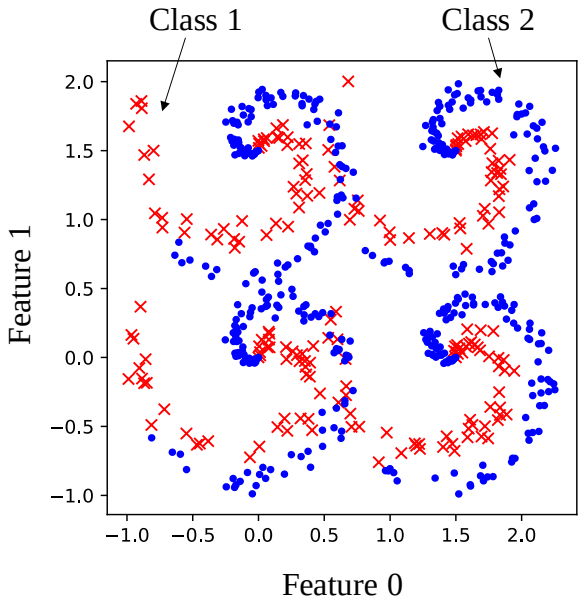

Figure 10: Useful Features of the Simulated Data

The correlation tree from CTC is shown in Figure 11. It correctly identified feature 0 and feature 8, feature 1 and feature 9, feature 2 and feature 10, feature 3 and feature 11 are mutually redundant features due to their linear correlations.

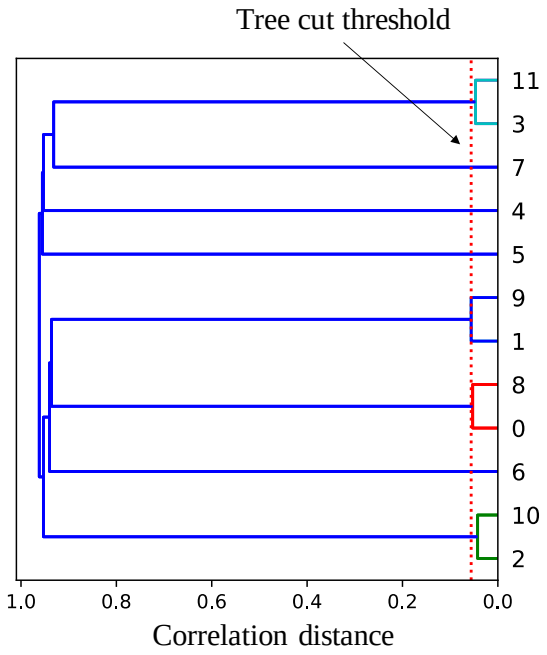

Figure 11: Correlation Tree of the Simulated Data

CTC automatically selected feature 1 and feature 2 as the useful features. The result is consistent with the ground truth. A comparison of the results from all the selected methods is shown in Table 1. Compared with other methods, CTC is the only method correctly identified the useful features. RF, SS, SVM, and LR mistakenly include noise features and some redundant features as useful features. Although RFE selected two features, one feature is a redundant feature of an actual useful feature. As a result, CTC has higher mean accuracy and lower std accuracy from the cross validation.

Table 2: Feature Selection Results of the Simulated Data

Method	CTC	RF	SS	RFE	SVM	LR
# Features	2	3	9	2	5	8
Mean Accuracy	88.3%	87.7%	86.2%	87.5%	86.2%	86.5%
std Accuracy	0.02	0.033	0.048	0.035	0.028	0.025

The performance of the methods was further evaluated by increasing the number of noise features and redundant features, as shown in Figure 12 and Figure 13.In addition to the 2 useful features, 2*m* noise features were added to the feature set. Subsequently, *m* redundant features were added to the feature set, some of which were linearly correlated with the useful features and some of which were linearly correlated with the noise features. To generate a redundant feature, a feature is randomly selected from the feature set, which consists of both useful features and noise features, and the selected feature is linearly transformed with a randomly generated constant term. As the *m* increases, the number of features selected by linear models such as SS, SVM, and LR increases. This is because the data are not linearly separable, and linear models failed to find a subset of features to maximize the classification accuracy. Instead, most features were assigned the same weight or importance. RF and REF using decision tree are capable of processing nonlinear data and therefore they selected a smaller number of features and achieved higher values of mean accuracy than SS, SVM, and LR. However, RF and REF randomly select features from correlated features. As a result, their performances fluctuated. CTC consistently selected the smallest number of features with the highest mean accuracy. An observation is, when the number of noise and redundant features is 90 and 120, besides two useful features CTC selected an additional noise feature and the mean accuracy was increased as a result. This is

because a randomly generated noise feature may form a separable pattern by chance and thus it contributes to the classification.

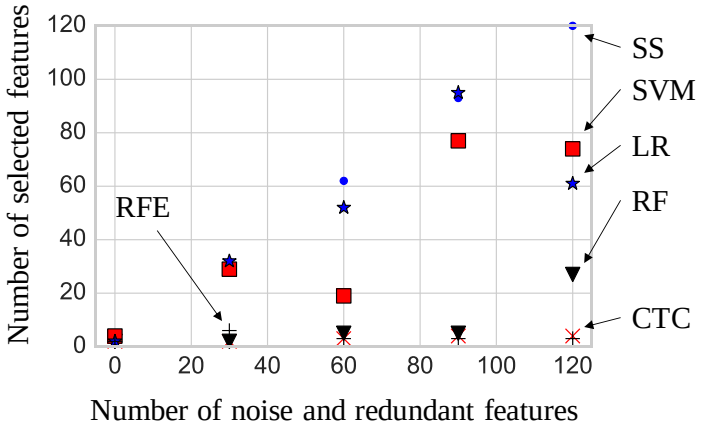

Figure 12: The Number of Selected Features as the Number of Noise and Redundant Features Increases

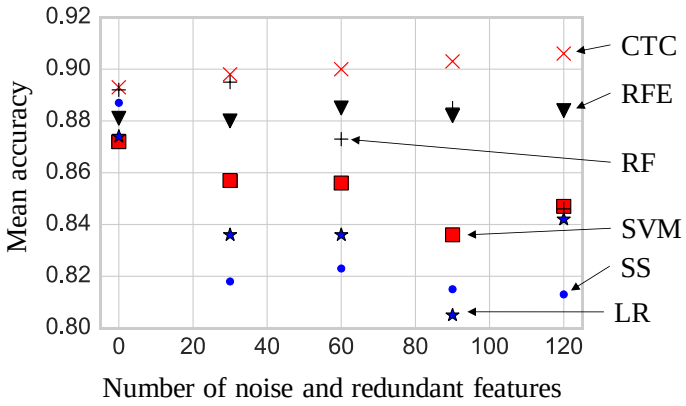

Figure 13: Mean Accuracy as the Number of Noise and Redundant Features Increases

3.2.2 Feature Selection Method Evaluation Using Benchmark Data

Wine dataset [89] was used as the benchmark data. This dataset has 3 classes, 13 features, and 178 observations. The correlation tree from CTC is shown in Figure 14. CTC regards feature 5, feature 6, and feature 11 as mutually redundant features. It identified feature 6, feature 9, and feature 12 as useful features. The results are shown in Table 2. Compared with other methods, the number of features selected by CTC is the smallest, and the mean accuracy is also the highest. However, its std accuracy is not the smallest.

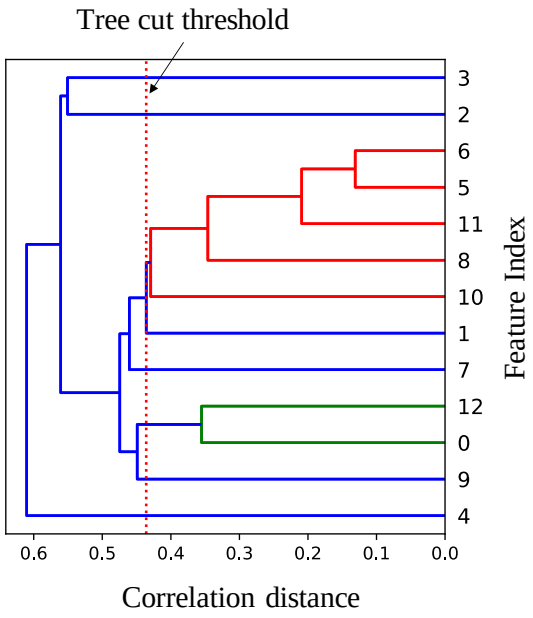

Figure 14: Correlation Tree of the Wine Data

Table 3: Results of the Wine Data

Method	CTC	RF	SS	RFE	SVM	LR
# Features	3	7	8	9	5	7
Mean Accuracy	97.2%	95.5%	95.0%	93.2%	93.2%	94.9%
std Accuracy	0.031	0.014	0.048	0.028	0.056	0.038

Similar to the evaluation of the simulated data, 2*m* noise features and *m* redundant features linearly correlated with the useful and noise features were added to the original 13 features. The results are shown in Figure 15 and Figure 16.

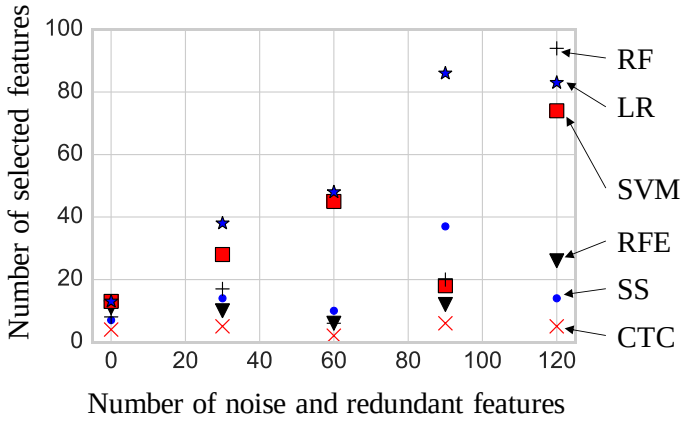

Figure 15: The Number of Selected Features as the Number of Noise and Redundant Features Increases

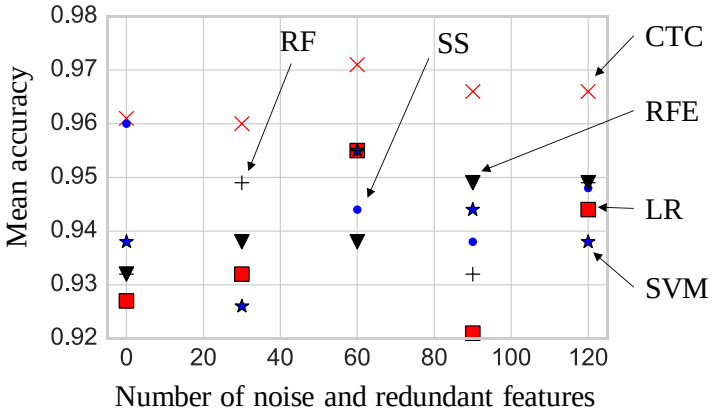

Figure 16: Mean Accuracy as the Number of Noise and Redundant Features Increases

CTC consistently selected the smallest number of features while maintaining the highest mean accuracy: independent of the choice of the number of noise and redundant features, CTC constantly selected the smallest number of features and it has the smallest fluctuation compared with other methods. When the number of noise and redundant features increased, the mean accuracy of CTC did not drop as other methods. In sum, compared with benchmark methods, CTC provided more accurate feature selection results and was influenced the least by noise and redundant features.

3.2.3 Feature Selection Method Evaluation Using Experimental Data

An application of feature selection is to identify useful features from a large number of features in fault diagnosis. Therefore, CTC was evaluated by an experimental study of fault diagnosis using a machinery fault simulator, as shown in Figure 17.

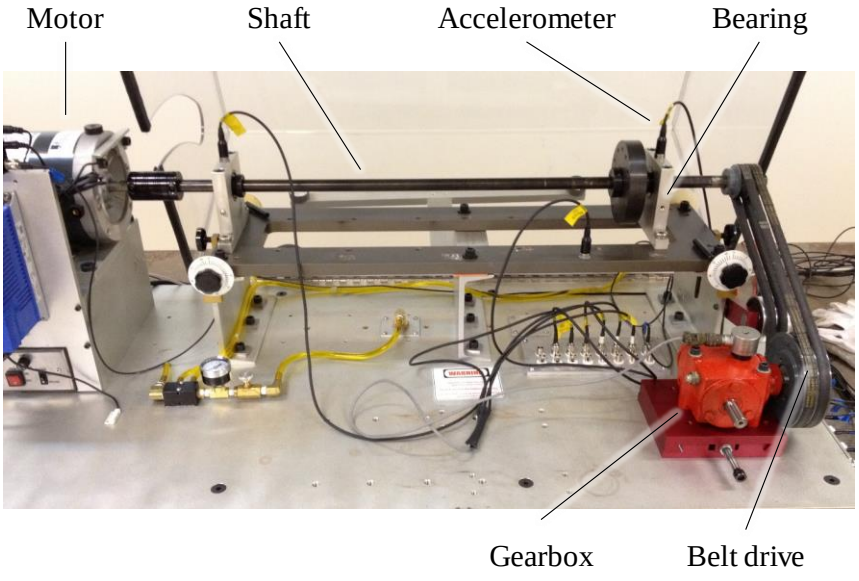

Figure 17: Experiment Setup

The experiment was step for fault diagnosis of rolling element bearings. The bearings under test include a healthy bearing, a bearing with outer race fault, a bearing with inner race fault, and a bearing with ball fault. The data generated were labeled accordingly. A gearbox was installed as a noise source, and the motor driving the bearing was run at different rotation speeds. The setup is to simulate the scenario of conducting health monitoring under changing operating conditions and masking noise.

An accelerometer was mounted on the housing of the bearing to collect vibration acceleration signals at a sampling rate of 25,600 Hz. The sampling rate was selected considering the resonance excited by a faulty bearing should be captured. A rectangular window with a length of 2 s (51,200 points) was applied to slide along the signals with a step of 1s (12,800 points). The window setup gives a frequency resolution of 0.5 Hz, and a time resolution of 1 s, which are appropriate for frequency domain analysis and in-situ monitoring. For each slide of the window, a vector of features was calculated from the portion of the signal inside the window. Widely accepted features in time domain, frequency domain, and time-frequency domain were calculated. This research used 11 time domain features, including peak-to-peak, rms, standard deviation, skewness, kurtosis, impulse factor, crest factor, the square root of the amplitude, margin factor, shape factor, and kurtosis factor. Math expressions of these features are described in [90]. Frequency domain feature used is the primary frequency of the enveloped signal. Time-frequency domain features used are the wavelet packet node energies of the first three levels, which generate 14 features. Together 26 features are used. The data in each dimension were normalized using Z-

score to avoid biased classification from the difference of scales among different features. There are 214 observations distributed in 4 classes of bearing health conditions.

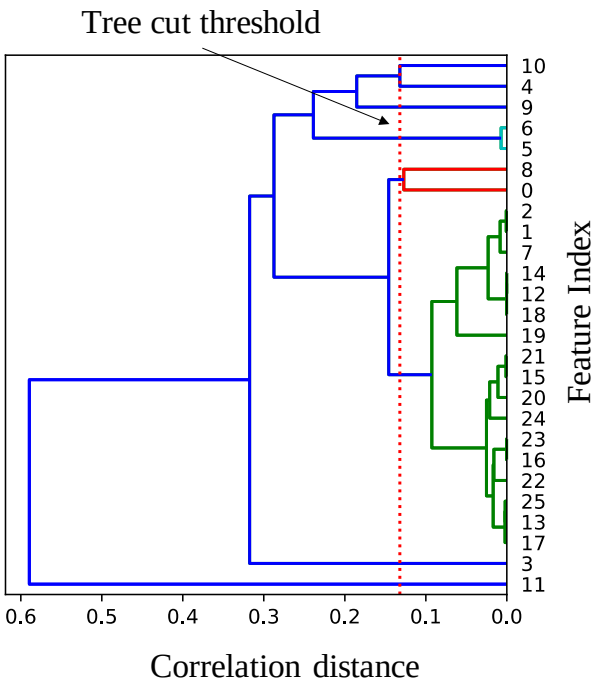

Figure 18: Correlation Tree of the Experimental Data

The correlation tree from CTC is shown in Figure 18. CTC found the features extracted from the wavelet transforms (feature 12-25) are correlated. In addition, they are also correlated with peak-to-peak and rms of the signal in time domain. Among these correlated features, feature 25 has the highest capability of separating different bearing classes. Impulse factor (feature 5) and crest factor (feature 6) are correlated and the impulse factor was selected.

The result is shown in Table 3. Compared with other methods, CTC selected the smallest number of features and achieved the highest mean accuracy and the smallest standard deviation. The two selected features are impulse factor (feature 5) of the signal in the time domain and the energy of wavelet packet (3, 7) (feature 25), which is the detail of the last node of the third level. This selection is consistent with the engineering analysis. When the bearing becomes faulty, rolling elements strike the fault as they run over it, generating impulses. When the signal becomes impulsive, the distribution of the signal has heavier tails that the kurtosis increases. The strike of the rolling elements on the fault excited the resonance of the mechanical structure. Due to the compact structure of the bearing, the resonance usually has a high frequency. As a result, the resonance information is picked up by wavelet packet node energy (WPNE) (3, 7), which represents the highest frequency band of the chosen wavelet packet transform.

Table 4: Results of the Experimental Data

Method	CTC	RF	SS	RFE	SVM	LR
# Features	2	12	25	13	20	12
Mean Accuracy	98.6%	97.2%	94.4%	95.8%	93.9%	96.3%
std Accuracy	0.019	0.028	0.024	0.023	0.044	0.024

The two selected features are plotted in Figure 19. Both features were normalized by calculating their Z score. Because rotation speed was changed several times, the observations of every class form multiple clusters. The clusters from different classes are separable by decision tree classifier in the space spanned by the two selected features and thus a high mean accuracy is achieved.

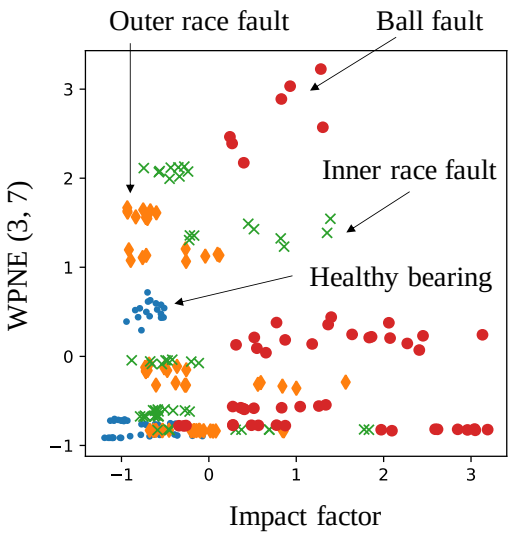

Figure 19: Selected Features

3.2.4 Field Study of the Feature Selection Method

CTC was implemented to evaluate the sensors of a fleet of carrier aircraft from Lockheed Martin. The data were collected from a fleet of more than 20 airplanes. Each airplane has used 57 sensors in health monitoring. A specific type of fault was observed on some airplanes and historic data were labeled as faulty and healthy according to the maintenance findings and engineering judgment. It is assumed some of the 57 sensors are sensitive to the fault of interest and some are not. The task is to identify the fault sensitive sensors, which are the useful features in this research. The original labeled data have several millions of observations. They were down-sampled and a sample of 648 observations was randomly selected for evaluation.

The result is shown in Table 4. CTC selected only one features but it achieved the highest mean accuracy. This result is practical that an interpretable rule can be setup with this feature using decision tree.

Table 5: Results of the Field Data

Method	CTC	RF	SS	RFE	SVM	LR
# Features	1	3	48	25	47	44
Mean Accuracy	87.5%	70.1%	65.4%	66.3%	64.5%	65.1%
std Accuracy	0.025	0.060	0.031	0.051	0.006	0.031

An excerpt of the correlation tree is shown in Figure 20. Several groups of correlated features are identified. However, the selected sensors are not correlated with any other sensors. Therefore, there is a potential to add redundant sensors for the useful sensor.

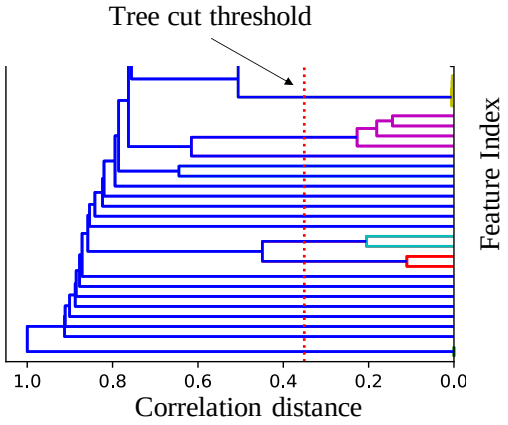

Figure 20: An Excerpt of the Correlation Tree for the Aircraft Data

3.3 Summary

Correlation tree cut (CTC) feature selection method was developed in this dissertation. It can select useful features under indeterminate operating conditions even when some features are correlated. In the simulation study, the CTC method was not affected by the number of noise and correlated features, and its overall performance is superior than that of benchmark methods. In the experimental study, under indeterminate operating conditions, the method provides the highest detection accuracy with the smallest standard deviation. In the field study, it chose a single feature and achieved the highest accuracy. The feature correlation tree constructed in the feature selection method provides an approach to identify redundancy among features. One feature from each cluster is adequate for modeling. Remaining features can be used for needed redundancy.

CTC feature selection is performed without the requirement on domain knowledge of the system. However, the initial set of features to be selected by CTC can benefit from the use of domain knowledge. The initial set of features should include fault sensitive features, and otherwise the result of CTC will be useless. To ensure fault sensitive features are included, domain knowledge can be used to generate a large set of features that have been proven useful theoretically or experimentally.

Chapter 4: Anomaly Detection Using Unimodality-Based Clustering

The focus of this research is semi-supervised anomaly detection. That is, only healthy training data are used and the training data from anomalies are not required. This focus is based on the situation of machinery data: the health states of the data are usually labeled using domain knowledge based on maintenance activities. If no fault is found, the data collected between two adjacent maintenances are labeled as healthy. If a fault is found, the data collected before the maintenance for a certain period are regarded as faulty, and the data collected for a period after the corrective maintenance are labeled as healthy. The periods before and after the maintenance for the determination of the faulty and healthy data are usually judged by engineering experience and thus uncertainties are introduced. To avoid increasing the risk of including faulty data to the set of healthy data, healthy data can be selected only from the period between two adjacent maintenances that no fault is found. Although this practice abandoned the data collected before the maintenances where faults are identified, healthy data are usually abundant to afford the abandoning of the uncertain data. However, the uncertainty involved in labeling faulty data cannot be avoided in the same way. Even when two adjacent maintenances identified the same fault, the data between them are still mixed with both healthy and faulty data. Moreover, a machinery system involves multiple failure modes, and some of them only occur once during the whole life cycle of the system. As a result, it is unlikely to get a data set covering a full range of failure modes for training. In sum, healthy training data are widely available and faulty training data are not, and thus using semi-supervised anomaly detection is practical.

One-class SVM and semi-supervised KNN have been applied as benchmark methods in semi-supervised anomaly detection due to their wide range of application and satisfactory results. However, these methods are sensitive to the choice of hyper-parameters that it is difficult to reach a trade-off between specificity and generality without the supervision of labeled anomalous data, which are usually not available at the stage of anomaly detection.

Clustering-based methods is semi-supervised and they do not have the problem of one-class SVM and KNN. They model the data under a state of operating conditions as a cluster. Each cluster is a group of observations generated by a state of operating conditions when the system is healthy. Thus, a cluster is a generalization of data under a state of operating conditions. The trade-off between specificity and generality is reached by identifying the data associated with different states of operating conditions and by generalizing the data for a given state of operating conditions. After clustering, a test observation is detected as an anomaly if it is not a member of any cluster. A general procedure of clustering-based anomaly detection is illustrated in Figure 21.

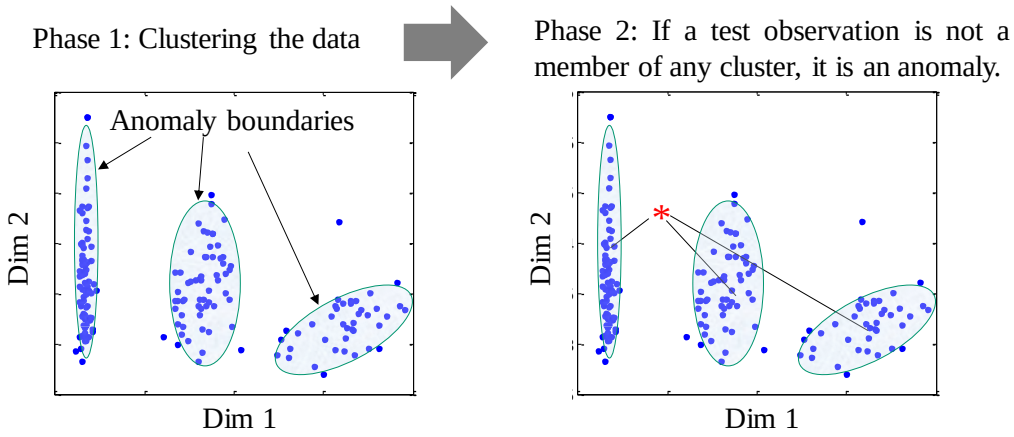

Figure 21: Procedure of Clustering-Based Anomaly Detection

4.1 Development of Unimodality-Based Clustering

Among various clustering approaches, partition-based clustering and hierarchy-based clustering are widely used in the condition monitoring of machinery systems. Representative partition-based clustering algorithms include the *k*-means clustering family, and representative hierarchy-based clustering algorithms include the linkage clustering family. Both approaches require that the number of clusters is known in advance either directly or indirectly. Determining the number of clusters gives rise to the research on clustering validation, where the clustering, given a certain choice of the number of clusters, is evaluated to determine if the clustering is successful under validation criteria[91].

Established methods for determining the number of clusters include the empirical approach, the cluster similarity-based approach, and the distribution-based approach. The empirical approach determines the number of clusters either using domain

knowledge about the data [92] or the elbow method, which tries to make a trade-off between the number of clusters and the variance explained from clustering. User experience is required for every specific clustering, and determining the threshold for the elbow method also requires prior understanding of the data although the threshold often cannot be identified [93]. Thus, the empirical approach is not suitable for automated clustering without user interference. The issue of elbow method is illustrated in Figure 22, where Iris data were used. The Iris data have 3 classes. When the data are clustered without using the labels, 3 clusters should be identified. Elbow method found both 2 clusters and 3 clusters look like the elbow.

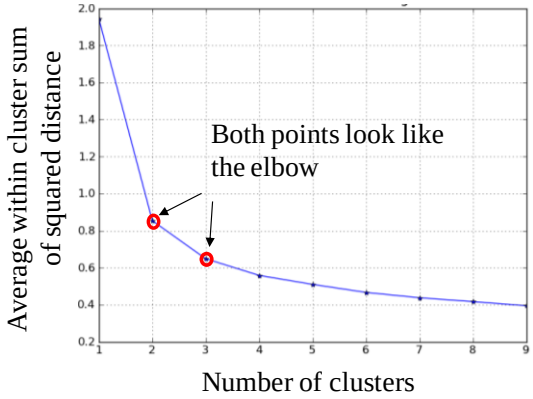

Figure 22: The Issue of Elbow Method

In the similarity-based approach, the number of clusters is determined by maximizing the ratio of intra-cluster similarity to the inter-cluster similarity. Representative methods include the silhouette index (SI) [94][95] and gap index [96]. SI takes value in [-1, 1]. Higher value indicates better result. To use SI, a clustering algorithm such as *k*-means is run by trying different number of clusters. The number of cluster gives the largest SI is selected. For a clustering result, SI is calculated as:

$$SI = \frac{b - a}{\max(a, b)} \tag{5}$$

where a is the mean of intro-cluster distance, and b is the mean of the distances of points to its nearest-clusters.

Similarity-based approach does not use an application-specific definition of clusters and the clustering result is often not valid. For example, SI is liable to choose a larger number of clusters, and gap statistics may not choose the optimum number of clusters if different dimensions have different scales. I

In the distribution-based approach, every cluster is assumed to be sampled from a parametric distribution, such as a Gaussian distribution [97] or a mixture of parametric models [98]. Distribution-based clustering methods give an unambiguous definition of each cluster by describing it with a distribution model. However, the data from a cluster often cannot be described by a known parametric distribution.

Distributions commonly observed in the health monitoring of machinery systems include normal, lognormal, Rice, Rayleigh, Nakagami, Student's *t*, and truncated versions of these distributions [99]. The common characteristic of these distributions is that they are unimodal. Based on this idea and the distribution-based approach, in this dissertation a cluster is defined as a group of observations following a unimodal distribution in every dimension. Thus, if a dataset has M modes, it can be partitioned

into M clusters of unimodal data. A unimodality-based clustering method was developed accordingly in this dissertation: different clustering partitions are generated, and the one validated by the unimodality test is the optimal partition. Compared with normality-based methods, the unimodality-based clustering method has a wider range of applications by considering all unimodal distributions. Based on unimodality-based clustering, an anomaly detection method is developed.

A unimodal distribution is a probability distribution that has a single mode, which means a single value appears most frequently without a local maximum. For the probability density function (pdf) $f(x)$ of a unimodal distribution with mode $x = m$, $f(x)$ is monotonically increasing for $x \leq m$, and monotonically decreasing for $x \geq m$. Intuitively, the pdf of a unimodal distribution has only one peak. However, due to the disturbance of noise, the pdf of the actual unimodal data may not strictly have one peak, which challenges the validation of unimodality.

In this research, a cluster is defined as a group of observations that follow a unimodal distribution in every dimension. This definition addresses the challenges of distribution-based clustering by treating the clusters as following a general form of unimodal distribution, which includes the case where the data cannot be described by a parametric unimodal distribution and the case where the data consists of clusters from different types of unimodal distributions.

If a set of multimodal data is correctly partitioned, the data in each cluster follow a unimodal distribution in every dimension, as illustrated in Figure 23: the data were simulated to have two modes. There would be at least one dimension where the original data are not unimodal, and when the data are partitioned to 2 clusters, the data in a cluster are unimodal, as shown in dimension 1.

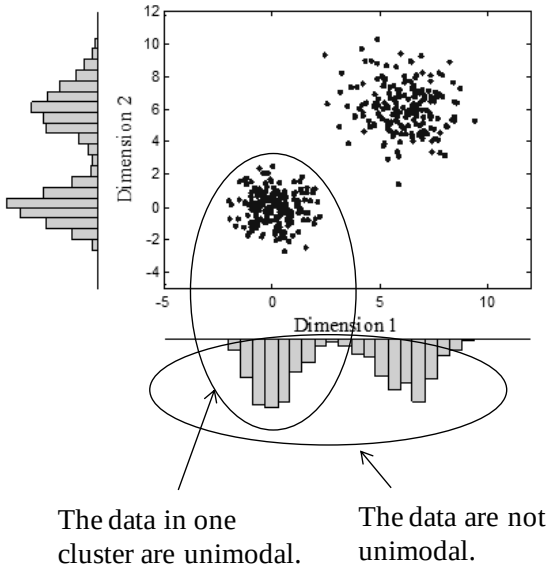

Figure 23: Unimodality and Clusters

To develop a unimodality validation method, the properties of unimodal distributions are investigated. The cumulative distribution function (cdf) has the following properties: first, the cdf is non-decreasing. Second, the cdf converges to 0 and 1. Third, the cdf of a unimodal distribution has only one inflection point. The first two properties hold for any distribution, and the third property is unique for unimodal distributions. The cdf satisfies these properties in the shape of a sigmoid function bounded between 0 and 1. In general, for a data with m modes, the number of inflection points is $2m - 1$. For example, for a bimodal distribution, $m = 2$, and there

are 3 inflection points. For a unimodal distribution, $m = 1$, and there is one inflection point.

If a dataset is unimodal, its empirical distribution converges to its underlying unimodal distribution. In general, if the empirical distribution of data x with n observations is $F_n(x)$, and the underlying distribution is $F(x)$, according to the law of large numbers, $F_n(x)$ converges to $F(x)$, as described in (6) and (7).

$$F_n(x) = P_n(X \le x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \le x) \tag{6}$$

$$\sup_{x \in \Re} | F_n(x) - F(x) | \to 0 \tag{7}$$

where $I(y) = 1$ for y = True, and 0 for y = False. n is the number of observations.

When the data are sampled from a unimodal distribution $G(x)$, the underlying distribution $F(x) = G(x)$:

$$d = \sup_{x \in \Re} | F_n(x) - G(x) | \to 0 \tag{8}$$

The supremum of the absolute difference d between the empirical distribution $F_n(x)$ and the $G(x)$ is the Kolmogorov–Smirnov (KS) statistic. d should converge to 0 if the null hypothesis is true that $G(x)$ is the underlying distribution. Otherwise, d is positive. A smaller d indicates the data are more likely being sampled from $G(x)$. Under the null hypothesis, $\sqrt{n}d$ follows KS distribution, and therefore the KS test can be used to perform the hypothesis test.

Based on the idea of the KS test, Hartigan's dip test [100] was developed to test unimodality. The dip test employs an empirical procedure to find a unimodal distribution that is most similar to the empirical distribution of the data by minimizing the KS statistic *d*. Then a hypothesis test is performed to determine if the data are sampled from the identified unimodal distribution.

The dip statistic was defined as the maximum difference in terms of KS distance between the empirical distribution function and the unimodal distribution function that minimizes that maximum difference. The dip statistic of a distribution function *F* is as defined in (9)

$$r(F) = \rho(F, \Omega) \tag{9}$$

where $r(F)$ is the dip statistic of F. Ω is the class of unimodal distributions. For any bounded functions F and G, $\rho(F, G)$ is defined in (10). For any class Λ of bounded functions, $\rho(F, \Lambda)$ is defined in (11)

$$\rho(F, G) = \sup_x \left| F(x) - G(x) \right| \tag{10}$$

$$\rho(F, \Lambda) = \inf_{G \in \Lambda} \rho(F, G) \tag{11}$$

In the unimodality test, F is replaced by the empirical distribution function of the data under test. Equation (5) is the calculation of the KS distance of the empirical distribution to a unimodal distribution G. The KS distance $\rho(F, G)$ is the largest absolute difference of F and G, as shown in Figure 24.

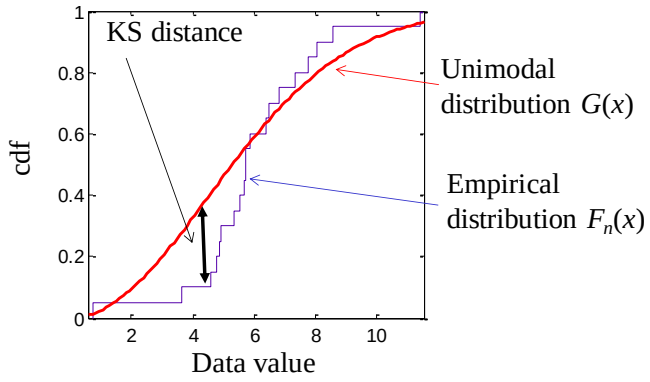

Figure 24: Dip Statistic

In Equation (6), *G* is the unimodal distribution that results in the smallest dip statistic of *F*. To find *G*, the properties of unimodal distributions are investigated. As a cumulative distribution function (cdf), *G* is non-decreasing and it converges to 0, and 1, as shown in Figure 25. *G* has an additional property: it has only one inflection point. Therefore, *G* has the shape of a sigmoid function.

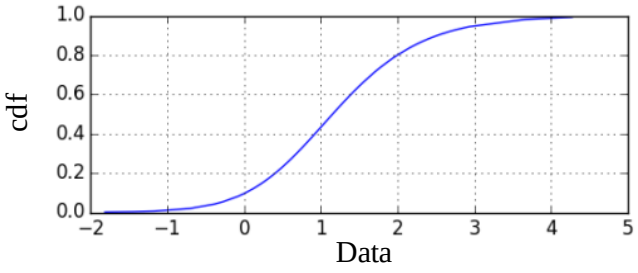

Figure 25: The Shape of Unimodal Distributions

It is not feasible to search all unimodal distributions for *G* in (6), and thus Hartigan and Hartigan [100] developed an estimate of *G* as a curve in the shape of a sigmoid

function. The sigmoid curve minimizes the KS distance between this curve, and the empirical distribution of the data is determined as the estimate of *G*.

A smaller dip statistic of a test dataset indicates the data are closer to a unimodal distribution. The dip statistic changes with the sample size of the data under test. To evaluate the unimodality of datasets with different sizes, the significance of the dip statistic is used, which is invariant with the sample size. It measures the chance that the dataset has a smaller dip statistic than the reference data of the same size from a uniform distribution. A uniform distribution is an extreme case of unimodal distribution; it is between unimodal and multimodal. Hartigan and Hartigan [100] proved that the dip statistic is stochastically larger for the uniform distribution than for any other unimodal distributions. If the test dataset has a dip statistic smaller than that of uniform distribution with a probability *p*, the test data are regarded as unimodal with probability *p*, which is the significance. For example, if the significance $p = 0.5$, it means the test dataset has a 50% chance that its dip statistic is smaller than that of the samples from a uniform distribution. By setting up a threshold of the significance, the unimodality of the clusters can be evaluated.

The unimodality test using the dip statistic significance is integrated with a clustering algorithm to construct a clustering method that automatically estimates the number of clusters. *K*-means clustering, due to its wide application range and the large amount of variant algorithms [101], is integrated as a demonstration. Using other algorithms in the unimodality-based clustering procedure is similar. The choice of clustering

algorithms depends on the application. For example, if the data are nonlinearly correlated or membership scores are needed, kernel fuzzy *c*-means clustering is a better choice, as in [102].

K-means clustering finds cluster centroids and identifies data observations that belong to the clusters of the centroids according to the distances of the observations to the centroids. An observation belongs to the cluster for which the observation has the shortest distance to the centroid. For a given set of centroids, the clustering criterion F is provided by (12).

$$F = \arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i \|^2 \qquad (12)$$

where $S = \{S_1, S_2, \ldots, S_k\}$ are k sets of clustered data; $(x_1, x_2, \ldots, x_n)$ are n observations to be clustered; and μ_i is the centroid of S_i.

To find the centroids of the clusters, Lloyd's algorithm is usually used. Centroids for the initial clusters are assigned randomly. When the distances between each observation and the centroids are calculated, the membership of the observations in the clusters is reassigned so that an observation becomes an member of a cluster with the nearest centroid. After that, the centroids of the updated clusters are calculated. This process is repeated until the membership of the data points in the clusters does not change any more.

In the unimodality-based clustering developed in this dissertation, *k*-means is implemented to partition the data into a small number of clusters, and then the unimodality of the data from each of the *k* clusters is tested on all dimensions. If the data from all the *k* clusters pass the unimodality test, the number of clusters is determined as *k*. Otherwise, the number of clusters is increased to $k + 1$, and the procedure is repeated. To avoid outliers being identified as a unimodal cluster, the size of each cluster is examined. If the cluster size is smaller than a certain number, it is not regarded as a valid cluster, and the method would repeat the procedure by using a new set of initial centroids for *k*-means. The procedure for the method is shown in Figure 26.

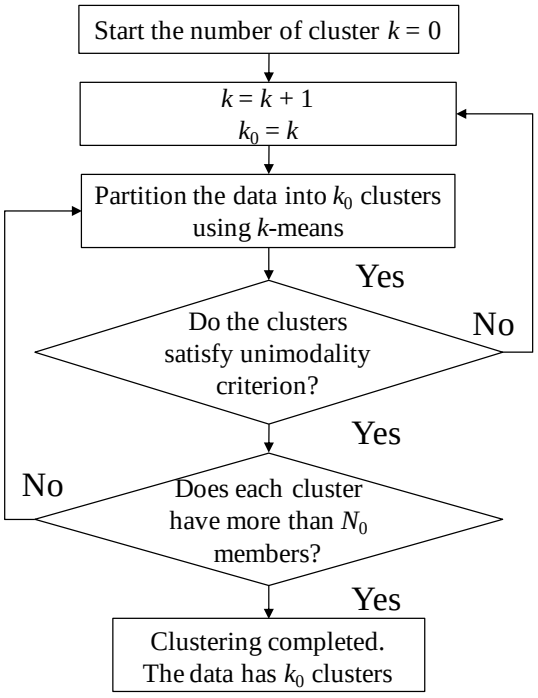

Figure 26: Flowchart of Unimodality-Based Clustering

4.2 Evaluation of Unimodality-Based Clustering

Clusters with different degrees of overlaps, dimensionalities, and distributions were simulated to evaluate the performance of unimodality-based clustering. It was compared with silhouette-based clustering and gap-based clustering due to their widely reported effectiveness [91] [95] [103]. In silhouette-based clustering, different partitions from *k*-means clustering are tried and the partition that maximizes the mean silhouette coefficient is selected [94]. In the same way, the partition that maximizes the gap statistic is selected by gap-based clustering [96].

Adjusted Rand score was applied to measure the clustering result. Adjusted Rand score measures the similarity between true cluster labels and estimated cluster labels. Adjusted Rand score is based on Rand score, which is defined in the following equation:

$$R = \frac{a+b}{a+b+c+d} \tag{13}$$

where R is Rand score; a is the number of times a pair of observations belongs to the same cluster for both the estimated clustering result and the actual clustering result; b is the number of times a pair of observations belongs to different clusters for both the estimated clustering result and the actual clustering result; c is the number of times a pair of observations belongs to different clusters for the estimated clustering result but in the same cluster for the actual clustering result; d is the number of times a pair of observations belongs to the same cluster for the estimated clustering result but in different clusters for the actual clustering result.

Adjusted Rand score (AR) improves Rand score as shown in the following equation to suppress the score of random labeling.

$$AR = \frac{R - E(R)}{\max(R) - E(R)} \tag{14}$$

A value of 0 indicates the estimated labels for the test data are randomly assigned, and a value of 1 indicates the estimate labels have a perfect match with the true labels [104] [105]. Compared with other metrics such as Rand score and V-measure [106], adjusted Rand score is not misled by the random guess of labels. In every evaluation, the simulation was repeated 10 times, and the mean value of the adjusted Rand score was calculated.

In the first evaluation, the clustering performance on the data with different degrees of overlaps was evaluated. The simulated data have 4 equally spaced clusters. Each cluster had 100 2-dimensional observations sampled from a Gaussian distribution with the same standard deviation (σ). The distance between centroids was changed from 0.5σ to 3σ to simulate different degrees of overlaps, as demonstrated in Figure 27. The results are shown in Figure 28. Unimodality-based clustering, silhouette-based clustering and gap-based clustering are denoted as UC, SC, and GC, respectively. All 3 methods use the same procedure that the data are clustered using *k*-means with the number of clusters increasing iteratively. The best partition is selected by UC if all clusters satisfy unimodality criterion and by SC and GC if the optimal SI and GI are obtained, respectively.

All 3 methods had increased scores when the separations among clusters were increasing, and they had a similar score. However, when the separations between clusters were small, UC had a similar score with SC and a higher score than GC.

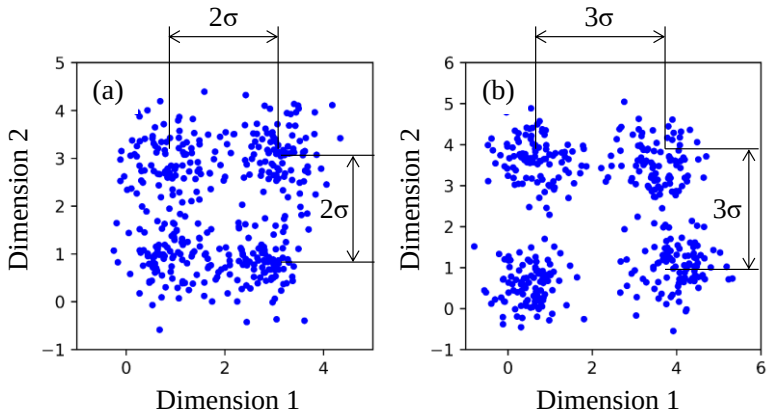

Figure 27: Control Overlaps by Setting the Distance Between Centroids: (a) separate the clusters by 2σ; (b) separate the clusters by 3σ

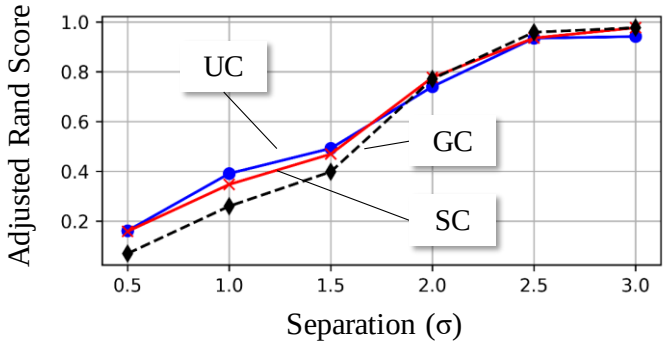

Figure 28: Clustering Performance of Gaussian Clusters Using Unimodality-Based Clustering (UC), Silhouette-Based Clustering (SC), and Gap-Based Clustering (GC)

In the second evaluation, the performance of clustering non-Gaussian data was evaluated. The simulated data have 4 clusters. Each cluster had 100 2-dimensional observations sampled from a lognormal distribution with the same scale parameter σ_L. The distance between the clusters was controlled by the difference of the cluster location parameters, which was set to $5\sigma_L$ in this evaluation. Due to the skewed shape of lognormal distributions, the right tail of the distribution from one cluster can overlap with a nearby cluster even when the distance between the clusters is large enough to separate Gaussian clusters. The value of σ has changed from 0.1 to 0.6 to simulate data with different skewness, as demonstrated in Figure 29. The results are shown in Figure 30. All 3 methods had decreased scores when σ_L was increasing, where the data from every cluster became more skewed. When σ_L was smaller than 0.4, all 3 methods provided the same result. When σ_L was increasing from 0.4, UC and SC had similar scores, and they were decreasing slower than GC.

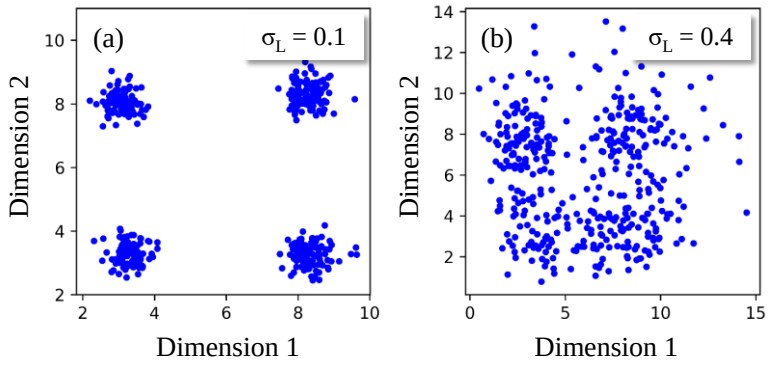

Figure 29: Control the Shape of Clusters: (a) set the scale parameter σ to 0.1; (b) set the scale parameter σ to 0.4

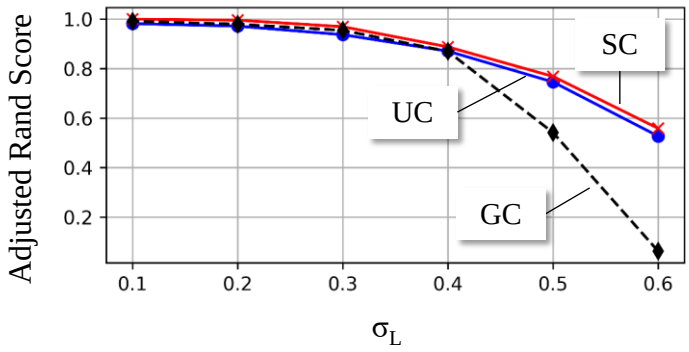

Figure 30: Clustering Performance of Lognormal Clusters

In the third evaluation, the performance of clustering from the influence of dimensionality on Gaussian data was evaluated. Four clusters of Gaussian data were simulated, and each cluster had 100 observations. The distance between cluster centroids was set to 2σ. The dimensionality was increased from 2 to 30. The results are shown in Figure 31. With the increase of dimensionality, the scores of UC and GC increased asymptotically, and the score of SC was decreasing.

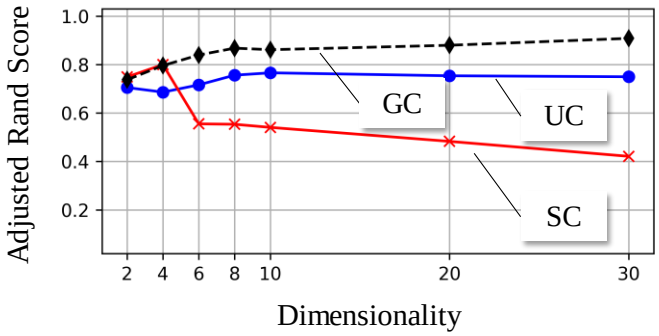

Figure 31: Clustering Gaussian Data of Different Dimensions

In the fourth evaluation, the performance of clustering from the influence of dimensionality on non-Gaussian data was evaluated. Four clusters of lognormal clusters were simulated, and each cluster had 100 observations. The difference of cluster location parameters was set to $5\sigma_L$. The scale parameter was set to 0.5. The dimensionality was increased from 2 to 30. The results are shown in Figure 32. When the dimensionality was increasing, UC provided a consistent score; SC decreased asymptotically; GC did not work when the dimensionality was small, and as the dimensionality increased, GC caught up with the performance of UC and SC.

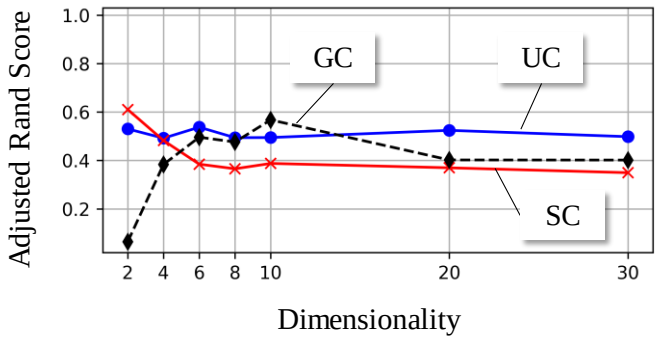

Figure 32: Clustering non-Gaussian Data of Different Dimensions

From the above 4 evaluations, although UC does not provide the highest score in all cases, it has the smallest negative influence from cluster overlapping, non-Gaussian data, and dimensionality compared with SC and GC. Therefore, UC can work as a general-purpose clustering algorithm. Especially, when the properties of the data are unknown, UC is the best choice. For specific applications, UC is the best choice for high dimensional non-Gaussian data. For Gaussian data and low dimensional non-Gaussian data, UC does not have advantage over SC or GC.

4.3 Development of the Anomaly Detection Method Using Unimodality-Based Clustering

An anomaly detection method was developed using unimodality-based clustering, as described in Figure 33. At first, the normal reference data are projected to the feature space where the anomaly detection will be conducted. Then the normal reference data are clustered using unimodality-based clustering. After that, a test data observation is projected to the same feature space. An anomaly indicator is calculated by measuring the distance between the test data to its nearest cluster. If the test data observation is not a member of the nearest cluster, it is identified as an anomaly. The idea can be summarized as a hypothesis test. The null hypothesis is that the test observation is a member of the clusters, and the alternative hypothesis is that the test observation is not a member of the clusters. In the fault detection of machinery systems, the null hypothesis corresponds to the hypothesis that the system is healthy, and the alternative hypothesis corresponds to the hypothesis that the system is faulty.

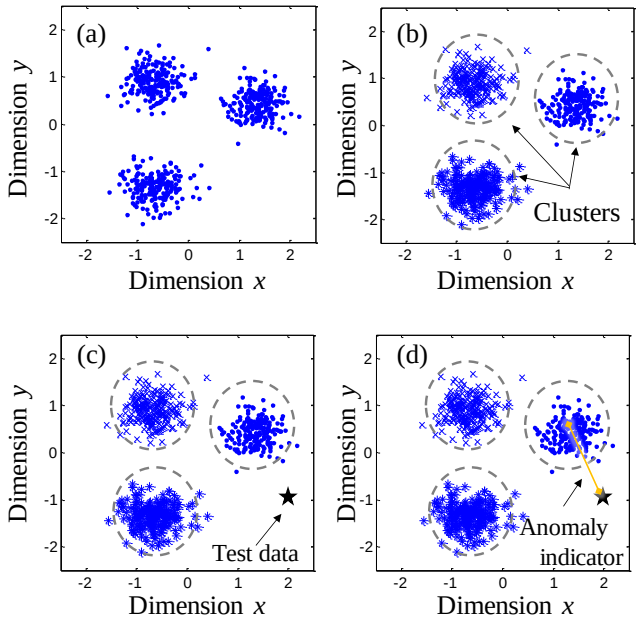

Figure 33: Clustering-Based Anomaly Detection

(a) project normal reference data to the feature space; (b) partition the normal reference data into clusters; (c) project test data to the feature space; and (d) calculate the anomaly indicator of the test data.

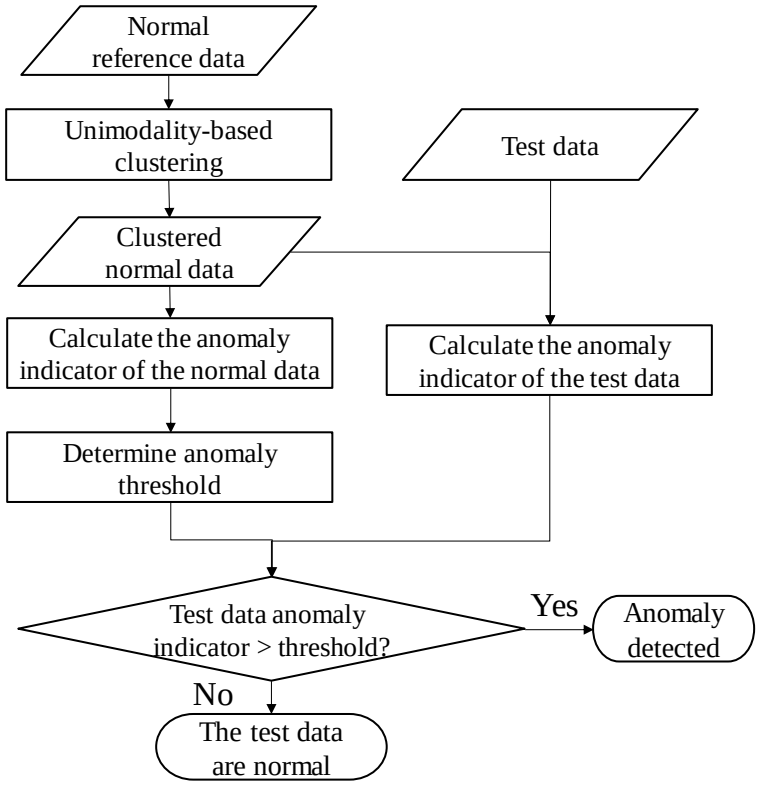

Figure 34: Anomaly Detection Procedure Using Unimodality-Based Clustering.

Figure 34 explains the details of the anomaly detection method. In the procedure, there are 3 major processing steps: unimodality-based clustering, anomaly indicator calculation, and anomaly threshold determination. An anomaly indicator is a measure of how an observation deviates from the normal reference data. Distance measures are widely used anomaly indicators. For example, Mahalanobis distance (MD) has been applied as an anomaly indicator to measure the probability of a test observation being a member of the normal reference data. If the reference data are non-Gaussian unimodal, the MD value loses its original meaning in terms of probability. However, because MD takes the reference data covariance into calculation, as in (15), the shape

of the reference data distribution has less influence on MD than on other distance measures such as Euclidean distance.

$$d_M(x) = \sqrt{(x-\mu)S^{-1}(x-\mu)^T} \qquad (15)$$

where $d_M(x)$ is the MD of an observation $x = (x_1, x_2, \ldots, x_N)$ to the reference data with mean $\mu = (\mu_1, \mu_2, \ldots, \mu_N)$ and covariance matrix S. N is the dimension of the data.

In the dissertation, the anomaly indicator of a test data observation is the MD of this observation to its nearest cluster from the normal reference data, as shown in Figure 35. To test if the test data observation is a member of the nearest cluster, a hypothesis test based on the anomaly indicator is set up. The null hypothesis is, the anomaly indicator value of the test data observation is sampled from the distribution of the anomaly indicator of the nearest cluster from the normal reference data.

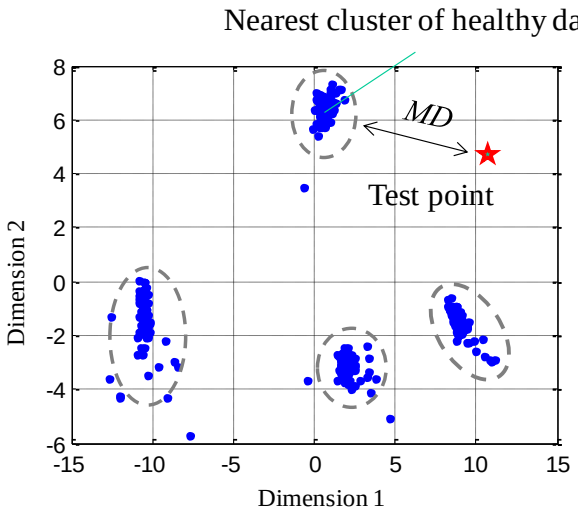

Figure 35: Using MD as Anomaly Indicator

To set up the anomaly threshold for the hypothesis test, the data from the selected nearest cluster are sampled to generate two datasets. One dataset is used as normal reference, and the other is used as test set. Anomaly indicator values for all the test set observations to the reference set are calculated, and then the distribution of these anomaly indicator values is estimated. The hypothesis test can then be conducted by choosing percentile of the data, which corresponds to the false positive rate selected by the user. If the anomaly indicator value of the test data observation is higher than the threshold value identified by the significance level, the null hypothesis is rejected and thus anomaly is detected. The threshold value is the hyperparameter. Its value can be determined by maximizing an anomaly detection performance metric [107] if training data for both normal and anomalies are given. A commonly used performance metric is anomaly detection accuracy, which is the ratio of the number of test observations assigned to the correct labels to the total number of test observations. Because anomaly data are usually unavailable, false positive rate is a practical measure to set up the threshold. The false positive rate equals to 1-(percentile/100). A larger false positive rate means more healthy data would be detected as anomalies by mistake and less true anomalies would be missed. In practice, the user can choose the largest affordable false positive rate as a start to avoid the risk of false negative detection, which is more destructive than false positive detection, and gradually change the threshold to reduce the false positive rate as the information of false negative rate is obtained during the monitoring.

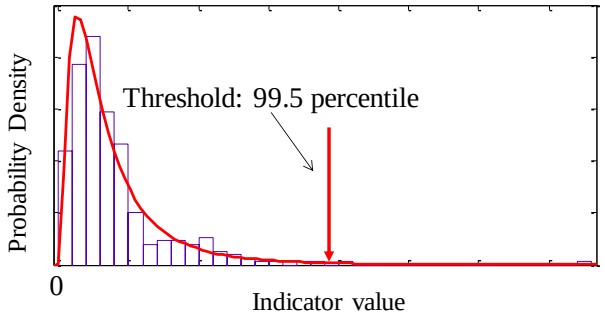

Figure 36: Determination of the Anomaly Threshold

In the health monitoring of machinery systems, when operating conditions are changing, the data have multiple modes. The unimodality-based clustering partitions the data accordingly, and the anomaly detection method is implemented as in Figure 34. When the operating conditions are stationary, in most cases the healthy data have only one mode, and the unimodality-based clustering regards all the healthy data as a single cluster, which is the nearest cluster to the test observation. In some cases, the healthy data have multiple modes. For example, a healthy system may have more than one resonance excited at the same time even when the operation conditions are stationary, and thus the health monitoring data have multiple modes. These data are partitioned into multiple clusters using unimodality-based clustering, and anomaly detection is still performed as in Figure 34. In all the cases, a cluster is a subspace of the healthy reference. If a test observation is not within a subspace, it is an anomaly.

4.4 Evaluation of the Anomaly Detection Method

The anomaly detection method was evaluated by simulated data, benchmark data, and experimental data.

4.3.1 Evaluation of the Anomaly Detection Method Using Simulated Data

Normal reference data of 3 clusters were simulated. Each cluster consisted of 100 2-dimensional observations from a Gaussian distribution. Using unimodality-based clustering, labels were assigned to the observations of the 3 clusters, as shown in Figure 37 (a). After calculating the anomaly indicator values, a lognormal distribution was fit to the anomaly indicator values of the normal data, and a significance level of 0.05 was used to set up the threshold.

Three sets of test data were simulated, and each set had 10 observations, as shown in Figure 37 (b). One of them consisted of normal data. The rest of the datasets were generated by not using the distributions of the normal clusters, and they were used as anomalies. These anomalies were generated to have overlaps with the normal reference data. By comparing the anomaly indicator value of the test data to the threshold, 19 of the 20 actual anomalies were detected, and 1 of the 10 actual normal observations was mistakenly detected as anomalies. The anomaly detection accuracy was 93.3%.

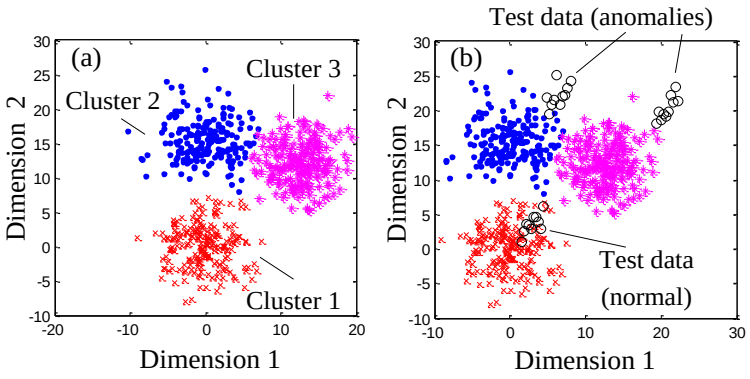

Figure 37: Simulated Normal Reference Data and Test Data

The results were compared with a conventional method without clustering normal reference data, and the anomaly indicator value is the MD value of a test data observation to the whole normal reference data. The anomaly threshold was set up in the same way as in the anomaly detection using unimodality-based clustering.

Using the conventional method, the anomaly detection accuracy is 76.6%, which is lower than the 93.3% accuracy from the method developed in this dissertation. The confusion matrix of the detection results is shown in Table I. The anomaly detection method using the unimodality-based clustering and the conventional method are denoted as ADU and CONV, respectively. The conventional method had fewer false positive detections. This is because the normal test data were surrounded by the normal reference data, the conventional method did not refine the reference, and any observation in the surrounded region is regarded as normal. If anomalies exist in the surrounded region, they will also be regarded as normal. On the contrary, the method developed in this dissertation refines the reference to be clusters that even if anomalies

exist in the surrounded regions, they will be detected because they are not members of the clusters.

Table 6: Results of Simulated Data Anomaly Detection

	Detected Normal ADU/ CONV	Detected Abnormal ADU/CONV
True Normal	9/10	1/0
True Abnormal	1/7	19 /13

4.3.2 Evaluation of the Anomaly Detection Method Using Simulated Data

Three most widely used benchmark datasets were used for evaluation: Iris data, Wine data, and Breast Cancer data. For every benchmark dataset, the data from one class was used as anomalies, and the data from other classes were used as normal data. During training, class labels were removed, and a portion of the normal data were used as reference. Remaining normal data and all the abnormal data were used for testing. Evaluation metrics were obtained using 5-fold cross validation. Unimodality Clustering Anomaly Detection (UCAD) were compared with benchmark methods.

The Iris dataset has 4 features with a sample size of 150 distributed in 3 classes. The data from the third class is designated as anomalies and they were not given in training. The data from the remaining two classes are concatenated as the normal data.

Iris data are non-Gaussian. However, anomalies and normal data do not have significant overlaps. UCAD has the best result.

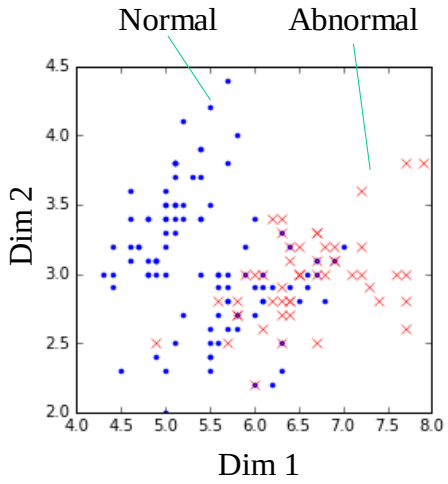

Figure 38: Iris Data

Table 7: Results of Iris Data Anomaly Detection

Iris Data	UCAD	1-SVM	KNN	GMM
Mean accuracy	0.92	0.86	0.80	0.91
STD accuracy	0.07	0.08	0.17	0.07
False negative rate	0.02	0.14	0.08	0.04
False positive rate	0.11	0.14	0.26	0.11

The Wine dataset has more features. It has 13 features with a sample size of 178 distributed in 3 classes. The data from the first class is designated as anomalies and they were not given in training. The data from the remaining two classes are concatenated as the normal data. The normal and abnormal data are overlapped. UCAD has the highest mean accuracy and the smallest standard deviation. However, it has 14% false negative rate.

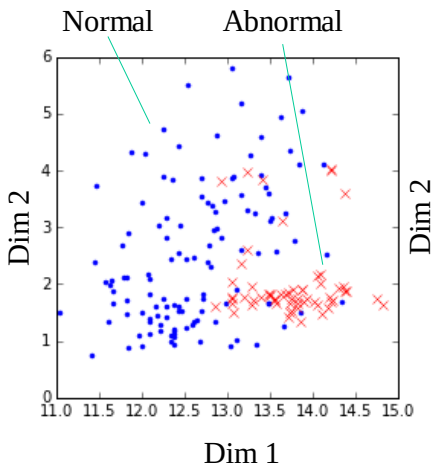

Figure 39: Wine Data

Table 8: Results of Wine Data Anomaly Detection

Wine Data	UCAD	1-SVM	KNN	GMM
Mean accuracy	0.90	0.83	0.81	0.82
STD accuracy	0.06	0.11	0.09	0.09
False negative rate	0.14	0.22	0.10	0.03
False positive rate	0.08	0.15	0.23	0.25

Breast cancer data have more features and more observations than Iris data and Wine data. It has 30 features with a sample size of 569 distributed in 2 classes. The data from the second class is designated as anomalies and they were not given in training. The data from the first class are designated as the normal data. The data are close to be Gaussian, and as a result, besides UCAD, GMM also gets similar results.

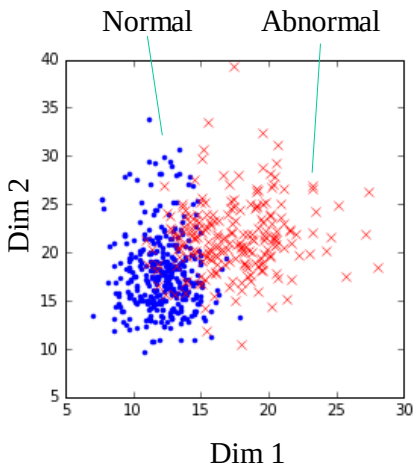

Figure 40: Breast Cancer Data

Table 9: Results of Breast Cancer Anomaly Detection

Breast Cancer Data	UCAD	1-SVM	KNN	GMM
Mean accuracy	0.90	0.90	0.83	0.90
STD accuracy	0.05	0.03	0.05	0.04
False negative rate	0.09	0.10	0.03	0.07
False positive rate	0.10	0.10	0.25	0.12

For all the 3 benchmark datasets, UCAD consistently provided the highest mean accuracy. It also had the smallest STD accuracy for the Iris data and the Wine data. Only in the analysis of the Breast Cancer data, 1-SVM and GMM had smaller STD accuracy than UCAD. However, 1-SVM, KNN, and GMM are sensitive to the choice of hyper-parameters. To set up their parameters, grid-search was applied where some labeled anomaly data were used. UCAD selected an acceptable false positive detection rate, such as 5% in the benchmark data analysis, and avoided the requirement on the labeled anomaly data.

4.3.2 Evaluation of the Anomaly Detection Method Using Experimental Data

Fault detection is a main concern in the maximization of the availability of wind turbines [108]. An experiment was conducted on a wind turbine simulator at the Korea Institute of Machinery and Materials. A wind turbine gearbox was tested because it is the top contributor to wind turbine downtime [109]. Due to the complex environment, the operating conditions of wind turbine gearboxes are usually changing and may not be monitored. The unimodality-based clustering method can partition the data into clusters such that each cluster corresponds to a state of the operating conditions, and thus the anomaly detection method developed in this dissertation can be implemented.

To implement the anomaly detection method to a machinery system, the health monitoring data are usually preprocessed so the information concerning the fault can be extracted. In this experimental study, the necessary preprocessing steps are introduced, which include the setup of signal acquisition, signal preprocessing, the extraction of raw features from the signals, and feature space construction.

The wind turbine simulator is shown in Figure 41. A motor was used to act as the wind power that drives the wind turbine. The gearbox connected to the main shaft of the wind turbine was tested. The gearbox has three stages. The pinion on the third stage, which is connected to the output shaft of the gearbox, was the component under test. Five tests were carried out. In the first test, a healthy pinion was installed to generate

healthy data. In the remaining 4 tests, faulty pinions with a 1 mm root crack, pitted teeth, worn teeth, and a missing tooth were tested, as shown in Figure 42. The healthy pinion was replaced by one of the faulty pinions in turn to generate test data. In each test, rotation speed was changed several times and its value was not given in the analysis, simulating the situation of unmonitored operating conditions. The objective was to check if the method developed in this dissertation can identify the existence of different faults under changing operating conditions.

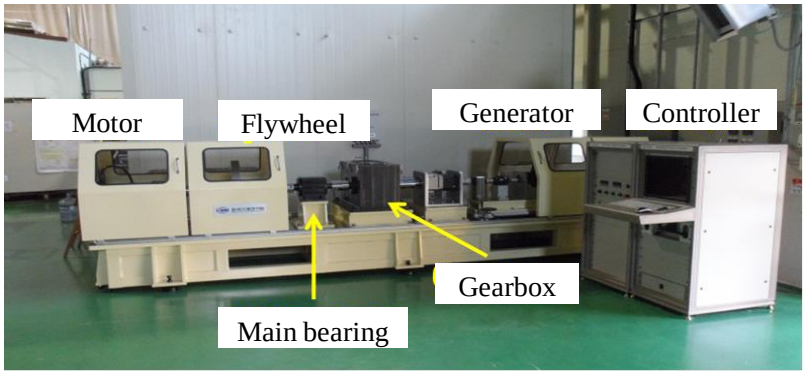

Figure 41: Wind Turbine Simulator

Eight accelerometers were mounted at different locations on the chassis of the gearbox. Vibration signal was used because it is available for most wind turbine condition monitoring systems [110]. In each test, one accelerometer collected 480 s of data under the sampling rate of 25,600 Hz. The sampling rate leads to a Nyquist frequency of 12,800 Hz, which is high enough to capture the fault information carried by the meshing frequency and the resonance frequencies of the gearbox.

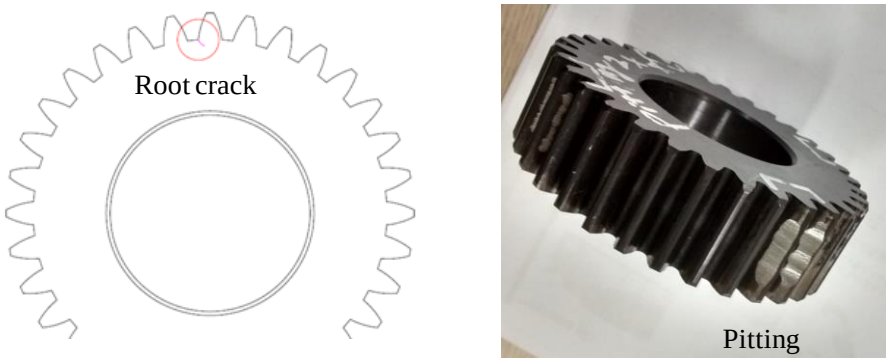

Figure 42: Pinion Faults

Vibration signals from gear meshing systems are modulated. Modulating frequency components have been widely used to indicate the gearbox faults. A major carrier of the gearbox modulated signal is the meshing frequency. Due to the low speed of the gearboxes in wind turbines, meshing frequency usually exists at the same frequency band as the low-frequency noise, and thus the modulated signal is contaminated. Therefore, wavelet thresholding de-noising was performed in this study. Figure 43 shows a comparison of the original data and the de-noised data of the first 12 s of healthy data.

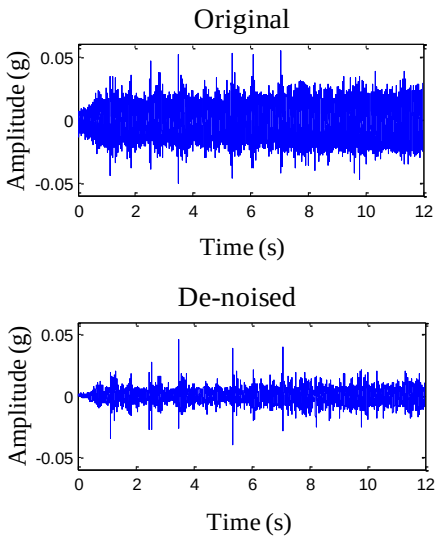

Figure 43: De-Noising Result

After denoising, features were extracted to give an account of the gearbox health state from various aspects. If all 480 s of data were analyzed as one dataset to provide an observation of features, the analysis would face two shortcomings: first, one observation of features is not adequate to draw a conclusion on the health state of the system. Second, the dynamics of the system, such as short-period fluctuations within 480 s, cannot be captured. Thus, the data were cut into 300 segments, and each segment generated one observation of features, so there were 300 observations after feature extraction. Every segment had 1.6 s of data, resulting in a frequency resolution of 0.625 Hz, which is fine enough to differentiate different frequency components in the gearbox signal.

For every signal segment from one of the 8 sensors, 10 widely used features were calculated [111]. Six features were in the time domain, including peak-to-peak, root-

mean-square (rms), standard deviation, crest factor, skewness, and kurtosis of the signal amplitude. Four features were in the frequency domain, including peak magnitude and rms of frequency components, peak magnitude and rms of the frequency components of the enveloped signal. Altogether, the 8 sensors generated 80 features. There are more sophisticated wind turbine gear fault features reported in different sources [112] [113]. These sophisticated features were not used in this study considering they are not available in many existing systems.

After raw feature extraction, the data were grouped into normal reference data and test data. The normal data were uniformly sampled without replacement to form two parts: the first part consisted of 80% of the normal data and was used as the normal reference data; the second part consisted of 20% of the normal data and was used as test data. All the faulty data were used as test data because in actual applications labeled faulty data are usually not available.

The feature space was spanned using the normal reference data by principal component analysis (PCA), which reduces the redundancy and the dimensionality, as introduced in [76]. The values of raw features have different scales so they are normalized by calculating the Z-score before PCA. The cumulative sum of the variance for the first 4 principal components accounts for more than 90% of the total variance, and they were used to span the feature space where the anomaly detection would be performed. The healthy data were projected to the feature space. Dimension 1, dimension 2, and dimension 3 correspond to the first 3 principal components.

The unimodality-based clustering method developed in this study identified the healthy data in the feature space has 4 clusters. This result is consistent with the fact that the data were generated under 4 different rotation speeds. The healthy data were partitioned accordingly, as shown in Figure 44.

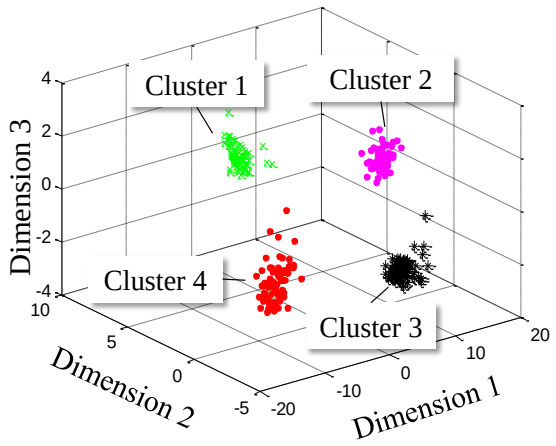

Figure 44: Clustering Result of the Normal Reference Data

Anomaly indicator values were calculated using the method introduced in Section IV. Using the KS test with a significance level of 0.05, the hypothesis that the healthy data anomaly indicator values follow a lognormal distribution was accepted. Using a significance level of 0.05 on the fitted lognormal distribution for the anomaly detection hypothesis testing, the anomaly threshold was set up. Then the test data were projected to the feature space for the calculation of their anomaly indicator values, as demonstrated in Figure 45, where the data from the pinion with 1 mm root crack were plotted.

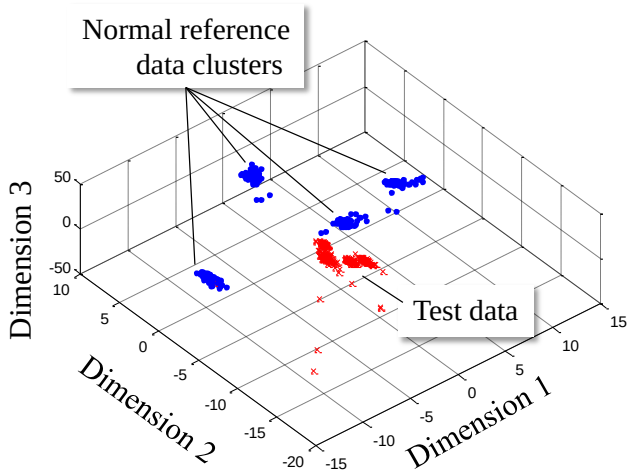

Figure 45: Projecting Test Data to the Feature Space

The anomaly detection method using unimodality-based clustering identified all the data from the faulty pinions as anomalies. Among the 60 test observations of normal data, 2 observations were misidentified as anomalies, resulting in a false positive rate of 3.3%. This is due to the choice of anomaly threshold using the significance level of 0.05, which means the expected false positive rate is 5%. The details of the analysis results are listed in Table 10.

Table 10: Results of Experimental Data Anomaly Detection

Experimental	UCAD	1-SVM	KNN	GMM
Mean accuracy	0.99	0.99	0.95	0.96
STD accuracy	0.02	0.03	0.06	0.04
False negative rate	0.00	0.00	0.01	0.00
False positive rate	0.03	0.03	0.10	0.10

4.4 Summary of Anomaly Detection

Unimodality-based clustering developed in this research provides consistent result in different cases of cluster overlapping, dimensionality, and different types of distributions. In the simulation study, the clustering method was not the best in every case. However, it provided consistent result and it was one of the best methods in all cases. The anomaly detection method addresses the challenge from indeterminate operating conditions. In the analysis of simulated data, UCAD shows superiority over the conventional method without clustering the healthy reference data. In the analysis of benchmark data and experimental data, the anomaly detection method provided one of the best results in all cases. Although 1-SVM can also provide comparable results, it requires anomaly data for hyper-parameter optimization.

As evaluated in the simulation study the unimodality-based clustering used in UCAD has the best performance, relative to other methods, when the healthy training data are high-dimensional and the clusters are non-Gaussian. Because machinery condition monitoring data are usually high-dimensional and non-Gaussian, UCAD is the best choice for machinery anomaly detection. In cases with different dimensions, different overlaps of clusters, and different distributions of clusters, the unimodality-based clustering does not have significant deterioration of performance but other methods do. Therefore, when the knowledge about the data to be processed is incomplete, UCAD is the best choice.

Since the training of UCAD depends on unimodality-based clustering that *k*-means clustering is performed iteratively, the training process is not efficient in computation when the actual number of clusters is large. Therefore, UCAD is not suitable for in-situ training. Although it was not designed to accept data stream for training, the healthy model of UCAD can still be updated periodically when the data labeled as healthy become available. Thus, UCAD matches the application scenario of machinery maintenance that the labeled healthy data are not continuously flowing data stream but data samples collected after maintenance inspections. However, in cases where in-situ training is required, UCAD is not suitable.

Chapter 5: Contributions and Future Work

A feature selection method and an anomaly detection method were developed in this dissertation from different aspect of the task of anomaly detection under indeterminate operating conditions. It has following contributions.

First, this dissertation developed a CTC feature selection method to identify fault sensitive features when some features are correlated, reducing false detection caused by improper selection of features. As shown in the evaluation, when the number of noise and redundant features increases, CTC consistently selects the smallest number of features and achieves the highest mean accuracy. Compared with PCA and manifold learning methods, the features selected by CTC maintain the original physical meaning. Moreover, the correlation tree and tree-cut method of CTC helps to

discover the redundant features, which is required in safety-critical applications. CTC sees a wide range of applications including variable interpretation, model simplification, and sensor system improvement.

Second, UCAD is the first semi-supervised method to perform anomaly detection under indeterminate operating conditions without anomaly training data, without monitoring operating conditions, and without human interference. It expands the application range with increased performance of current anomaly detection practices and contributes to the automation of PHM.

Third, this dissertation developed a unimodality-based clustering method that provides a formal definition of the clusters contained in the machinery health monitoring data that clusters are samples of unimodal distributions. It thereby expands the normality assumption from one that has been conventionally employed to one that is more general. As a result, clustering is feasible for more applications.

Finally, both CTC and UCAD are carried out automatically without human intervention. Since setting up feature selection and anomaly detection methods usually requires human intervention, CTC and UCAD reduces man hours and makes feature selection and anomaly detection easier to implement. Moreover, through this research, the industry can benefit from machine learning in the automation of maintenance practices.

Both methods are designed to work under certain scenarios: CTC feature selection has the best performance compared with available methods when some features are correlated with each other. It does not have significant advantage if the features are not correlated. The application of CTC is also limited by the availability of training data. If training data for some classes are missing, CTC cannot be used. UCAD works best when the healthy data form unimodal clusters. If the clusters are not unimodal, UCAD does not have significant advantage over other clustering-based methods. Although UCAD does not require training data from anomalies, it requires a full coverage of healthy training data.

Future work includes improving the CTC feature selection method for semi-supervised learning, adapting the UCAD for online updating, and the integration of domain knowledge. The CTC feature selection method only work in the supervised mode and therefore both healthy and faulty training data are required. This requirement may not be an issue at any early stage when the experiment can be carried out to generate the data, but for systems not affordable to get faulty training data, a semi-supervised CTC feature selection is needed. The current version of UCAD trains the model for the healthy data only once or periodically updates the model with newly collected healthy data. This property is appropriate for machinery condition-based maintenance but does not satisfy the needs of applications where in-situ updating of models is required. Therefore, a UCAD with in-situ updating capability using data stream should be developed in the future work. Both CTC and UCAD work without domain knowledge

of the machinery. They are useful when the domain knowledge is not available. However, in real applications usually some domain knowledge is available. For example, for a given failure mode, the physics-of-failure model has been established. Physics-of-failure model can be used to identify sensitive features. Such domain knowledge has been used to narrow the search range of the subset of features. That is, a subset of features was selected using domain knowledge, and then a feature selection method such as CTC is applied to perform the final selection. This procedure requires human intervention and the domain knowledge is not integrated with the data-driven method. If the domain knowledge can be integrated automatically with CTC and UCAD, the errors in the analysis are expected to be reduced.

During the research of the dissertation, following papers were published and two more papers were completed. These papers describe specific aspects related to the dissertation and together they depict the roadmap of the research, which is helpful for researchers interested in doing research in machinery anomaly detection.

J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, "Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with *K*-nearest neighbor distance analysis," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1793–1803, 2016.

J. Tian, M. H. Azarian, M. Pecht, G. Niu, and C. Li, "An ensemble learning-based fault diagnosis method for rotating machinery," in *Prognostics and System Health Management Conference (PHM-Harbin), 2017*, 2017, pp. 1–6.

J. Tian, M. H. Azarian, and M. Pecht, “Rolling element bearing fault detection using density-based clustering,” in *Prognostics and Health Management (PHM), 2014 IEEE Conference on*, 2014, pp. 1–7.

J. Tian, M. H. Azarian, and M. Pecht, “Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm,” in *Proceedings of the European Conference of the Prognostics and Health Management Society*, 2014.

J. Tian, C. Morillo, and M. G. Pecht, “Rolling element bearing fault diagnosis using simulated annealing optimized spectral kurtosis,” in *2013 IEEE Conference on Prognostics and Health Management (PHM)*, 2013, pp. 1–5.

Bibliography

[1] S. Sheng *et al.*, “Wind turbine drivetrain condition monitoring during GRC phase 1 and phase 2 testing,” National Renewable Energy Laboratory (NREL), Golden, CO., 2011.

[2] C. Bianchini, F. Immovilli, M. Cocconcelli, R. Rubini, and A. Bellini, “Fault detection of linear bearings in brushless AC linear motors by vibration analysis,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1684–1694, 2011.

[3] W. Qiao and D. Lu, “A survey on wind turbine condition monitoring and fault diagnosis—Part I: Components and subsystems,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6536–6545, 2015.

[4] S. Cheng, M. H. Azarian, and M. G. Pecht, “Sensor systems for prognostics and health management,” *Sensors*, vol. 10, no. 6, pp. 5774–5797, 2010.

[5] M. Pecht, *Prognostics and health management of electronics*. Wiley Online Library, 2008.

[6] T. A. Garcia-Calva, D. Morinigo-Sotelo, and R. de Jesus Romero-Troncoso, “Non-Uniform Time Resampling for Diagnosing Broken Rotor Bars in Inverter-Fed Induction Motors,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 3, pp. 2306–2315, 2017.

[7] F. Dalvand, A. Kalantar, and M. S. Safizadeh, “A Novel Bearing Condition Monitoring Method in Induction Motors Based on Instantaneous Frequency of Motor Voltage,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 1, pp. 364–376, 2016.

[8] S. H. Kia, H. Henao, and G.-A. Capolino, “Gear tooth surface damage fault detection using induction machine stator current space vector analysis,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1866–1878, 2015.

[9] 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.

[10] Z. Gao, C. Cecati, and S. X. 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*, vol. 62, no. 6, pp. 3768–3774, Jun. 2015.

[11] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[12] L. Zhou, F. Duan, E. Faris, and D. Mba, “Seeded Planetary Bearing Fault in a Helicopter Gearbox—A Case Study,” in *International Conference Design and Modeling of Mechanical Systems*, 2017, pp. 495–505.

[13] Z. Huo, Y. Zhang, P. Francq, L. Shu, and J. Huang, “Incipient fault diagnosis of roller bearing using optimized wavelet transform based multi-speed vibration signatures,” *IEEE Access*, vol. 5, pp. 19442–19456, 2017.

[14] J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, “Motor Bearing Fault Detection Using Spectral Kurtosis-Based Feature Extraction Coupled With K-Nearest Neighbor Distance Analysis,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1793–1803, Mar. 2016.

[15] Z. Xia, S. Xia, L. Wan, and S. Cai, “Spectral regression based fault feature extraction for bearing accelerometer sensor signals,” *Sensors*, vol. 12, no. 10, pp. 13694–13719, 2012.

[16] H. Oh, J. H. Jung, B. C. Jeon, and B. D. Youn, “Scalable and Unsupervised Feature Engineering Using Vibration-Imaging and Deep Learning for Rotor System Diagnosis,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3539–3549, 2018.

[17] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[18] K. Torkkola, “Feature extraction by non-parametric mutual information maximization,” *Journal of machine learning research*, vol. 3, no. Mar, pp. 1415–1438, 2003.

[19] R. Battiti, “Using mutual information for selecting features in supervised neural net learning,” *IEEE Transactions on neural networks*, vol. 5, no. 4, pp. 537–550, 1994.

[20] 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.

[21] P. A. Mundra and J. C. Rajapakse, “SVM-RFE with MRMR filter for gene selection,” *IEEE transactions on nanobioscience*, vol. 9, no. 1, pp. 31–37, 2010.

[22] J. Reunanen, “Overfitting in making comparisons between variable selection methods,” *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1371–1382, 2003.

[23] S.-W. Lin, Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng, “Parameter determination of support vector machine and feature selection using simulated annealing approach,” *Applied soft computing*, vol. 8, no. 4, pp. 1505–1512, 2008.

[24] J. Yang and V. Honavar, “Feature subset selection using a genetic algorithm,” *IEEE Intelligent Systems and their Applications*, vol. 13, no. 2, pp. 44–49, 1998.

[25] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, “Feature selection based on rough sets and particle swarm optimization,” *Pattern recognition letters*, vol. 28, no. 4, pp. 459–471, 2007.

[26] R. Archibald and G. Fann, “Feature selection and classification of hyperspectral images with support vector machines,” *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 674–677, 2007.

[27] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine learning*, vol. 46, no. 1, pp. 389–422, 2002.

[28] E. Romero and J. M. Sopena, “Performing feature selection with multilayer perceptrons,” *IEEE Transactions on Neural Networks*, vol. 19, no. 3, pp. 431–441, 2008.

[29] A. Painsky and S. Rosset, “Cross-Validated Variable Selection in Tree-Based Methods Improves Predictive Performance,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2142–2153, 2017.

[30] A. Krishnakumari, A. Elayaperumal, M. Saravanan, and C. Arvindan, “Fault diagnostics of spur gear using decision tree and fuzzy classifier,” *Int J Adv Manuf Technol*, vol. 89, no. 9–12, pp. 3487–3494, Apr. 2017.

[31] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[32] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys, “Robust biomarker identification for cancer diagnosis with ensemble feature selection methods,” *Bioinformatics*, vol. 26, no. 3, pp. 392–398, 2009.

[33] N. Meinshausen and P. Bühlmann, “Stability selection,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 417–473, 2010.

[34] M. Belgiu and L. Drăguţ, “Random forest in remote sensing: A review of applications and future directions,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 114, pp. 24–31, 2016.

[35] A. K. Das, S. Goswami, A. Chakrabarti, and B. Chakraborty, “A new hybrid feature selection approach using feature association map for supervised and unsupervised classification,” *Expert Systems with Applications*, vol. 88, pp. 81–94, 2017.

[36] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, “Conditional variable importance for random forests,” *BMC bioinformatics*, vol. 9, no. 1, p. 307, 2008.

[37] L. Toloşi and T. Lengauer, “Classification with correlated features: unreliability of feature ranking and solutions,” *Bioinformatics*, vol. 27, no. 14, pp. 1986–1994, 2011.

[38] M. Y. Park, T. Hastie, and R. Tibshirani, “Averaged gene expressions for regression,” *Biostatistics*, vol. 8, no. 2, pp. 212–227, 2006.

[39] G. Gan and M. K.-P. Ng, “Subspace clustering with automatic feature grouping,” *Pattern Recognition*, vol. 48, no. 11, pp. 3703–3713, 2015.

[40] R. B. Randall and J. Antoni, “Rolling element bearing diagnostics—A tutorial,” *Mechanical systems and signal processing*, vol. 25, no. 2, pp. 485–520, 2011.

[41] J. Wang, Y. Peng, and W. Qiao, “Current-aided order tracking of vibration signals for bearing fault diagnosis of direct-drive wind turbines,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6336–6346, 2016.

[42] B. L. Song and J. Lee, “Framework of designing an adaptive and multi-regime prognostics and health management for wind turbine reliability and efficiency improvement,” *Framework*, vol. 4, no. 2, 2013.

[43] N. Sammaknejad, B. Huang, and Y. Lu, “Robust Diagnosis of Operating Mode Based on Time-Varying Hidden Markov Models,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 1142–1152, Feb. 2016.

[44] N. Görnitz, M. M. Kloft, K. Rieck, and U. Brefeld, “Toward supervised anomaly detection,” *Journal of Artificial Intelligence Research*, vol. 46, no. 1, pp. 235–262, 2013.

[45] A. Soualhi, G. Clerc, and H. Razik, “Detection and diagnosis of faults in induction motor using an improved artificial ant clustering technique,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4053–4062, 2013.

[46] G. Lu, Y. Zhou, C. Lu, and X. Li, “A novel framework of change-point detection for machine monitoring,” *Mechanical Systems and Signal Processing*, vol. 83, pp. 533–548, 2017.

[47] J. Tian, M. H. Azarian, and M. Pecht, “Rolling element bearing fault detection using density-based clustering,” in *Prognostics and Health Management (PHM), 2014 IEEE Conference on*, 2014, pp. 1–7.

[48] R. Laxhammar and G. Falkman, “Online learning and sequential anomaly detection in trajectories,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1158–1173, 2014.

[49] J. Tian, M. H. Azarian, and M. Pecht, “Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm,” in *Proceedings of the European Conference of the Prognostics and Health Management Society*, 2014.

[50] G. You, S. Park, and D. Oh, “Diagnosis of Electric Vehicle Batteries Using Recurrent Neural Networks,” *IEEE Transactions on Industrial Electronics*, 2017.

[51] W. Du, J. Tao, Y. Li, and C. Liu, “Wavelet leaders multifractal features based fault diagnosis of rotating mechanism,” *Mechanical Systems and Signal Processing*, vol. 43, no. 1–2, pp. 57–75, 2014.

[52] Y. Li, M. Xu, R. Wang, and W. Huang, “A fault diagnosis scheme for rolling bearing based on local mean decomposition and improved multiscale fuzzy entropy,” *Journal of Sound and Vibration*, vol. 360, pp. 277–299, 2016.

[53] Y. Li, M. Xu, H. Zhao, and W. Huang, “Hierarchical fuzzy entropy and improved support vector machine based binary tree approach for rolling bearing fault diagnosis,” *Mechanism and Machine Theory*, vol. 98, pp. 114–132, 2016.

[54] M. S. Safizadeh and S. K. Latifi, “Using multi-sensor data fusion for vibration fault diagnosis of rolling element bearings by accelerometer and load cell,” *Information Fusion*, vol. 18, pp. 1–8, 2014.

[55] J. B. Ali, N. Fnaiech, L. Saidi, B. Chebel-Morello, and F. Fnaiech, “Application of empirical mode decomposition and artificial neural network for

automatic bearing fault diagnosis based on vibration signals," *Applied Acoustics*, vol. 89, pp. 16–27, 2015.

[56] M. Gan, C. Wang, and C. Zhu, "Construction of hierarchical diagnosis network based on deep learning and its application in the fault pattern recognition of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 72–73, pp. 92–104, 2016.

[57] X. Jin, M. Zhao, T. W. S. Chow, and M. Pecht, "Motor Bearing Fault Diagnosis Using Trace Ratio Linear Discriminant Analysis," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2441–2451, May 2014.

[58] N. Zhang, L. Wu, J. Yang, and Y. Guan, "Naive Bayes Bearing Fault Diagnosis Based on Enhanced Independence of Data," *Sensors*, vol. 18, no. 2, p. 463, 2018.

[59] H. Zhou, J. Chen, G. Dong, and R. Wang, "Detection and diagnosis of bearing faults using shift-invariant dictionary learning and hidden Markov model," *Mechanical systems and signal processing*, vol. 72, pp. 65–79, 2016.

[60] J. Phillips, E. Cripps, J. W. Lau, and M. R. Hodkiewicz, "Classifying machinery condition using oil samples and binary logistic regression," *Mechanical Systems and Signal Processing*, vol. 60–61, pp. 316–325, Aug. 2015.

[61] D. H. Pandya, S. H. Upadhyay, and S. P. Harsha, "Fault diagnosis of rolling element bearing by using multinomial logistic regression and wavelet packet transform," *Soft Computing*, vol. 18, no. 2, pp. 255–266, 2014.

[62] N. E. I. Karabadji, H. Seridi, I. Khelf, N. Azizi, and R. Boulkroune, "Improved decision tree construction based on attribute selection and data sampling

for fault diagnosis in rotating machines," *Engineering Applications of Artificial Intelligence*, vol. 35, pp. 71–83, Oct. 2014.

[63] Z. Wang, Q. Zhang, J. Xiong, M. Xiao, G. Sun, and J. He, "Fault Diagnosis of a Rolling Bearing Using Wavelet Packet Denoising and Random Forests," *IEEE Sensors Journal*, vol. 17, no. 17, pp. 5581–5588, Sep. 2017.

[64] M. Cerrada, G. Zurita, D. Cabrera, R.-V. Sánchez, M. Artés, and C. Li, "Fault diagnosis in spur gears based on genetic algorithm and random forest," *Mechanical Systems and Signal Processing*, vol. 70–71, pp. 87–103, 2016.

[65] Y. Lei, M. J. Zuo, Z. He, and Y. Zi, "A multidimensional hybrid intelligent method for gear fault diagnosis," *Expert Systems with Applications*, vol. 37, no. 2, pp. 1419–1430, Mar. 2010.

[66] J. Tian, M. H. Azarian, M. Pecht, G. Niu, and C. Li, "An ensemble learning-based fault diagnosis method for rotating machinery," in *Prognostics and System Health Management Conference (PHM-Harbin), 2017*, 2017, pp. 1–6.

[67] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018.

[68] S. Fu, K. Liu, Y. Xu, and Y. Liu, "Rolling Bearing Diagnosing Method Based on Time Domain Analysis and Adaptive Fuzzy -Means Clustering," *Shock and Vibration*, 2016. [Online]. Available: https://www.hindawi.com/journals/sv/2016/9412787/abs/. [Accessed: 13-Mar-2018].

[69] A. Rodríguez Ramos, O. Llanes-Santiago, J. M. Bernal de Lázaro, C. Cruz Corona, A. J. Silva Neto, and J. L. Verdegay Galdeano, "A novel fault diagnosis

scheme applying fuzzy clustering algorithms," *Applied Soft Computing*, vol. 58, pp. 605–619, Sep. 2017.

[70] A. Soualhi, G. Clerc, and H. Razik, "Detection and Diagnosis of Faults in Induction Motor Using an Improved Artificial Ant Clustering Technique," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 9, pp. 4053–4062, Sep. 2013.

[71] Z. Wei, Y. Wang, S. He, and J. Bao, "A novel intelligent method for bearing fault diagnosis based on affinity propagation clustering and adaptive feature selection," *Knowledge-Based Systems*, vol. 116, pp. 1–12, Jan. 2017.

[72] J. Li, R. Zhao, and L. Deng, "Application of EWT AR model and FCM clustering in rolling bearing fault diagnosis," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 2017, pp. 1–6.

[73] J. Tian, M. H. Azarian, and M. Pecht, "Rolling element bearing fault detection using density-based clustering," in *2014 International Conference on Prognostics and Health Management*, 2014, pp. 1–7.

[74] C. T. Yiakopoulos, K. C. Gryllias, and I. A. Antoniadis, "Rolling element bearing fault detection in industrial environments based on a K-means clustering approach," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2888–2911, Mar. 2011.

[75] M. Inacio, A. Lemos, and W. Caminhas, "Fault diagnosis with evolving fuzzy classifier based on clustering algorithm and drift detection," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[76] J. Tian, C. Morillo, M. H. Azarian, and M. Pecht, "Motor bearing fault detection using spectral kurtosis-based feature extraction coupled with K-nearest

neighbor distance analysis," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 3, pp. 1793–1803, 2016.

[77] D. Fernández-Francos, D. Martínez-Rego, O. Fontenla-Romero, and A. Alonso-Betanzos, "Automatic bearing fault diagnosis based on one-class ν-SVM," *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 357–365, Jan. 2013.

[78] S. Yin, X. Zhu, and C. Jing, "Fault detection based on a robust one class support vector machine," *Neurocomputing*, vol. 145, pp. 263–268, Dec. 2014.

[79] D. Martínez-Rego, O. Fontenla-Romero, A. Alonso-Betanzos, and J. C. Principe, "Fault detection via recurrence time statistics and one-class classification," *Pattern Recognition Letters*, vol. 84, pp. 8–14, Dec. 2016.

[80] J. Yu, "Bearing performance degradation assessment using locality preserving projections and Gaussian mixture models," *Mechanical Systems and Signal Processing*, vol. 25, no. 7, pp. 2573–2588, 2011.

[81] E. B. Martin and A. J. Morris, "Non-parametric confidence bounds for process performance monitoring charts," *Journal of Process Control*, vol. 6, no. 6, pp. 349–358, 1996.

[82] Y. Pan, J. Chen, and X. Li, "Bearing performance degradation assessment based on lifting wavelet packet decomposition and fuzzy c-means," *Mechanical Systems and Signal Processing*, vol. 24, no. 2, pp. 559–566, 2010.

[83] R. Huang, L. Xi, X. Li, C. R. Liu, H. Qiu, and J. Lee, "Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods," *Mechanical Systems and Signal Processing*, vol. 21, no. 1, pp. 193–207, 2007.

[84] J. Tian, M. H. Azarian, and M. Pecht, “Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm,” in *Proceedings of the European Conference of the Prognostics and Health Management Society*, 2014.

[85] G. Georgoulas, T. Loutas, C. D. Stylios, and V. Kostopoulos, “Bearing fault detection based on hybrid ensemble detector and empirical mode decomposition,” *Mechanical Systems and Signal Processing*, vol. 41, no. 1, pp. 510–525, 2013.

[86] J. Tian, M. H. Azarian, and M. Pecht, “Rolling element bearing fault detection using density-based clustering,” in *Prognostics and Health Management (PHM), 2014 IEEE Conference on*, 2014, pp. 1–7.

[87] J. Tian, C. Morillo, and M. G. Pecht, “Rolling element bearing fault diagnosis using simulated annealing optimized spectral kurtosis,” in *2013 IEEE Conference on Prognostics and Health Management (PHM)*, 2013, pp. 1–5.

[88] J. Tian, M. H. Azarian, M. Pecht, G. Niu, and C. Li, “An ensemble learning-based fault diagnosis method for rotating machinery,” in *Prognostics and System Health Management Conference (PHM-Harbin), 2017*, 2017, pp. 1–6.

[89] M. Forina, “An extendible package for data exploration, classification and correlation,” *Institute of Pharmaceutical and Food Analisys and Technologies, Via Brigata Salerno*, vol. 16147, 1991.

[90] Z. Xia, S. Xia, L. Wan, and S. Cai, “Spectral regression based fault feature extraction for bearing accelerometer sensor signals,” *Sensors*, vol. 12, no. 10, pp. 13694–13719, 2012.

[91] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, “An extensive comparative study of cluster validity indices,” *Pattern Recognition*, vol. 46, no. 1, pp. 243–256, 2013.

[92] J. Yu, “Bearing performance degradation assessment using locality preserving projections and Gaussian mixture models,” *Mechanical Systems and Signal Processing*, vol. 25, no. 7, pp. 2573–2588, 2011.

[93] D. J. Ketchen Jr and C. L. Shook, “The application of cluster analysis in strategic management research: an analysis and critique,” *Strategic management journal*, pp. 441–458, 1996.

[94] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[95] A. Gupta and P. S. Merchant, “Automated Lane Detection by K-means Clustering: A Machine Learning Approach,” *Electronic Imaging*, vol. 2016, no. 14, pp. 1–6, 2016.

[96] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[97] J. Jacques and C. Preda, “Model-based clustering for multivariate functional data,” *Computational Statistics & Data Analysis*, vol. 71, pp. 92–106, 2014.

[98] W. Labeeuw and G. Deconinck, “Residential electrical load model based on mixture model clustering and Markov models,” *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1561–1569, 2013.

[99] E. Bechhoefer and A. P. Bernhard, “A generalized process for optimal threshold setting in HUMS,” in *Aerospace Conference, 2007 IEEE*, 2007, pp. 1–9.

[100] J. A. Hartigan and P. M. Hartigan, “The dip test of unimodality,” *The Annals of Statistics*, pp. 70–84, 1985.

[101] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[102] K. Peng, K. Zhang, B. You, J. Dong, and Z. Wang, “A quality-based nonlinear fault diagnosis framework focusing on industrial multimode batch processes,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2615–2624, 2016.

[103] E. Hancer and D. Karaboga, “A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number,” *Swarm and Evolutionary Computation*, vol. 32, pp. 49–67, 2017.

[104] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.

[105] H.-S. Park and C.-H. Jun, “A simple and fast algorithm for K-medoids clustering,” *Expert systems with applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[106] A. Rosenberg and J. Hirschberg, “V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure.,” in *EMNLP-CoNLL*, 2007, vol. 7, pp. 410–420.

[107] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 281–305, 2012.

[108] S. Simani, S. Farsoni, and P. Castaldi, “Fault diagnosis of a wind turbine benchmark via identified fuzzy models,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3775–3782, 2015.

[109] W. Qiao and D. Lu, “A survey on wind turbine condition monitoring and fault diagnosis—Part I: Components and subsystems,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6536–6545, 2015.

[110] W. Qiao and D. Lu, “A survey on wind turbine condition monitoring and fault diagnosis—Part II: Signals and signal processing methods,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 10, pp. 6546–6557, 2015.

[111] Y. Lei, J. Lin, M. J. Zuo, and Z. He, “Condition monitoring and fault diagnosis of planetary gearboxes: A review,” *Measurement*, vol. 48, pp. 292–305, 2014.

[112] S. T. Kandukuri, A. Klausen, H. R. Karimi, and K. G. Robbersmyr, “A review of diagnostics and prognostics of low-speed machinery towards wind turbine farm-level health management,” *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 697–708, 2016.

[113] Y. Wang, J. Xiang, R. Markert, and M. Liang, “Spectral kurtosis for fault detection, diagnosis and prognostics of rotating machines: A review with applications,” *Mechanical Systems and Signal Processing*, vol. 66, pp. 679–698, 2016.