STATISTICAL ANALYSIS OF OPERATIONAL DATA FOR MANUFACTURING SYSTEM PERFORMANCE IMPROVEMENT

by

Zhenrui Wang

Copyright © Zhenrui Wang 2013

A Dissertation Submitted to the Faculty of the
DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2013
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Zhenrui Wang entitled STATISTICAL ANALYSIS OF OPERATIONAL DATA FOR MANUFACTURING SYSTEM PERFORMANCE IMPROVEMENT and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Date: 30 April 2013

Dr. Jian Liu

Date: 30 April 2013

Dr. Young-Jun Son

Date: 30 April 2013

Dr. Sean Dessureault

Date: 30 April 2013

Dr. Yue Niu

Date: 30 April 2013

Final approval and acceptance of this dissertation is contingent upon the candidate’s submission of the final copies of the dissertation to the Graduate College. I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Date: 30 April 2013

Dissertation Director: Dr. Jian Liu
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: Zhenrui Wang
ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Jian Liu, for his guidance, supervision, and encouragement throughout my research. Without his effort, this dissertation would not have become possible.

I would like to thank my other committee members, including Dr. Young-Jun Son, who has enthusiastically introduced to me many concepts of the manufacturing system; Dr. Sean Dessureault, who provided me guidance for an in-depth understanding of the real operational data that this research work focused on; and Dr. Yue Niu for giving me valuable suggestions on addressing technical difficulties.

I would also like to thank all the fellow students in the department and my friends, especially Mingyang Li, Qingpeng Zhang, Fujun Lan, and Dong Xu for their constructive comments and discussions.

Lastly, I would like to thank my parents and my wife Nan Zhang, who have unconditionally supported me. This dissertation is dedicated to them.
DEDICATION

To my loving parents, and my dear wife Nan
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................. 8  
LIST OF TABLES ................................................................. 9  
ABSTRACT .............................................................................. 10  

**CHAPTER 1** INTRODUCTION ................................................. 12  
  1.1 Motivation ................................................................. 12  
  1.2 Research problems ..................................................... 15  
  1.3 Related work ............................................................... 16  
    1.3.1 Identification of under-performing operators .......... 16  
    1.3.2 Pattern detection in temporal data sequence ......... 17  
    1.3.3 Failure prediction ................................................. 17  
  1.4 Overview of proposed methods ..................................... 19  
  1.5 Organization of the dissertation .................................... 20  

**CHAPTER 2** QUALITY DRIVEN WORKFORCE PERFORMANCE EVALUATION BASED ON ROBUST REGRESSION AND ANOMR/ANORMV CHART ................................................................. 21  
  2.1 Problem statement ....................................................... 21  
  2.2 Proposed methodology ................................................ 25  
    2.2.1 Quantification of other impacting factors’ effects .... 26  
    2.2.2 Identification of under-performing machines ......... 30  
    2.2.3 Identification of under-performing operators ........ 32  
    2.2.4 Summary of the proposed methodology ................. 35  
  2.3 Demonstration and validation ......................................... 38  
    2.3.1 Demonstration ...................................................... 38  
    2.3.2 Validation ........................................................... 44  
  2.4 Conclusion ................................................................. 47  

**CHAPTER 3** DETECTING ASYNCHRONOUS PERIODIC PATTERNS OF INTERVALS IN TEMPORAL SEQUENCE DATA ................................................. 49  
  3.1 Problem statement ....................................................... 49  
  3.2 Related work .............................................................. 55  
  3.3 Proposed methodology ................................................ 58  
    3.3.1 Hierarchical clustering for asynchronous periodicity  59
TABLE OF CONTENTS – Continued

3.3.2 Multi-start local search algorithm for modified hierarchical clustering .................................................. 62
3.3.3 Convolution-based periodicity detection ................................................................. 65
3.4 Case studies .................................................................................................................. 69
  3.4.1 Detecting APPs from simulated interval sequence ................................................. 70
  3.4.2 Detecting APPs from mining operation sequence ................................................. 72
3.5 Conclusion ................................................................................................................... 73

CHAPTER 4 PROPORTIONAL HAZARD MODEL INFEERENCE UNDER MIXTURE OF CLASSICAL AND BERKSON ERROR ........................................ 76
4.1 Problem statement ....................................................................................................... 76
4.2 Proposed approach ..................................................................................................... 80
  4.2.1 Modeling mixture of measurement error ........................................................... 80
  4.2.2 Model inference under mixture of error .......................................................... 82
4.3 Simulation study ......................................................................................................... 88
4.4 Conclusion .................................................................................................................. 90

CHAPTER 5 CONCLUSIONS AND FUTURE RESEARCH ........................................ 92
5.1 Conclusion .................................................................................................................. 92
5.2 Future work ............................................................................................................... 94

REFERENCES .................................................................................................................. 96
LIST OF FIGURES

1.1 Illustration of implicit information ........................................ 15
2.1 Relationship among $S_d$, $J$ and $N_P$ ................................. 28
2.2 Framework of proposed performance evaluation methodology ....... 35
2.3 Goodness of fit plot for case study ........................................ 42
2.4 Outcome of ANOMRV and ANOMR chart ............................... 43
2.5 Goodness of fit plot for validation based on simulated data ........... 46
3.1 Transformation from temporal sequences data to interval sequences with periodic patterns (Unit: min): (a) sequence with exact intervals and synchronous periodicity and (b) interval sequence with disturbance and asynchronous periodicity ........................................ 51
3.2 Clustering configurations and cluster index sequences: (a) clustering configuration I and the symbol sequence and (b) clustering configuration II and the symbol sequence ........................................ 52
3.3 Illustration of terms in objective function ............................... 61
3.4 Dendrogram for MSLS algorithm ......................................... 63
3.5 Flow chart of MSLS algorithm ........................................... 64
3.6 Illustration of the convolution based periodicity detection .......... 65
3.7 Performance evaluation in simulation study ............................. 71
3.8 Interval sequence and cluster index sequence ......................... 73
3.9 Pattern plots on real data ............................................... 74
LIST OF TABLES

2.1 Multiple hypothesis tests ........................................... 32
2.2 Summary of the dataset .............................................. 38
2.3 Frequency distribution of observations in the NoD dataset .......... 39
2.4 Estimates of final ZIP model coefficients .......................... 40
2.5 Identification of under-performing machines ...................... 41
2.6 Assigned mean and variance of machines and operators .......... 44
2.7 True coefficients of other impacting factors ....................... 44
2.8 False positive and false negative rate .............................. 46
3.1 Comparison of existing and proposed approach .................... 57
3.2 Performance index calculation in simulation study ................. 70
4.1 Performance comparison under different error variance settings .... 89
ABSTRACT

The performance of a manufacturing system relies on its four types of elements: operators, machines, computer system and material handling system. To ensure the performance of these elements, operational data containing various aspects of information are collected for monitoring and analysis. This dissertation focuses on the operator performance evaluation and machine failure prediction. The proposed research work is motivated by the following challenges in analyzing operational data. (i) the complex relationship between the variables, (ii) the implicit information important to failure prediction, and (iii) data with outliers, missing and erroneous measurements. To overcome these challenges, the following research has been conducted.

To compare operator performance, a methodology combining regression modeling and multiple comparisons technique is proposed. The regression model quantifies and removes the complex effects of other impacting factors on the operator performance. A robust zero-inflated Poisson (ZIP) model is developed to reduce the impacts of the excessive zeros and outliers in the performance metric, i.e. the number of defects (NoD), on regression analysis. The model residuals are plotted in non-parametric statistical charts for performance comparison. The estimated model coefficients are also used to identify under-performing machines. To detect temporal patterns from operational data sequence, an algorithm is proposed for detecting interval-based asynchronous periodic patterns (APP). The algorithm effectively and efficiently detects pattern through a modified clustering and a convolution-based template matching method. To predict machine failures based on the covariates with erroneous measurements, a new method is proposed for statistical inference of proportional hazard model under a mixture of classical and Berkson errors. The method estimates the model coefficients with an expectation-maximization (EM)
algorithm with expectation step achieved by Monte Carlo simulation. The model estimated with the proposed method will improve the accuracy of the inference on machine failure probability.

The research work presented in this dissertation provides a package of solutions to improve manufacturing system performance. The effectiveness and efficiency of the proposed methodologies have been demonstrated and justified with both numerical simulations and real-world case studies.
CHAPTER 1

INTRODUCTION

1.1 Motivation

Manufacturing is important to the technological development and economic prosperity of a nation. The United States has a top manufacturing value added equal to 1,771 billion around the world in 2010, and the manufacturing industries accounts for approximately 15% of gross domestic product (GDP) (Groover (2007)). The significant contribution of manufacturing industry to the welfare of the United States is attributed to the development and successful implementation of modern manufacturing system.

A modern manufacturing system contains mainly four elements: machines, human workers to operate and manage the machines, computer system and material handling system (Groover, 2007). Their performance must be guaranteed to ensure the satisfactory performance (e.g. quality, reliability and productivity) of manufacturing systems. For example, the performance of the whole system can be compromised when human workers perform poorly in operation, or when machines fail frequently. Thus, performance of operators and status of machines are monitored. Analysis of the operational data generated from such monitoring can provide insightful information to support decisions in operator performance evaluation and machine failure prediction. Based on the evaluation and prediction results, individualized training and preventive maintenance can be initiated in a timely manner, to improve the performance of operators and machines and thus, the overall system.

Analysis of the operational data, however, can be a challenging task due to the following facts.

- The operational data usually contain information of a variety of variables whose relationships are complex. Such variables may include operator per-
formance, production date and shift, level of operator experience and crew supervision, machine maintenance records, and machine health status, etc. Operator performance may be affected by the shift when he/she works, the level of the crew supervision, and/or the machine he/she operates, etc. Their relationships may be ambiguous and thus, are extremely difficult to quantify.

- The operational data often contain valuable information that is not directly observable. For example, operation log lists a sequence of alarms in a chronological order. The triggered alarms may indicate a machine health deterioration, which eventually leads to more devastating failures. However, it is difficult, if not impossible, to directly use alarm sequences to predict machine failures. The implicit information contained in alarm sequences needs to be detected and extracted as patterns, which explicitly represent temporal information directly related to failure.

- The operational data are often contaminated with outliers, missing or erroneous measurements. Such low quality data are frequently collected from real-world case studies. The relationship quantified and the patterns detected from such data will be either inaccurate or imprecise. Thus, evaluation and prediction results may be misleading. It is desirable to improve the robustness of the methods in analyzing operational data.

One application demonstrating these challenges can be found in the heavy industry such as mining. In mining industry, operation data are collected continuously for production monitoring. Plenty of sensors are installed on mining equipment, e.g. mining trucks, to collected equipment health and operation process data. In addition, the information regarding truck operators is also collected. Hundreds of Gigabytes of data with large number of variables and attributes are stored and maintained in database. In this data-rich environment, the relationship between different variables is often complex and unknown. This corresponds to the first challenge. For example, the truck operator performance may be affected by various factors such as operator experience, machine health status, and shift etc. As oper-
ator performance directly impacts the productivity of the operation, it is desirable that performance of all operators meets a specification or maintains constantly high. This is the reason operator performance data is recorded and analyzed. However, as multiple factors can influence the operator performance, direct comparison of operator performance without quantifying these effects produces misleading conclusion. The second challenge is related to analyzing the collected truck health event log. The event log recording the time stamp of each alarm occurrence can be crucial for understanding the truck health status and preventing truck failure event. However, the information stored in the log, such as time intervals between consecutive occurrence of alarms, is often subject to noise and cannot be directly used. Instead, the implicit information after removing these noises should be extracted and studied. Figure 1.1 illustrates the case. In the figure, the left panel shows the interval sequence data obtained, and the middle panel indicate the potential true time interval values along with the added noise. The patterns formed by these true interval values may be linked with different operation or health status and can serve as predictor for potential truck failure, as shown in the right panel. The third challenge comes from the data collection process in real-world operation. Due to the unreliable sensor readings and human mistakes, the collected data often contain erroneous, missing values or outliers. One example is the measured load weight data. Due to the limited accuracy of weight scales, the load weight measurement is subject to measurement error of +/- 5 tons. In addition, the weight scale often generates invalid measurement of zero tons even when truck is loading hundreds of tons of coal. The outliers, as observation of few measurements that are significantly larger than others, are observed in collected number of alarms occurred to each truck per shift. While most of the observations are less than 5, some large values about 30 are collected. These extremely large values may reflect the true data distribution, and therefore should not be directly removed. Advanced statistical method is needed to analyze the data with these outliers. Altogether, these challenges complicate the operational data analysis and motivate new research proposed in this dissertation.
1.2 Research problems

Corresponding to the challenges mentioned in the previous section, this dissertation aims at addressing the following three research problems.

- Effectively compare operator performance based on operational data and given operator performance metrics. As multiple factors may impact the operator performance, the comparison should quantify such impacts and eliminate them before the comparison. In addition, due to the existence of possible outliers in the data, the quantification should be insensitive to outliers.

- Efficiently detect temporal patterns from operational data sequence, e.g., alarm sequence. Specifically, the periodic patterns reserve the time interval information between alarms and are of special interest in machine failure prediction. However, due to random disturbance, the periodic patterns may be drifted, resulting in asynchronous periodic patterns (APPs). There is a need of systematic method to first tolerate the disturbance and thus reserve the periodicity, and then expedite the detection of the periodic patterns from massive alarm sequences.
Robustly predict machine failures based on the error-prone covariates, including those obtained from the detected periodic patterns. The errors may come from the measurements and data processing. Therefore, to use them as covariates, a modeling inference method that takes into account the erroneous measurement must be developed. Furthermore, multiple types of errors often coexist in covariates. This makes the statistical inference even more challenging. Therefore, it is crucial to develop a methodology to improve the estimation by taking into account of the errors associated with the covariates.

1.3 Related work

Corresponding to the research problems identified in the previous section, the proposed research work is decomposed into three components: (i) operator performance evaluation, i.e. identification of under-performing operators; (ii) pattern detection in temporal data sequence, i.e. extraction of temporal patterns that may be useful for failure prediction; (iii) failure prediction model inference, i.e. estimation of model coefficients when covariates are measured with errors. A comprehensive review of the existing methods in these fields are presented in this section.

1.3.1 Identification of under-performing operators

The identification of under-performing operators needs to be based on certain performance metrics. Bommer et al. (1995) summarized a variety of metrics adopted in practice of workforce performance evaluation. Among them, quality-based metrics indicate the operators’ performance evaluated by product quality. In this dissertation, the number of defects (NoD) generated by an operator in a certain period of time is used as such a quality-based metric.

Intuitively, the identification of under-performing operators may be conducted by comparing the NoDs grouped by operator ID. This is equivalent to the multiple comparisons of means and variances. Existing methods for such comparisons include multiple hypothesis tests by Carroll and Schneider (1985), analysis of means
(ANOM) chart (Ott, 1967), analysis of means for variance (ANOMV) chart (Nelson et al., 2005), ANOM using ranks (ANOMR) chart (Bakir (1989)) and ANOMR-type test for homogeneity of variances (ANOMRV) chart (Bakir (2010)) etc.

These existing approaches have various drawbacks. Some approaches cannot identify the under-performing operators or analyze non-normal data. Their common limitation is that due to the complex relationship between operator performance and a variety of impacting factors, none of these methods can be directly used in comparison.

1.3.2 Pattern detection in temporal data sequence

Limited research has been conducted to detect periodic patterns directly from temporal sequences. Ma and Hellerstein (2001) introduced the problem of asynchronization in time of occurrence of event when analyzing periodic patterns with unknown period. The method extended previous approach by Han et al. (1999) to detect periodic patterns with consideration of asynchrony. Lee et al. (2008) defined fuzzy periodicity with fuzzy periodic calendar aiming to handle pattern detection under disturbance.

Both methods require specification of a tolerance value with respect to disturbance. However, such specification is subjective and therefore may be misleading if mis-specified. In addition, Ma and Hellerstein (2001) only used inter-arrival time as a candidate pool for pattern period and method by Lee et al. (2008) only detected patterns with period of 1.

1.3.3 Failure prediction

For the topic of failure prediction, various methods have been proposed. Kumar and Klefsjö (1992) conducted a preliminary study of reliability statistics of a fleet of load-haul-dump machines and predicted the occurrence of failures using power law process models. Vilalta and Ma (2002) proposed a rule-based approach that predicted rare failure event based on extracted eventset patterns prior to the failure.
Lin and Siewiorek (1990) developed dispersion frame technique (DFT) to analyze the time of warning signals in near past and used heuristic rules to predict failure. Salfner and Malek (2007) adopted hidden semi-Markov models for failure prediction. Apart from these methods, reliability models such as Cox proportional hazard model have been adopted for machine failure prediction, such as the methodology proposed by ?.

As a well-established statistical method in analyzing time to failure data, the Cox proportional hazard model assumes that covariates are error-free. However, this assumption can be violated in many real-world applications. Existing approaches consider mainly two types of measurement errors associated with covariates, i.e., classical error and Berkson error. Abundant research efforts have been made to study Cox proportional hazard model with classical measurement error. Hughes (1993) analyzed the effect of the classical measurement error on parameter estimation and concluded that the occurrence of classical errors leads to biased estimates. To address the estimation bias, many existing approaches to handling measurement error were adopted or extended to analyze Cox model. The regression calibration technique, which replaces the covariate of interest by its conditional expectation given the observed covariate, was proposed by Prentice (1982). The simulation-extrapolation (SIMEX) procedure proposed by Cook and Stefanski (1994) represented another mainstream technique. The other group of methods, initiated by research work from Nakamura (1992), proposed a corrected score function so that even using the erroneous covariate still results in an approximately and asymptotically unbiased estimate. However, for Cox model estimation, corrections are only approximate. The likelihood-based approach proposed in Hu et al. (1998) differed from the above-mentioned approaches in that the unobserved covariate is treated as independent normal random effects and a full likelihood function consisting of joint density functions of observed covariate and true covariate is used.

In contrast to the variety of methods proposed for Cox model with classical measurement error, the research of estimation of regression coefficient under the mixture of classical errors and Berkson errors is relatively limited. Reeves et al.
(1998) pioneered the research work of mixture of errors and discussed how the errors affect the estimates from logistic regression in an epidemiological study. Mallick et al. (2002) proposed Bayesian methods for semi-parametric estimation approach in an analysis where the response variable was binary and covariates were measured with mixture of errors. A study considering the correlated structure of the Berkson error in the mixture of errors was conducted by Li et al. (2007a). Carroll et al. (2007) proposed a non-parametric regression approach that used kernel estimation and characteristic functions of Berkson and classical error. The estimation procedure of Cox proportional hazard model under the mixture of errors, however, has not been studied.

1.4 Overview of proposed methods

Compared to existing research conducted to the aforementioned three research problems identified in section 1.2, this dissertation has made the following original contributions.

To compare the operator performance of different operators, a methodology is proposed based on the regression modeling and multiple comparisons technique. The methodology enables the comparison of operator performance by quantifying and removing the complex effects of machines and other impacting factors with a regression model. In order to construct the regression model, the challenges caused by the NoD data collected directly from production operations are addressed. In practice, NoD data often contain excessive number of zeros, which make the Poisson distribution inadequate to model the counts of NoD. In addition, NoD data may contain outliers, whose values are significantly larger than that of majority of data. It should be noted that the outliers may come from both normal- and under-performing operators. To reduce the impact of excessive zeros and outliers on model fitting, robust zero-inflated Poisson (ZIP) model is adopted in such a situation. The residuals of the robust ZIP regression model are used as the independent inputs to the ANOMR/ANOMRV charts for performance comparison. The esti-
mated regression model coefficients are used in multiple hypothesis tests to identify under-performing machines. This identification enables preventive maintenance to avoid further performance deterioration and reduce machine downtime. The identification of both under-performing operators and under-performing machines will help reduce NoD level and improve product quality. This eventually improves the performance of the manufacturing system.

To detect temporal patterns from operational data sequence, an algorithm is proposed for detecting interval-based asynchronous periodic patterns (APP). The proposed algorithm allows objective determination of tolerance values while existing methods require subjective specification. The algorithm also modifies existing clustering methods by adding new terms in the clustering objective functions to obtain patterns with better interpretability. Based on the symbol sequence generated by this modified clustering method, a convolution-based template matching approach is developed to enhance pattern detection efficiency and improve the mapping flexibility.

To predict the machine failure based on the error-prone covariates, a new method is proposed for statistical inference of Cox proportional hazard model. Compared with existing approaches, this dissertation studies the Cox model under a mixture of classical and Berkson errors. The inference method treats the unobservable errors as latent variables and develops an EM algorithm with Monte Carlo simulation.

1.5 Organization of the dissertation

This dissertation presents the research effort in analyzing operational data for the purpose of performance improvement in manufacturing systems. The proposed methodology incorporates three components, operator performance evaluation, APP detection in temporal data sequence, and Cox model inference with erroneous covariates. Chapter 2, Chapter 3 and Chapter 4 are organized as individual research papers, addressing these research problems, respectively. The conclusion of the research work and future directions are summarized in Chapter 5.
CHAPTER 2

QUALITY DRIVEN WORKFORCE PERFORMANCE EVALUATION BASED ON ROBUST REGRESSION AND ANOMR/ANORMV CHART

2.1 Problem statement

Product quality improvement has been extensively studied in the last five decades. Classical techniques developed from this effort, such as statistical process control and total quality management, have significantly enhanced manufacturers’ capability in delivering quality products and thus, their competitiveness. With the advent of modern manufacturing technologies, manufacturers adopt more advanced methodologies, such as flexible manufacturing and reconfigurable manufacturing, to succeed in an increasingly demanding global market. The requirements on flexibility and reconfigurability for the new manufacturing systems lead to greater complexity in system/process design and quality management. Such complexity becomes the driving impetus for state-of-the-art quality assurance research by (Koren et al. (1999); Shi (2006); Li and Shi (2007); Liu et al. (2009); Wang et al. (2010)).

While researchers have made significant improvements in quality assurance of complex manufacturing system, less attention was paid to the integration of quality control and the management of manufacturing systems. This type of integration has emerged as a promising research topic in recent years. Early efforts have been focused on quantifying and integrating three measures of manufacturing systems: productivity, quality and flexibility/reconfigurability for overall evaluation as demonstrated by Son and Park (1987), and Koren and Shpitalni (2010). Khouja et al. (1995) discussed the trade-off between productivity and quality and the product quality was used as a metric to guide the system design. Quality assurance activities have been linked with various aspects of manufacturing system management. Urban (1998) investigated the relationship between batch size and quality.
Li and Huang (2007) analyzed the effects of various production scheduling schemes with regard to product quality. Humphrey et al. (1998) studied the behaviors of different inventory stocking methodologies in repair and rework operations. Advanced research by Li et al. (2008) was conducted recently to guard against fluctuation in the first time quality by optimally designing the repair and rework system.

As a significant factor of manufacturing system, operators have direct impacts on product quality. Thus, it is critical to ensure that operators’ performance, which can be quantitatively evaluated based on certain performance metrics, meets production requirements. The performance evaluation is an important component of workforce management, which is of great concerns to the manufacturing system management. From the evaluation outcome, under-performing operators can be identified. Analysis of their performance discrepancy will unveil the causes and enforced training programs for these operators may improve their performance.

A variety of metrics have been adopted in practice of workforce performance evaluation, as summarized by Bommer et al. (1995). Among them, quality-based metrics indicate the operators’ performance in terms of product quality. For instance, the number of defects (NoD) generated by an operator in a certain period of time is often used as such a quality-based metric. In addition to operators’ performance, other impacting factors may contribute to the NoD. For instance, machines with different performance may generate different NoDs, even if they are operated by the same operator. In other words, the NoDs of different operators are not independent with the effects of other impacting factors. Therefore, operators’ performance should be compared after quantifying and removing those effects. Besides, the quantification may bring extra insights for quality improvement, such as the identification of machines that, on average, produce higher NoDs than a nominal level. This leads to the following definition:

**Definition 1:** an under-performing operator/machine is defined as the operator/machine that generates excessively large mean and/or variance of NoD, given the effects of other impacting factors considered.

After identifying under-performing machines, proactive activities, such as
scheduling maintenance service, can be taken to avoid further performance deterioration and reduce machine downtime. Therefore, the identification of under-performing operators and machines reduces NoD level and improves product quality. This eventually increases the productivity.

Intuitively, the identification of under-performing operators may be conducted by comparing the NoDs grouped by operator ID. This is equivalent to the multiple comparisons of means and variances. Standard statistical methods for such comparisons include multiple hypothesis tests by Carroll and Schneider (1985), analysis of means (ANOM) chart (Ott, 1967), analysis of means for variance (ANOMV) chart (Nelson et al., 2005), ANOM using ranks (ANOMR) chart Bakir (1989) and ANOMR-type test for homogeneity of variances (ANOMRV) chart Bakir (2010) etc. Multiple hypothesis tests, such as F-test in ANOVA and Levene’s test, give conclusions on whether different groups have the same mean and variance, respectively. The drawback of these tests is that they do not readily identify which groups are different and require post-hoc procedures. The ANOM/ANOMV chart addresses this issue by directly pinpointing groups with different means or variances, respectively, but these charts have normality assumption that limits their applications to non-normal NoD data. The nonparametric ANOMR/ANOMRV charts are potential solutions for comparisons of non-normal data.

However, multiple comparison methods for non-normal NoD data, such as ANOMR and ANOMRV charts, cannot be directly applied. As aforementioned, the NoDs are not independent without removing the effects of other impacting factors before comparison. This problem can be solved by fitting a regression model with other impacting factors (including machine) as predictor variables. The model residuals are independent and thus can be used as direct measures of operators’ performance. Furthermore, the model isolates the effect of machine on the NoD from other factors, and thus can be used to identify under-performing machines. Therefore, this chapter proposes a methodology for workforce performance evaluation based on regression modeling and multiple comparisons technique. The methodology enables the comparison of operator performance by quantifying and removing
effects of machines and other impacting factors with a regression model.

To apply the proposed methodology to real-world applications, the challenges caused by the NoD data collected directly from production operations need to be addressed. In practice, NoD data often contain excessive number of zeros, which make the Poisson distribution inadequate to model the counts of NoD. In addition, real-world NoD data may contain outliers, whose values are significantly larger than that of majority of data. It should be noted that the outliers may come from both normal- and under-performing operators. To reduce the impact of excessive zeros and outliers on model fitting, robust zero-inflated Poisson (ZIP) model (Hall and Shen (2010)) is adopted in such a situation. The model quantifies the effects of other impacting factors and identifies under-performing machines. The independent, non-normal Pearson residuals of the regression model are plotted in nonparametric ANOMR/ANOMRV chart. Under-performing operators are identified by comparing multiple means and variances of the residuals grouped by operators. Then the performance of these under-performing machines and operators can be improved by maintenance checks and enforced training, respectively. The manufacturing system will generate lower NoD levels, reduce process fallout by producing larger proportion of conforming products, and increase the productivity measured by overall equipment effectiveness (OEE) (Ljungberg (1998)).

Robust ZIP regression relies on outlier-resistant estimators such as fast least trimmed square (FAST-LTS) algorithm to improve the probability of convergence of the regression algorithm. However, the effectiveness of the FAST-LTS can be seriously degraded when some or all predictor variables in the regression model are categorical. To improve the effectiveness of model fitting, a modified sampling scheme is proposed for FAST-LTS.

This chapter provides a systematic solution to the workforce performance evaluation that can improve both the quality and productivity. More specifically,

- A new methodology is proposed for multiple comparisons with the quantification of other impacting factors.
A combination of robust model fitting and non-parametric multiple comparisons technique is adopted to meet the challenges posed by data from real-world production operations.

A modified sampling scheme is developed to improve the effectiveness of FAST-LTS for regression model with categorical predictors.

The remainder of this chapter is organized as follows. Section 2.2 decomposes the workforce evaluation task into three sub-problems, quantification of effects of other impacting factors, identification of under-performing machines, and identification of under-performing operators, and explains the proposed solution to each sub-problem. Section 2.3 discusses a case study on data from a real-world production system and a simulation experiment to demonstrate the effectiveness of the proposed approach. The chapter concludes with a summary of the methodology and a discussion on future research in section 2.4.

2.2 Proposed methodology

This chapter proposes a methodology of operator performance evaluation to improve quality as well as productivity of manufacturing system. The following three sub-problems are addressed:

- Quantification of effects of other impacting factors (including machine)
- Identification of under-performing machines
- Identification of under-performing operators

The solutions to these three sub-problems are introduced in subsection 2.2.1 to subsection 2.2.3, respectively. Subsection 2.2.1 focuses on the robust ZIP regression modeling to quantify the effects of other impacting factors. The limitation of FAST-LTS for models with categorical predictors is overcome by proposing a modified sampling algorithm. Based on the regression coefficient, subsection 2.2.2 identifies under-performing machines with multiple hypothesis tests. Subsection 2.2.3 explains
the comparison of grouped Pearson residuals using ANOMR/ANOMRV chart. An implementation procedure is summarized in subsection 2.2.4.

### 2.2.1 Quantification of other impacting factors’ effects

In the proposed methodology, the first step is to quantify the relationship between the performance metric, NoD, and other impacting factors, using regression model. The regression model coefficients quantify predictors’ (i.e., other impacting factors’) effects on the response variable and the residual represents the variation that cannot be explained by the model. Since NoDs are counted within a pre-specified unit of time, Poisson regression can be selected. However, the distribution of count data is often more complex. For instance, NoD data may contain excessive proportion of zeros, compared to a Poisson distribution. This phenomenon, referred to as zero inflation, is often observed in NoD data. It makes the statistical inference based on Poisson distribution inaccurate and misleading.

To model NoD data with zero inflation, the zero-inflated Poisson (ZIP) distribution (Cohen (1963)) was introduced. A ZIP random variable comes from either a constant zero value or a Poisson distribution with mean $\lambda$. Denoting the probability of an observation being a constant zero by $p$, the ZIP distribution can be fully specified with $\lambda$ and $p$. Based on the distribution, ZIP regression model (Lambert (1992)) was later proposed and its coefficients are estimated with maximum likelihood estimator (MLE). Due to the use of MLE, sensitivity to outliers becomes a practical concern in applying ZIP regression to possibly contaminated data, such as the NoD data collected from real production operations. To improve the robustness, robust ZIP regression model was proposed to improve the model estimation with contaminated data.

Denoting each observation in NoD dataset as the response variable, $Y_i$, it can be associated with its corresponding $p_i$ and $\lambda_i$. where $i = 1, 2, ..., N$, and $N$ is the sample size. The robust ZIP regression models $Y_i$ with Poisson and logistic
regression as follows:

\[
\log(\lambda_i) = B_i^T \beta \quad \text{and} \quad \log\left( \frac{p_i}{1 - p_i} \right) = G_i^T \gamma
\]  

(2.1)

where \( \beta \) and \( \gamma \) are column vectors of unknown regression coefficients; \( B_i \) and \( G_i \) are column vectors consisting of predictor variables for \( Y_i \) in Poisson and logistic regression, respectively.

Robust ZIP regression requires a good initial estimate of coefficients \( \beta \) to improve the probability of convergence. FAST-LTS algorithm proposed by Rousseeuw and Van Driessen (2006) can be used to produce such an estimate, denoted as \( \hat{\beta}_0 \), using all positive \( Y_i \)'s in NoD data. While the algorithm is effective with continuous predictor variables, its effectiveness can be seriously reduced when (some or all) predictor variables are categorical.

The issue is related to the initial step of FAST-LTS algorithm, where coefficient candidates are generated. In this initial step, the method first collects \( D \) disjoint random sample sets \( S_d, \ d = 1, 2, ..., D, \) of the same size \( s \). The sample size \( s \) should be large enough to make \( S_d \) a representative sample of the whole dataset. Based on the case study in Rousseeuw and Van Driessen (2006), empirical values of \( s = 300 \) and \( D = 5 \) are recommended. Since the observations may contain outliers, coefficient candidates based on these data are likely to be biased. To reduce the possible effects of outliers, FAST-LTS repeatedly selects a random subset of size \( [(h/N_P)S_d] \) from \( S_d \) and denotes it as \( H_I \). Here \([.]\) represents rounding calculation. \( N_P \) is the set of positive \( Y_i \)'s and \( h \) is the trimming constant, which reflects the belief on the number of non-outliers and can be chosen as any integer between \( N_P/2 \) and \( N_P \). Initial coefficient candidates are estimated based on \( H_I \) as the subset is less likely to contain outliers.

To obtain \( H_I \), a random subset \( J \) containing \( p \) observations is selected from \( S_d \), where \( p \) corresponds to the number of coefficients to estimate in FAST-LTS, i.e. the dimension of vector \( \beta = [\beta_1, \beta_2, ..., \beta_p]^T \). The relationship between sets \( S_d \)'s, \( J \)'s and \( N_P \) is illustrated in figure 2.1, which shows that \( S_d, \ d = 1, 2, ..., D \) are subsets
of $N_P$ and set $J$ is contained in $S_d$. The circles represent elements of $N_P$, i.e. the positive observations of NoD data. The blank circles indicate normal observations whereas the filled circles represent outliers. From set $J$, a very rough estimate for $\beta_0$ can be determined by solving equation of $Y_J = X_J \beta$, where $Y_J$ consists of all $p$ observations and $X_J$ is the design matrix of predictor variables from observations in $J$ after transformation of data as described by Shen (2006). Then for all the observations in $S_d$, the residuals, $r_k = Y_k - V_k^T \hat{\beta}_0$, $k = 1, 2, ..., s$ are calculated, where the column vector $V_k$ consists of predictor variables corresponding to $k$th observation $Y_k$. Next, observations are ranked in ascending order of $|r_k|$ and the first $h$ observations constitute the set $H_I$.

The random selection of $p$ observations may result in a singular $X_J$. For model with continuous predictors, the singularity problem may be solved by adding random observations to subset $J$ until $X_J$ has full rank, as proposed by Rousseeuw and Van Driessen (2006). However, for models with categorical predictors, the singularity problem mainly attributes to the absence of the effects of a certain level(s) of a categorical predictor, which are coded with dummy variables. To estimate $p$ coefficients, the subset $J$ must at least contain one observation for the effect of each dummy variable. Otherwise, $X_J$ will have a column of zeros and become singular. As the sampling is completely random, the set $J$ with more observations does not necessarily contain effects of all the predictor variables. Meanwhile, increasing the size of subset $J$ leads to a higher probability of containing outliers in $J$. When $J$ contains outliers, the set $H_I$ obtained from $J$ adversely affects the performance of
FAST-LTS, so simply adding random observations is not an effective solution.

To address this issue, this chapter imposes restrictions on the sampling of \( p \) observations. To ensure that all the predictors’ effects are present in each group of collected \( p \) observations, \( X_J \) cannot have any column of zeros. Specifically, the following steps, denoted by “SR” (“SR” stands for “steps for modified random sampling”), are proposed.

**SR1. Construct index sets.** For columns representing \( p \) effects of \( \beta_1, \beta_2, \ldots, \beta_p \) in the design matrix based on set \( S_d \), define the corresponding \( p \) index sets as \( R_z, z = 1, 2, \ldots, p \). \( R_z \) contains the row indices where the matrix \( S_d \) has nonzero values in column \( z \). Here nonzero values indicate the presence of effect of certain categorical predictor variables. These nonzero values result from the transformation of dummy variables that are equal to one in the original design matrix.

**SR2. Sample random observations.** One observation index is randomly selected from each \( R_z \) and the corresponding rows in the design matrix of \( S_d \) are combined to form matrix \( X_J \). This ensures that each factor’s effect appears in the selected \( p \) observations. The probability of singularity, which now mainly results from having equal values in multiple rows, is much smaller.

In case that \( X_J \) is singular, a random sample of size \( t (t = 1) \) can be obtained from the rest of \( S_d \) and combined with existing \( p \) observations. A new set \( J^* \), vector and design matrix are formed based on combined observations. If \( X^*_J \) is still singular, step **SR2** is repeated and the obtained \( p \) observations are combined with a sample of size \( t = t + 1 \) from the rest of \( S_d \) to form a new \( X^*_J \). This iteration ends until \( X^*_J \) is non-singular, then the estimate for \( \beta_0 \) can be determined by solving a linear regression with value of the response and predictor variables specified in \( Y^*_J \) and \( X^*_J \), respectively. In addition, \( S_d \) needs to be checked to ensure the presence of effects of all factors. The column-wise sum is calculated for the design matrix from set \( S_d \). If the obtained sum vector contains zero elements, some predictors’ effects are missing in set \( S_d \). In this case, it is adequate to simply resample all the sets until no \( S_d \) contains missing effects.
2.2.2 Identification of under-performing machines

After the effects of each individual machine on the NoD are quantified with robust ZIP regression, under-performing machines can be identified by comparing the estimates of population mean and variance of NoD for each machine through hypothesis test. This involves the following steps, denoted by “SH” (“SH” stands for “steps in multiple hypothesis tests”):

**SH1.** Derive the distribution for estimates of population mean and variance. The comparison is conducted by setting all the predictors, except machine, to benchmark levels. Without loss of generality, assume that there are \( M \) machines (with ID \( j = 1, 2, ..., M \)) in the study. The effect of machine with ID 1 on NoD is included in the estimated intercepts, \( \hat{\gamma}_* \) and \( \hat{\beta}_* \), of logistic regression and Poisson regression, respectively. The effects of the other \( M - 1 \) machines are calculated as \( \hat{\gamma}_* + \hat{\gamma}_{[j]} \) and \( \hat{\beta}_* + \hat{\beta}_{[j]} \), where \( \hat{\gamma}_{[j]} \) and \( \hat{\beta}_{[j]} \), \( j = 2, ..., M \), are the coefficient estimates of machine \([j]\). Define

\[
\begin{align*}
   a_{[j]} &= \begin{cases} 
   \hat{\gamma}_* & \text{if } j = 1, \\
   \hat{\gamma}_* + \hat{\gamma}_{[j]} & \text{if } j \neq 1,
   \end{cases} \\
   b_{[j]} &= \begin{cases} 
   \hat{\beta}_* & \text{if } j = 1, \\
   \hat{\beta}_* + \hat{\beta}_{[j]} & \text{if } j \neq 1,
   \end{cases}
\end{align*}
\]

(2.2)

The estimates of population mean and variance of NoD for machine \([j]\) can be calculated as

\[
\hat{\mu}_{[j]} = (1 - \hat{p}_{[j]})(1 - \hat{\mu}_{[j]}) = f(a_{[j]}, b_{[j]}) = \frac{\exp(b_{[j]})}{1 + \exp(a_{[j])}}, \quad (2.3)
\]

\[
\hat{\sigma}_{[j]}^2 = (1 - \hat{p}_{[j]})(1 + \hat{\mu}_{[j]})(1 + \hat{\mu}_{[j]}) = g(a_{[j]}, b_{[j]})
= \frac{\exp(b_{[j]})}{1 + \exp(b_{[j]})} \left( 1 + \frac{\exp(b_{[j]})\exp(a_{[j]})}{1 + \exp(a_{[j]})} \right), \quad (2.4)
\]

where \( \hat{p}_{[j]} \) is the probability of NoD being a constant zero and \( \hat{\mu}_{[j]} \) is the estimated Poisson mean for machine \([j]\) based on the regression coefficients.
Hall and Shen (2010) conclude that regression coefficients $\hat{\theta} = [\hat{\gamma}^T, \hat{\beta}^T]^T$ asymptotically follow normal distribution, i.e., $\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{D} N(0, V)$, where $\theta$ denotes the true values of the coefficients and $V$ is the asymptotic variance-covariance matrix $V_N$. Since $\hat{\beta}_*, \hat{\gamma}_*$, $\hat{\beta}_{[j]}$'s and $\hat{\gamma}_{[j]}$'s are elements of vector $\hat{\beta}$ and $\hat{\gamma}$, these estimates can be approximated by normal distribution given a large sample size. For example, $\hat{\beta}_*$ asymptotically follows normal distribution with mean $\beta_*$ (i.e. true value of the coefficient) and variance $\sigma^2_{\beta_*}$ obtained from matrix $V_N$. $\hat{\beta}_* + \hat{\beta}_{[j]}$ follows the asymptotic distribution of $N(\beta_* + \beta_{[j]}, \sigma^2_{\beta_*} + \sigma^2_{\beta_{[j]}} + \text{cov}(\hat{\beta}_*, \hat{\beta}_{[j]}))$. The distribution of estimates of population mean $\hat{\mu}_{[j]}$ and variance $\hat{\sigma}^2_{[j]}$ of NoD can be approximated with delta method as

$$N\left(\frac{\exp(\mu_{b_{[j]}})}{1 + \exp(\mu_{a_{[j]}})} \left(\frac{\partial f}{\partial a_{[j]}|_{\mu_{a_{[j]}}}}\right)^2 \sigma^2_{a_{[j]}} + \left(\frac{\partial f}{\partial b_{[j]}|_{\mu_{b_{[j]}}}}\right)^2 \sigma^2_{b_{[j]}}, \left(\frac{\partial f}{\partial b_{[j]}|_{\mu_{b_{[j]}}}}\right)^2 \sigma^2_{b_{[j]}}\right)$$

(2.5)

$$N\left(\frac{\exp(\mu_{b_{[j]}})}{1 + \exp(\mu_{a_{[j]}})}\left(1 + \frac{\exp(\mu_{b_{[j]}})\exp(\mu_{a_{[j]}})}{1 + \exp(\mu_{a_{[j]}})}\right), \left(\frac{\partial f}{\partial a_{[j]}|_{\mu_{a_{[j]}}}}\right)^2 \sigma^2_{a_{[j]}} + \left(\frac{\partial f}{\partial b_{[j]}|_{\mu_{b_{[j]}}}}\right)^2 \sigma^2_{b_{[j]}}\right)$$

(2.6)

respectively. Given that $\hat{\mu}_{[j]}$ and $\hat{\sigma}^2_{[j]}$, $j = 1, 2, ..., M$, follow asymptotically normal distributions, under-performing machines are identified through multiple hypothesis tests. To construct the tests, estimates $\hat{\mu}_{[j]}$ and $\hat{\sigma}^2_{[j]}$ are calculated using Eq.(2.3) and Eq.(2.4). In addition, $\text{var}(\hat{\mu}_{[j]})$ and $\text{var}(\hat{\sigma}^2_{[j]})$ need to be determined. As shown in Hall and Shen (2010), vector $\hat{\theta} = [\hat{\gamma}^T, \hat{\beta}^T]^T$ is an almost surely consistent estimator for $\hat{\theta} = [\gamma^T, \beta^T]^T$. Under mild regularity condition, $\mu_{a_{[j]}}$ and $\mu_{b_{[j]}}$, $j = 1, ..., M$, can be estimated with $a_{[j]}$ and $b_{[j]}$, respectively, and $\text{var}(\hat{\mu}_{[j]})$ and $\text{var}(\hat{\sigma}^2_{[j]})$ are calculated as Eq.(2.7), where the value of $\sigma^2_{a_{[j]}}$ and $\sigma^2_{b_{[j]}}$ can be readily found in matrix $V_N$ from regression outcome

$$\text{var}(\hat{\mu}_{[j]}) = \text{var}(\hat{\sigma}^2_{[j]}) = \left(\frac{\partial g}{\partial a_{[j]}|_{a_{[j]}}}\right)^2 \sigma^2_{a_{[j]}} + \left(\frac{\partial g}{\partial b_{[j]}|_{b_{[j]}}}\right)^2 \sigma^2_{b_{[j]}}$$

(2.7)

**SH2.** Conduct multiple hypothesis tests. Multiple hypothesis tests are conducted to compare the population mean/variance of each machine with benchmark values. Two comparison procedures with Bonferroni adjustment are suggested de-
pending on the availability of domain knowledge regarding benchmark performance. If the upper and lower specification limits for the mean and variance of NoD are known from domain knowledge, these limits (denoted as $U_\mu$, $L_\mu$, $U_{\sigma^2}$ and $L_{\sigma^2}$, respectively) are used as knowledge based benchmarks and the super-performing machines can also be identified. Otherwise, the proposed methodology considers benchmarks solely based on the data. Under this condition, the maximum estimates of mean $\hat{\mu}_{\max}$ and variance $\hat{\sigma}^2_{\max}$ of NoD, among all the machines in comparison, are used as the benchmarks. By collecting IDs of machines that fail to reject the null hypotheses of these tests (summarized in table 2.1), the under-performing/super-performing machines are identified.

It is noted that when sample size gets very large, even a small difference between the mean/variance of one machine and the benchmark value will result in a rejection of null hypothesis. To avoid wrong decisions caused by such purely data-driven approach, a reexamination of the results is suggested to combine the statistical testing results with domain expertise.

<table>
<thead>
<tr>
<th>Benchmark based on knowledge</th>
<th>Multiple hypothesis tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Underperforming</td>
<td>$H_0: \hat{\mu}<em>{[j]} = U</em>\mu$, $H_1: \hat{\mu}<em>{[j]} &gt; U</em>\mu$</td>
</tr>
<tr>
<td>Overperforming</td>
<td>$H_0: \hat{\mu}<em>{[j]} = L</em>\mu$, $H_1: \hat{\mu}<em>{[j]} &gt; L</em>\mu$</td>
</tr>
<tr>
<td>Underperforming</td>
<td>$H_0: \hat{\mu}<em>{[j]} = U</em>{\sigma^2}$, $H_1: \hat{\mu}<em>{[j]} &gt; U</em>{\sigma^2}$</td>
</tr>
<tr>
<td>Overperforming</td>
<td>$H_0: \hat{\mu}<em>{[j]} = L</em>{\sigma^2}$, $H_1: \hat{\mu}<em>{[j]} &gt; L</em>{\sigma^2}$</td>
</tr>
</tbody>
</table>

Benchmark based on data

| Underperforming | $H_0: \hat{\mu}_{[j]} = \hat{\mu}_{\max}$, $H_1: \hat{\mu}_{[j]} < \hat{\mu}_{\max}$ |
| Overperforming | Not available due to lack of justifiable benchmark |
| Underperforming | $H_0: \hat{\sigma}^2_{[j]} = \hat{\sigma}^2_{\max}$, $H_1: \hat{\sigma}^2_{[j]} < \hat{\sigma}^2_{\max}$ |
| Overperforming | Not available due to lack of justifiable benchmark |

2.2.3 Identification of under-performing operators

Based on the robust ZIP regression, effects of other impacting factors in the regression model can be quantified and under-performing machines are identified. Next,
these effects should be eliminated to obtain a direct and independent measure for operator performance evaluation. Given that the effects of other impacting factors are quantified by the regression model coefficients, the model residuals can be grouped by operator ID to serve as such a performance measure.

To retain the physical meaning of response variable (i.e. NoD) and account the effects of regression coefficients on both the population mean and variance, Pearson residual is selected for operator performance comparison. The residual is defined as 

\[ R_i = \frac{(Y_i - \hat{\mu}_i)}{\sqrt{\hat{\text{var}}(Y_i)}} \]

where \( \hat{\mu}_i = (1 - \hat{p}_i)\hat{\lambda}_i \) and \( \hat{\text{var}}(Y_i) = (1 - \hat{p}_i)\hat{\lambda}_i(1 + \hat{p}_i\hat{\lambda}_i), \) \( i = 1, 2, ..., n. \) Since the ANOMRV/ANOMR chart takes Pearson residual as input, the effectiveness of the charts relies on the goodness of fit of the regression model. A mis-specified model will produce erroneous residuals and make the corresponding identification results unreliable. As a result, the model fitting needs to be checked before applying ANOMRV/ANOMR charts to Pearson residual. This chapter adopts the same chi-square goodness of fit plot proposed by Hall and Shen (2010). The Pearson chi-square goodness-of-fit statistics are plotted against the quantiles of a \( \chi^2_{(1)} \) random variable. The idea behind this plot is that if the model is correctly specified, the statistics that are calculated from observations without outliers should fall approximately along a line inside an envelope. This envelope represents the estimated range of these statistics and is conceptually similar to a confidence interval. In contrast, the statistics that are calculated from observations with outliers will fall outside and away from the envelope.

As aforementioned, both ANOM/ANOMV and ANOMR/ANOMRV charts are effective tools for detecting groups with significant different means/variances than overall levels. However, Pearson residuals may not be normally distributed in many situations (for example, due to existence of outliers); therefore the ANOMR and ANOMRV charts are adopted for multiple comparisons. The chart is characterized with centerline (CL), upper and lower decision limit (UDL and LDL). It identifies groups with their mean/variance values beyond the UDL or LDL as the ones that are different. Although either chart is distribution-free method, they each have their own assumptions. ANOMR chart assumes homogeneity of variance and ANOMRV
assumes that the grouped data have equal means. For Pearson residuals grouped by operators with no equal mean or variance assumption, these charts cannot be applied directly. Suppose that the NoD data are collected from \( m \) operators, and each operator has \( n_g \) observations, \( g = 1, 2, ..., m \), the sample size is \( N \). Denoting the \( l \)th Pearson residual from operator \( g \) as \( R_{g,l} \), the following steps, denoted by “SC” (“SC” stands for “steps in charting”), are proposed:

**SC1. Transform Pearson residuals.** Since ANOMRV chart assumes equal group mean between compared groups, the residuals are firstly transformed to have all group means equal to zero. This is done by subtracting each \( R_{g,l} \) by corresponding group mean, denoted as \( \bar{R}_g = \frac{1}{n_g} \sum_{l=1}^{n_g} R_{g,l} \).

The \( l \)th transformed Pearson residual of operator \( g \), denoted as \( \tilde{R}_{g,l} \), is calculated by \( \tilde{R}_{g,l} = R_{g,l} - \bar{R}_g \). With \( \tilde{R}_{g,l} \), the assumption of equal group mean is satisfied, while the original variance of each group of residuals remains unchanged.

**SC2. Identify operators with abnormal variance.** The ANOMRV chart is applied to identify operators with chart statistics beyond UDL or LDL. The statistic \( \tilde{U}_{g,l} = (\tilde{R}_{g,l} - \bar{\tilde{R}})^2 \), where \( \bar{\tilde{R}} \) is the sample mean of all \( \tilde{R}_{g,l} \) values, is calculated and then transformed to \( \tilde{U}_{g,l}^* = \frac{(\tilde{U}_{g,l} - \bar{\tilde{U}}_{g,l})}{S_{g,l}} + \bar{\tilde{U}}_{g,l} \), where \( \bar{\tilde{U}}_{g,l} \) and \( S_{g,l} \) are sample mean and sample standard deviation of \( \tilde{U}_{g,l} \) for a specific operator \( g \). \( \tilde{U}_{g,l}^* \) is used as input for ANOMRV chart. Operators with chart statistics exceeding the UDL or LDL are considered as having abnormal variances. Their IDs are grouped in a set, denoted as \( I_{O,V} \).

**SC3. Remove non-HOV residuals.** Before using ANOMR chart, the HOV assumption among residuals from different operators must hold. To validate this assumption, the residuals corresponding to operators in set \( I_{O,V} \) are removed.

**SC4. Identify operators with abnormal means.** The ANOMR chart is applied to the rest of the residuals to identify operators with chart statistics exceeding UDL or LDL. The IDs of the identified operator are grouped in a set, denoted as \( I_{O,M} \).

With the above four steps, the obtained set \( I_{O,V} \) and \( I_{O,M} \) contain operator
IDs with either excessively large or small mean and/or variance of NoD. Under-performing operators are identified as operators with chart statistics exceeding the UDLs of either chart.

2.2.4 Summary of the proposed methodology

The overall procedure of the proposed performance evaluation can be summarized by the following three steps (denoted by “S”) and illustrated in figure 2.2.

![Figure 2.2: Framework of proposed performance evaluation methodology](image)

**S1. Quantify the effects of other impacting factors.** The quantification is achieved by constructing the robust ZIP regression model with the other impacting factors as the predictor variables. The regression starts with the initial estimates of coefficients based on the modified FAST-LTS algorithm, which consists of index set formation (SR1) and modified random sampling (SR2).

**S2. Identify under-performing machines.** Based on the estimated regression coefficients, the estimates of population mean and variance of NoD for each machine are calculated. The distributions of these estimates are derived by delta method (SH1). Multiple hypothesis tests with Bonferroni adjustment are conducted (SH2).

As both machines and operators have effects on NoD data, various predictors related to these two, such as cumulative operation hour for machine and operator, age of operator, operator’s skill levels etc., may be added to the regression model. Then adding the machine ID associates the unexplained variation to each machine.
This quantifies the individual difference of each machine in affecting the NoD and enables the identification of under-performing ones. In contrast, operator ID is not included in the regression model, due to the following reasons.

1) Since the objective of this research is workforce evaluation, the trained model should allow the monitoring of the future performance of operators, especially new ones. However, the monitoring cannot be based on a new ID that is not equal to any used ID in model fitting.

2) The number of operators can be very large (such as 235 in the case study). Adding IDs as a predictor results in a model with hundreds of predictor variables. Previous study (Long, 1997) suggests that the logistic regression requires at least 10 observations for each predictor. Given that robust ZIP regression includes logistic part and certain operators may have very limited number (fewer than 10) of observations, the operator ID cannot be put in the model.

As both individual operator and machine have effects on NoD, the exclusion of operator ID relies on the assumption of independence between operator and machine. This assumption fits the practical condition that operators are randomly assigned to different machines. Under this condition, the potential effect of individual operator on the NoD is averaged out within each machine. Thus the comparison of machine performance will not be biased by certain operator’s effect. In many manufacturing systems (such as the one in the case study), operators work on different machines and each machine is operated by different operators in an almost random fashion. This is an analogy to the complete randomization in an experimental design. Therefore, the individual operator’s effect can be moved out of the regression model.

S3. Identify under-performing operators. The Pearson residuals from the robust ZIP regression are grouped by operator ID for performance evaluation. The residuals are first transformed to have equal group mean (SC1). Then ANOMRV chart is applied to identify operators with abnormal variances (SC2). The grouped residuals that violate HOV assumption are removed before the comparison with ANOMR chart (SC3). The ANOMR chart is then applied to the rest of the grouped residuals to identify operators with abnormal means (SC4). From the ANOMRV and
ANOMR charts, operators with chart statistics exceeding the UDLs—i.e. having mean and/or variance of grouped Pearson residuals significantly larger than the overall level—are identified as under-performing ones.

Though the structure of the proposed methodology resembles a regression control chart, it addresses a different problem. The methodology adopts the ANOMR and ANOMRV charts instead of Shewhart control chart, as the former is designed specifically for multiple comparisons. This is because ANOM type of charts (including ANOM/ANOMV and ANOMR/ANOMRV chart) consider the correlation between the samples in calculating the decision limits, as suggested by Nelson et al. (2005). This correlation, however, is ignored in the calculation of control limits of conventional Shewhart control chart. Besides, the problem of workforce evaluation requires that the residuals used for performance evaluation contain the physical meaning of the effect, i.e. the NoD level. In the proposed framework, Pearson residual is selected as it represents the level of NoD after quantifying the effects of other impacting factors. Deviance residual does not fit into this purpose as it represents each observation’s contribution to the model’s lack of fit measured in log-likelihood. Besides, the proposed methodology and regression control chart fit in different application contexts. The former collects the historical data and compares the residuals linked to each operator using one-time multiple comparisons. The latter collects real-time process quality statistics and continuously monitors the residual along time using single comparison. Based on the identification outcome, maintenance checks and enforced training programs can be scheduled for under-performing machines and operators, respectively. These actions reduce NoD level and improve product quality, which eventually increase productivity. This relationship can be demonstrated with the productivity metric of overall equipment effectiveness (OEE) (Ljungberg, 1998) and its quality-based component (OEE\textsubscript{Q}), where OEE \propto OEE\textsubscript{Q} and OEE\textsubscript{Q} is calculated as

$$OEE\textsubscript{Q} = \frac{\text{conforming product output (units)}}{\text{theoretical attainable product output (units)}}.$$  (2.8)
A reduction in NoD level will increase the number of conforming products manufactured. If the theoretical attainable product output is also fixed, it increases OEE and further increases OEE.

2.3 Demonstration and validation

A case study on data from a real-world production system and a simulation experiment are used to demonstrate the effectiveness of the proposed approach.

2.3.1 Demonstration

The proposed method is applied in a case study of the workforce performance evaluation with the NoD data collected from a real-world production system. Table 2.2 summarizes the dataset, which contains 4,472 observations collected from 12 machines operated by 235 operators. These observations represent the NoD of a specific type recorded during one operation shift, from the machines of the same model and the same functionality. In practice, the differences among machines may contribute to the variation of NoD. The features of machines, if available, should be added as predictor variables. Such features may include types of machines, functionalities, and age etc. The NoD count is assumed to be contributed by operators whose performances are affected by machines and three other impacting factors: $F_1$, $F_2$ and $F_3$.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value attribute</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td>Non-negative integer</td>
<td>NoD count</td>
</tr>
<tr>
<td>$mch$</td>
<td>Categorical</td>
<td>Identifier of machines</td>
</tr>
<tr>
<td>$F_1$</td>
<td>Categorical</td>
<td>A 4-level predictor variable</td>
</tr>
<tr>
<td>$F_2$</td>
<td>Categorical</td>
<td>A 2-level predictor variable</td>
</tr>
<tr>
<td>$F_3$</td>
<td>Continuous</td>
<td>Cumulative experience</td>
</tr>
</tbody>
</table>

Table 2.3 lists the sample distribution of NoD values in the dataset. The table contains all the unique values of NoD (row name ‘V’), its corresponding frequency (row name ‘F’), percentage (row name ‘P’) and cumulative percentage (row name ‘Pc’).
It is observed that zero takes up about 66% of the data and possible outliers exist in the data, such as the observation with value 30. Due to zero inflation and outliers, robust ZIP regression method is fitted to the data. The same predictors of mch, F_1, F_2 and F_3 are included for both Poisson and logistic regression. According to domain experts, no interactions are assumed among the predictor variables. The final model, which contains only significant predictors after backward elimination, is listed in table 2.4. In the table, F_{1,2} represents the coefficient of F_1 at level 2. The regression selects the benchmark values with predictors mch, F_1, F_2 and F_3 at level of mch.1, F_{1,1}, F_{2,1} and 0 (representing newly recruited operators), respectively.

Table 2.3: Frequency distribution of observations in the NoD dataset

<table>
<thead>
<tr>
<th>V</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>2949</td>
<td>522</td>
<td>445</td>
<td>306</td>
<td>138</td>
<td>54</td>
<td>20</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>P</td>
<td>0.6594</td>
<td>0.1167</td>
<td>0.0995</td>
<td>0.0684</td>
<td>0.0309</td>
<td>0.0121</td>
<td>0.0045</td>
<td>0.0013</td>
<td>0.0011</td>
</tr>
<tr>
<td>CP</td>
<td>0.6594</td>
<td>0.7762</td>
<td>0.8757</td>
<td>0.9441</td>
<td>0.9750</td>
<td>0.9870</td>
<td>0.9915</td>
<td>0.9928</td>
<td>0.9940</td>
</tr>
<tr>
<td>V</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>16</td>
<td>17</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>F</td>
<td>8</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>P</td>
<td>0.0018</td>
<td>0.0011</td>
<td>0.0004</td>
<td>0.0007</td>
<td>0.0007</td>
<td>0.0002</td>
<td>0.0002</td>
<td>0.0004</td>
<td>0.0002</td>
</tr>
<tr>
<td>CP</td>
<td>0.9958</td>
<td>0.9969</td>
<td>0.9973</td>
<td>0.9980</td>
<td>0.9987</td>
<td>0.9989</td>
<td>0.9991</td>
<td>0.9996</td>
<td>0.9998</td>
</tr>
<tr>
<td>V</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>0.0002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

With regression coefficients from table 2.4, the estimates of population mean and population variance for different machines are calculated using Eq.(2.3) and Eq.(2.4). Then variances of these estimates are obtained from Eq.(2.7). Multiple hypothesis tests with Bonferroni adjustment are conducted to identify under-performing machines. The outcomes are listed in table 2.5. The two groups of hypothesis tests used in identification are $H_0 : m_{[j]} = 1.305$ vs $H_1 : m_{[j]} < 1.305, [j] \neq 11$ and $H_0 : m_{[j]} = 2.872$ vs $H_1 : m_{[j]} < 2.872, [j] \neq 9$.

In the case study, the benchmark value is selected as the maximum estimate among different machines due to confidentiality of benchmark value based on domain knowledge. In comparing the estimates of population mean of NoD, the maximum
value 1.305 is selected as the benchmark and p-values of these tests are shown in the last column of the left part of table 2.5. To ensure the family-wise error rate is at the level of 0.05 in testing, Bonferroni adjustment is applied. Tests show that machines with ID of 5, and 7 to 11 (highlighted in bold letter) are associated with higher mean values of NoD. For the estimates of population variance, similar steps follow and the maximum value 2.872 is adopted as benchmark. Tests results show that machines with ID of 7 to 12 are associated with higher variance. It is noted that half of the machines are identified as under-performing. This can be due to the use of conservative Bonferroni adjustment. Further discussion with domain experts will be necessary for interpretation of the results.

<table>
<thead>
<tr>
<th>Predictors</th>
<th>Estimate</th>
<th>Stdev</th>
<th>p-Value</th>
<th>Predictors</th>
<th>Estimate</th>
<th>Stdev</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Icpt</td>
<td>1.000</td>
<td>0.147</td>
<td>0.000</td>
<td>Icpt</td>
<td>0.738</td>
<td>0.097</td>
<td>0.000</td>
</tr>
<tr>
<td>mch.2</td>
<td>-0.820</td>
<td>0.207</td>
<td>0.000</td>
<td>mch.2</td>
<td>-0.095</td>
<td>0.121</td>
<td>0.432</td>
</tr>
<tr>
<td>mch.3</td>
<td>-0.373</td>
<td>0.201</td>
<td>0.063</td>
<td>mch.3</td>
<td>-0.147</td>
<td>0.123</td>
<td>0.235</td>
</tr>
<tr>
<td>mch.4</td>
<td>0.771</td>
<td>0.247</td>
<td>0.002</td>
<td>mch.4</td>
<td>-0.303</td>
<td>0.173</td>
<td>0.080</td>
</tr>
<tr>
<td>mch.5</td>
<td>-1.140</td>
<td>0.212</td>
<td>0.000</td>
<td>mch.5</td>
<td>-0.076</td>
<td>0.116</td>
<td>0.512</td>
</tr>
<tr>
<td>mch.6</td>
<td>0.113</td>
<td>0.222</td>
<td>0.611</td>
<td>mch.6</td>
<td>-0.266</td>
<td>0.142</td>
<td>0.060</td>
</tr>
<tr>
<td>mch.7</td>
<td>-1.523</td>
<td>0.285</td>
<td>0.000</td>
<td>mch.7</td>
<td>-0.129</td>
<td>0.138</td>
<td>0.351</td>
</tr>
<tr>
<td>mch.8</td>
<td>-0.881</td>
<td>0.186</td>
<td>0.000</td>
<td>mch.8</td>
<td>0.109</td>
<td>0.107</td>
<td>0.306</td>
</tr>
<tr>
<td>mch.9</td>
<td>-0.472</td>
<td>0.179</td>
<td>0.008</td>
<td>mch.9</td>
<td>0.292</td>
<td>0.107</td>
<td>0.006</td>
</tr>
<tr>
<td>mch.10</td>
<td>-1.455</td>
<td>0.194</td>
<td>0.000</td>
<td>mch.10</td>
<td>-0.053</td>
<td>0.105</td>
<td>0.610</td>
</tr>
<tr>
<td>mch.11</td>
<td>-1.183</td>
<td>0.187</td>
<td>0.000</td>
<td>mch.11</td>
<td>0.134</td>
<td>0.104</td>
<td>0.198</td>
</tr>
<tr>
<td>mch.12</td>
<td>0.030</td>
<td>0.183</td>
<td>0.868</td>
<td>mch.12</td>
<td>0.382</td>
<td>0.109</td>
<td>0.000</td>
</tr>
</tbody>
</table>

The goodness-of-fit statistics are plotted before the comparison using ANOMR and ANOMRV charts. Figure 2.3 shows the chi-square goodness-of-fit statistics plotted as small circles. It is noted that most plotted statistics generally fall inside the envelope whereas the rightmost statistic falls outside and far away. As aforementioned, the envelope is conceptually similar to a confidence interval and gives
an estimated range of the plotted statistics if they are calculated from observations without outliers. Since the statistics inside the envelope fall approximately along a line, the plot confirms the model adequacy (i.e. the model is correctly specified). In addition, it indicates that the rightmost statistic is calculated from observations that include outliers, which are very large NoD values (e.g. 30) in the data.

<table>
<thead>
<tr>
<th>ID</th>
<th>Estimates</th>
<th>Stdev</th>
<th>p-Value</th>
<th>ID</th>
<th>Estimates</th>
<th>Stdev</th>
<th>p-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mch.1</td>
<td>0.563</td>
<td>0.081</td>
<td>0.000</td>
<td>mch.1</td>
<td>1.423</td>
<td>0.251</td>
<td>0.000</td>
</tr>
<tr>
<td>mch.2</td>
<td>0.866</td>
<td>0.148</td>
<td>0.001</td>
<td>mch.2</td>
<td>1.763</td>
<td>0.360</td>
<td>0.001</td>
</tr>
<tr>
<td>mch.3</td>
<td>0.629</td>
<td>0.117</td>
<td>0.000</td>
<td>mch.3</td>
<td>1.370</td>
<td>0.306</td>
<td>0.000</td>
</tr>
<tr>
<td>mch.4</td>
<td>0.225</td>
<td>0.062</td>
<td>0.000</td>
<td>mch.4</td>
<td>0.522</td>
<td>0.176</td>
<td>0.000</td>
</tr>
<tr>
<td>mch.5</td>
<td>1.037</td>
<td>0.165</td>
<td>0.052</td>
<td>mch.5</td>
<td>1.972</td>
<td>0.374</td>
<td>0.008</td>
</tr>
<tr>
<td>mch.6</td>
<td>0.397</td>
<td>0.088</td>
<td>0.000</td>
<td>mch.6</td>
<td>0.875</td>
<td>0.233</td>
<td>0.000</td>
</tr>
<tr>
<td>mch.7</td>
<td>1.155</td>
<td>0.207</td>
<td>0.234</td>
<td>mch.7</td>
<td>1.946</td>
<td>0.400</td>
<td>0.010</td>
</tr>
<tr>
<td>mch.8</td>
<td>1.097</td>
<td>0.165</td>
<td>0.104</td>
<td>mch.8</td>
<td>2.453</td>
<td>0.451</td>
<td>0.176</td>
</tr>
<tr>
<td>mch.9</td>
<td>1.039</td>
<td>0.168</td>
<td>0.057</td>
<td>mch.9</td>
<td>2.872</td>
<td>0.581</td>
<td>-</td>
</tr>
<tr>
<td>mch.10</td>
<td>1.213</td>
<td>0.166</td>
<td>0.290</td>
<td>mch.10</td>
<td>2.148</td>
<td>0.354</td>
<td>0.020</td>
</tr>
<tr>
<td>mch.11</td>
<td>1.305</td>
<td>0.184</td>
<td>-</td>
<td>mch.11</td>
<td>2.723</td>
<td>0.475</td>
<td>0.377</td>
</tr>
<tr>
<td>mch.12</td>
<td>0.806</td>
<td>0.144</td>
<td>0.000</td>
<td>mch.12</td>
<td>2.628</td>
<td>0.587</td>
<td>0.339</td>
</tr>
</tbody>
</table>

The identification of under-performing operators is based on Pearson residual. As shown in figure 2.4, all the chart statistics beyond the decision limits are highlighted with square symbol. As the under-performing operators are defined as operators with excessively large mean or variance in terms of NoD, operators with chart statistics exceeding UDL in ANOMR or ANOMRV charts are identified as under-performing. Note that the charts also identify operator with small means of NoD. Similar to the control chart case, it is necessary to re-examine the data against measurement error before concluding that this operator has indeed small means.
To illustrate the benefits on the increase of productivity from identification outcome, a comparison is conducted between current OEE_Q and potential OEE_Q after improving the performance of under-performing machines and operators. Based on domain knowledge, only one product is manufactured on each operation shift and the conforming products are defined as having NoD less than one. Under this condition, OEE_Q is calculated as

\[
OEE_Q = \frac{\text{product output (units) with NoD less than one}}{\text{theoretical attainable product output (units)}}.
\]  

Based on given data, OEE_Q is 77.6%. Suppose that five under-performing operators identified in ANOMRV chart and the machines with maximum predicted NoD mean/variance (i.e. mch.11 and mch.9) have their performances improved so that all the products manufactured with them are conforming, then potential improved OEE_Q becomes 84.1%. This improvement on OEE_Q increases OEE by 8%.
Figure 2.4: Outcome of ANOMRV and ANOMR chart
2.3.2 Validation

As the information regarding the performance of the machines and operators is kept confidential, no evidence is available for the validation of the identification outcome. Therefore, a simulation experiment is conducted to demonstrate the effectiveness of the proposed approach. The experiment simulates a production system with 10 machines and 10 operators. One machine is simulated as under-performing with higher mean (2) and higher variance (6) of NoD, compared to its normal-performing counterparts with lower means (1) and lower variances (3). The experiment also assumes two under-performing operators. One of them has higher mean (1.5) of NoD and the other has higher variance (6), compared to their normal-performing counterparts with lower means (1) and lower variances (3). The assigned means and variances for different machines and operators are summarized in table 2.6.

<table>
<thead>
<tr>
<th>Table 2.6: Assigned mean and variance of machines and operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Variance</td>
</tr>
</tbody>
</table>

Based on these assigned values, the corresponding true coefficients for each machine and each operator are calculated. Three other impacting factors, $F_1$, $F_2$ and $F_3$, are simulated. The corresponding true coefficients—i.e. elements of vector $\beta$ and $\gamma$ in Eq.(2.1), are summarized in table 2.7. In this table, the notation of different levels of factor $F_1$, $F_2$ are defined in the same way as in the case study.

<table>
<thead>
<tr>
<th>Table 2.7: True coefficients of other impacting factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_{1,1}$</td>
</tr>
<tr>
<td>Element in $\beta$</td>
</tr>
<tr>
<td>Element in $\gamma$</td>
</tr>
</tbody>
</table>

NoD data are simulated based on the given true coefficients with a complete randomized design, i.e. operators are randomly assigned to different machines. Five (5) replicates are generated for each combination of machine, operator, $F_1$ and $F_2$. 
The values of $F_3$ are generated from $N(2000, 400)$ with parameter setting similar to that of the $F_3$ observed in case study. A total of 4,000 observations are simulated. As aforementioned, both normal- and under-performing operators may generate outliers. This phenomenon has been observed in the real-world NoD data used in the case study. To reflect this in the simulated data, the highest 1% of simulated observations are selected and added with a value generated from a discrete uniform distribution of $U(10, 20)$ to represent the outliers. Then same robust ZIP model as in case study is fitted to the simulated data. The performances of the standard FAST-LTS and the modified FAST-LTS (i.e. the FAST-LTS with modified sampling scheme) are compared regarding the number of singular matrix $X_J$ obtained in the estimation process. The average number of singular $X_J$ from one thousand (1,000) replications using standard method is 493.5, while no singular $X_J$ is generated with the modified sampling. This demonstrates the effectiveness of the proposed modified sampling technique in solving the singularity problem of the standard FAST-LTS.

The same goodness of fit checking method is used in simulation. Figure 2.5 shows the plot where three statistics fall outside and far away from the envelope illustrated by dashed lines. This indicates that three statistics are calculated from the observations (i.e. highest 1% of the data) containing outliers. As the other statistics inside the envelope again fall approximately in a line, this plot confirms the model adequacy and ANOMR/ANOMRV chart is set up based on the Pearson residual. To evaluate the performance, the false positive rate ($FP$) and false negative rate ($FN$) are adopted and defined as:

\[
FP = \frac{\text{number of normal machine/operators misidentified as underperforming}}{\text{number of normal machines/operators}}, \tag{2.10}
\]

\[
FN = \frac{\text{number of underperforming machine/operators misidentified as normal}}{\text{number of underperforming machines/operators}}. \tag{2.11}
\]
One thousand (1,000) replications of experiments are conducted. The mean values of $FP$ and $FN$ for identification of under-performing machine and operators are summarized in table 2.8. The table shows small error rates except the FP rate for detecting operators with higher variance. As the FP rate is not much higher than the type I error level (i.e. 0.05) used in ANOMR/ANOMRNV chart, the results are acceptable and they prove the effectiveness of the method.
2.4 Conclusion

Integration of manufacturing system design and quality improvement has received increasing attention in recent years. Performance evaluation, which makes up an important part of the workforce management, can be joined with quality improvement for the competitiveness enhancement. Operators with unsatisfactory performance in delivering quality products need to be identified for enforced training programs. In this chapter, a methodology incorporating regression model and multiple comparison methods is proposed as a solution to the identification of under-performing machines and under-performing operators. Considering the outliers and zero-inflation in NoD data, robust ZIP regression is employed to quantify the effects of other impacting factors on the operator performance. A modified sampling step is proposed for improving the effectiveness of FAST-LTS to provide initial coefficient estimates for a ZIP regression model containing categorical predictor variables. Under-performing machines are identified through multiple hypothesis tests that compare the estimates of population mean and variance of individual machine with benchmark values. The ANOMR and ANOMRV charts are employed for comparing the mean and variance of Pearson residuals grouped by operators for identifying under-performing ones. Once these underperformers are identified, corrective actions can be taken to improve product quality and it increases productivity measured by OEE.

One limitation of the methodology proposed in this chapter is the choice of parameter values. Empirical values based on experimentation or existing studies are adopted. For example, the size of set $H_I$ in FAST-LTS uses the empirical value. However, better parameter values may exist. Similarly for the tuning parameter $c$ in robust ZIP regression, the case study and simulation follow the setting in Hall and Shen (2010) and choose $c = 0.01$. A data-dependent selection method of these parameters can be investigated. More efforts will be put on developing systematic methods of deriving optimal parameters in the future research. In addition, the derivation of distributions for estimated mean and variance of NoD level relies on asymptotic theory. Therefore the proposed methodology does not fit the applications
with small sample size. Further investigation will be also conducted on this topic.
CHAPTER 3

DETECTING ASYNCHRONOUS PERIODIC PATTERNS OF INTERVALS IN TEMPORAL SEQUENCE DATA

3.1 Problem statement

Temporal sequence data are often collected from industrial and service operations to provide valuable information for management decision making. For instance, “Ready” alarms are triggered and logged by sensors equipped on mining haul trucks to indicate that the truck is ready for loading operation. As aforementioned in Chapter 1, the alarm occurrence data are collected through on-board sensors and reflect the truck’s health status. By detecting the temporal patterns embedded in, such alarm log data can be incorporated into the equipment prognostics and health management (PHM) (Pecht (2009)) to evaluate equipment performance and predict potential critical events, such as machine failures. Condition-based system maintenance can then be scheduled according to the pattern-based prediction to reduce the maintenance cost and improve production safety (Li et al. (2007b)).

As one type of temporal sequence data, event logs such as “Ready” alarms are often stored as an ordered sequence of event identifiers together with their corresponding occurrence timestamps. Given a specific event type, the temporal sequence can be represented as \( \{t_1, t_2, ..., t_n\} \), where \( t_i \in R^+, i = 1, 2, ..., n, \ n \in N^+ \), \( (R^+ \) and \( N^+ \) are sets of positive real and integer numbers, respectively), denotes the timestamp for the \( i \)th occurrence of an event. As illustrated in Figure 3.1 (a), an interval sequence can be obtained from the temporal sequence by calculating the time intervals between occurrences of 'Ready'.

**Definition 1 (Interval sequence)** Given a temporal sequence \( \{t_1, t_2, ..., t_n\} \) for a specific event type, the interval sequence is \( \tau = (\tau_1, \tau_2, ..., \tau_{n-1}) \), where \( \tau_k = t_{k+1} - t_k, \ k = 1, 2, ..., n - 1. \)
The interval sequence in Figure 3.1 (a) shows two patterns. Pattern I consists of “Ready” alarms that repeat every 9 minutes and pattern II consists of repetitive pairs of different but consecutive intervals of 5 minutes, 5 minutes and 9 minutes. Such interval sequence segments with repetitive interval values are defined as “periodic patterns” in this chapter. In real-world operations, however, event occurrence is subject to random disturbance. It is extremely rare, if not impossible, to observe that events repeat exactly after a certain period of time. Thus, the interval sequences collected rarely show the ideal patterns illustrated in Figure 3.1 (a). Instead, the sequence may contain time intervals that are only approximately equal to each other. If such intervals within a proximity of a certain target value are treated as equal, periodic patterns can be defined based on these approximately equal intervals.

**Definition 2 (Periodic pattern)** In an interval sequence, a segment of length $L$ is a repetition of $v$ patterns with period $p$ if, for any two elements $\tau_i$ and $\tau_{i+vp}$ in this segment, $\tau_i, \tau_{i+vp} \in [T_j - \delta_j, T_j + \delta_j]$, for $v = 1, 2, ..., \text{ and } i, i + vp \in [1, L]$. $\delta_j$ is the tolerance that defines the maximum allowable disturbance around the $j$th target interval, $T_j$, for $j = 1, ..., K$, and $K$ is the number of potential target values. When $\delta_j = 0$ (i.e., $\tau_i = \tau_{i+vp}$), a *synchronous periodic pattern* will be formed. When $\delta_j > 0$, an *asynchronous periodic pattern* (APP) will be formed.

For example, pattern I and pattern II in Figure 3.1 (a) are two synchronous periodic patterns with period 1 and 3, respectively. In Figure 3.1 (b), some intervals are 7˜11 min long. If $T_1 = 9$ and $\delta_1 = 2$, an APP with period 1 can be found. Another group of intervals are 5˜6 min long, which are approximately equal to a target interval of 5 min. Thus further using $T_2 = 5$ and $\delta_2 = 1$ will produce an APP with period 3.

It is crucial yet challenging to detect the APPs from temporal sequence data with approximately equal time intervals between event occurrences, as shown in Figure 3.1 (b). Research (Ma and Hellerstein (2001), Lee et al. (2008)) has been conducted to address the random disturbance and detect the consequent APPs of interval sequences shown in Figure 3.1 (b). One common limitation of the existing
Figure 3.1: Transformation from temporal sequences data to interval sequences with periodic patterns (Unit: min): (a) sequence with exact intervals and synchronous periodicity and (b) interval sequence with disturbance and asynchronous periodicity

### Figure 3.1a

<table>
<thead>
<tr>
<th>Timestamp of 'Ready'</th>
<th>Interval Sequence</th>
<th>Expected Interval + Disturbance</th>
<th>Symbol Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/12 12:05AM</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1/1/12 12:10AM</td>
<td>5 mins</td>
<td>5+0 mins</td>
<td>A</td>
</tr>
<tr>
<td>1/1/12 12:15AM</td>
<td>5 mins</td>
<td>5+0 mins</td>
<td>A</td>
</tr>
<tr>
<td>1/1/12 12:20AM</td>
<td>9 mins</td>
<td>9+0 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/1/12 12:25AM</td>
<td>5 mins</td>
<td>5+0 mins</td>
<td>A</td>
</tr>
<tr>
<td>1/1/12 12:30AM</td>
<td>5 mins</td>
<td>5+0 mins</td>
<td>A</td>
</tr>
<tr>
<td>1/1/12 12:35AM</td>
<td>9 mins</td>
<td>9+0 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/2/12 1:05AM</td>
<td>50 mins</td>
<td>50+0 mins</td>
<td>C</td>
</tr>
<tr>
<td>1/2/12 1:15AM</td>
<td>9 mins</td>
<td>9+0 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/2/12 1:25AM</td>
<td>9 mins</td>
<td>9+0 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/2/12 1:35AM</td>
<td>9 mins</td>
<td>9+0 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/2/12 1:45AM</td>
<td>9 mins</td>
<td>9+0 mins</td>
<td>B</td>
</tr>
</tbody>
</table>

### Figure 3.1b

<table>
<thead>
<tr>
<th>Timestamp of 'Ready'</th>
<th>Interval Sequence</th>
<th>Expected Interval + Disturbance</th>
<th>Symbol Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1/12 12:05AM</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1/1/12 12:10AM</td>
<td>5 mins</td>
<td>5+0 mins</td>
<td>A</td>
</tr>
<tr>
<td>1/1/12 12:15AM</td>
<td>6 mins</td>
<td>5+1 mins</td>
<td>A</td>
</tr>
<tr>
<td>1/1/12 12:20AM</td>
<td>7 mins</td>
<td>9+2 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/1/12 12:25AM</td>
<td>5 mins</td>
<td>5+0 mins</td>
<td>A</td>
</tr>
<tr>
<td>1/1/12 12:30AM</td>
<td>6 mins</td>
<td>5+1 mins</td>
<td>A</td>
</tr>
<tr>
<td>1/1/12 12:35AM</td>
<td>7 mins</td>
<td>9+2 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/2/12 1:05AM</td>
<td>55 mins</td>
<td>50+5 mins</td>
<td>C</td>
</tr>
<tr>
<td>1/2/12 1:15AM</td>
<td>7 mins</td>
<td>9+2 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/2/12 1:25AM</td>
<td>9 mins</td>
<td>9+0 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/2/12 1:35AM</td>
<td>7 mins</td>
<td>9-2 mins</td>
<td>B</td>
</tr>
<tr>
<td>1/2/12 1:45AM</td>
<td>10 mins</td>
<td>9-1 mins</td>
<td>B</td>
</tr>
</tbody>
</table>
method, however, is the requirement on the specification of tolerance values, i.e., \( \delta_j \)'s. Choosing proper tolerance values is often difficult and the chosen values are often unjustifiable. This is because that (i) there may be insufficient prior knowledge on the magnitude of disturbance; (ii) when time intervals are with different target values, i.e., \( T_j \)'s, as the 5-min-intervals and the 9-min-intervals shown in Figure 3.1 (b), different tolerance values, i.e., \( \delta_j \)'s, may be needed and their magnitudes are difficult to give with a priori justification. Thus, it is desirable to objectively specify the tolerance values based on the observed data. After objectively setting \( \delta_j \)'s, the intervals in a sequence can be grouped with respect to different target interval \( T_j \)'s and labeled with symbolic values to form a symbol sequence.

**Definition 3 (Symbol sequence)** An interval sequence, \( \tau = (\tau_1, \tau_2, ..., \tau_{n-1}) \), can be converted to a symbol sequence, \( s = (s_1, s_2, ..., s_{n-1}) \), based on a converting
function, $C(\tau_i)$, as

$$s_i = C(\tau_i) = S_j, \text{ if } \tau_i \in [T_j - \delta_j, T_j + \delta_j], \quad (3.1)$$

for $i = 1, \ldots, n - 1$, and $j = 1, \ldots, K$,

where $S_j$ is the symbolic value (e.g. letters 'A', 'B',...) corresponding to target value $T_j$ and the tolerance $\delta_j$.

Given $T_j$'s and $\delta_j$'s, the disturbance associated with interval values is neglected and thus, the periodic patterns embedded in an interval sequence can be reserved in its corresponding symbol sequence. For instance, two symbol sequence segments with periodic patterns, 'BBBBB' and 'AABAAB', can be obtained from the interval sequence in Figure 3.1 (a), with target values $T_1 = 9$ and $T_2 = 5$, and tolerances $\delta_1 = \delta_2 = 0$. These two segments are with period 1 and 3, respectively, the same as that of the interval sequence segments. The same symbol sequence segments can also be achieved with the same target values and tolerances $\delta_1 = 2$, $\delta_2 = 1$, with the same periodic patterns in the interval sequence as shown in Figure 3.1 (b).

The data-driven interval value grouping and labeling is conceptually equivalent to statistical clustering, which categorizes unlabeled intervals in a dataset into homogeneous groups based on certain proximity measures. This data-driven tolerance specification is to evaluate the proximity of the $n - 1$ time intervals, form $K$ groups of intervals bounded by tolerance $\delta_j$'s, and represent their target value, $T_j$'s, with $K$ representative values. The within-group difference among time intervals from the same group is significantly smaller than the between-group difference among time intervals from different groups. The goal of clustering in data analysis is to identify the hidden pattern or structure of data. Among various existing clustering methods, hierarchical clustering has been widely used as it discovers the hierarchical structure of data corresponding to different numbers of clusters. Specifically, hierarchical clustering methods Murtagh (1983) construct a dendrogram, which is a tree diagram, in representing potential data structure. The methods are further classified into two types, namely agglomerative and divisive approaches. Agglomerative approach
merges clusters in a bottom-up fashion while divisive approach that splits clusters proceeds in an opposite way. Cluster merging and splitting are determined according to proximity measures, such as Ward’s distance, to maximize the within-cluster similarity and minimize the between-cluster similarity Olson (1995). As a result, intervals that are approximately equal to each other are clustered together.

In this chapter, statistical clustering is adopted to detect APPs in interval sequences. The clustering of interval values close to cluster boundaries is often uncertain and mis-clustered intervals may change the detection results and the periodicity interpretation. Clustering configurations with high pattern interpretability are the ones leading to more and longer periodic patterns. For example, two potential clustering configurations may be formed from an interval sequence (5,6,7,5,6,7,9,8,11,7,45,9,8,7,9,55,7,9,7,10,7) shown in Figure 3.2 (a) and (b). The clustering configurations are illustrated with intervals and labels, 'A', 'B' and 'C' inside boxes. The periodic patterns are detected based on the two corresponding symbol sequences, which differ only in the clustering (labeling) of the interval 7, as shown in Figure 3.2. In configuration I, the interval 7 is on the boundary of cluster \{5, 6, 7\} and in configuration II, it is clustered with intervals 8, 9, 10, and 11. A comparison of the two symbol sequences shows that configuration I results in clusters with more homogeneous interval values but configuration II results in more and longer periodic patterns. Specifically, in configuration II, all the symbols belong to certain periodic pattern, i.e. 'AABAAB', 'BBBBCBBBBB' or 'BBBBBB'. In contrast, configuration I results in four short periodic patterns ('AAAAA', 'BBB', 'BB' and 'ABAB', with some symbols not belonging to any pattern). Some segments, such as BB, are generally too short to interpret. The example demonstrates that since interval values on the cluster boundary are less certain to belong to one cluster, it is desirable to address this uncertainty by considering the benefits of pattern interpretability. The optimal clustering configuration needs to balance the trade-off between data similarity and pattern interpretability.

The clustering of interval sequence transforms APPs into periodic patterns in a symbol sequence. Thus, the detection of APPs can be achieved by analyzing
the symbol sequence. Existing methods on periodic pattern detection in symbol sequence often mapped each symbol in the sequence into a numeric value. These mapping methods provided low computational efficiency and limited flexibility or capability to handle larger number of symbol types. To address the limitation of existing methods, this chapter proposes a convolution-based template matching approach that allows flexible mapping and efficient pattern detection. The flexible mapping refers to the fact that the detection result will not change as long as symbols are uniquely mapped. The use of convolution ensures high computational efficiency, which is crucial for applications in industrial practice where data size can be extremely large.

In this chapter, a statistical clustering-based periodic pattern detection method is proposed to address the following problems:

(i) The detection of APPs from temporal sequence data.

(ii) The transformation from interval sequences to symbol sequences, considering both disturbances and interpretability.

(iii) The efficient detection of periodic patterns in symbol sequence with flexible mapping.

The remainder of this chapter is organized as follows. The related work is introduced in Section 3.2. Section 3.3 provides the details of the proposed approach. Simulation experiment and case study on real data are presented in Section 3.4 to demonstrate the effectiveness of the proposed approach. Section 3.5 concludes the chapter with topics for future work.

3.2 Related work

Limited research has been conducted to detect periodic patterns directly from interval sequences. Ma and Hellerstein Ma and Hellerstein (2001) introduced the problem of asynchrony in time of occurrence of event when analyzing periodic patterns with unknown period. The approach of setting tolerance values, which was also used to address asynchrony issues in symbol sequences (Yang et al. (2003),
Huang and Chang (2005), was adopted. To discover period patterns in an interval sequence, a chi-squared hypothesis test was developed to check whether each inter-arrival time (i.e. time intervals between consecutive occurrences) between events of the same type is a candidate period. The method by Ma and Hellerstein Ma and Hellerstein (2001) extended previous approach Han et al. (1999) to detect periodic patterns with consideration of asynchrony. Since only inter-arrival time was considered in selecting candidate period, this method failed to capture the periodic pattern shown in Figure 3.1 (a), where the period of the pattern is 19 minutes not equal to inter-arrival time of 5 or 9 minutes. Lee et al. Lee et al. (2008) defined fuzzy periodicities with fuzzy periodic calendar aiming to handle pattern detection under disturbance. The fuzzy periodic calendar adopted a transformation function for fuzzy set to quantify 'closeness' of intervals. The format of this transformation function, however, requires subjective specification. In addition, it only detected patterns consisting of repeating occurrence at every time stamp. The methodology proposed in this chapter transforms the interval sequence to a symbol sequence first, and then detects APPs from the symbol sequence. The transformation is implemented by adopting an agglomerative hierarchical clustering approach. Various clustering methods with different proximity measures have been proposed to evaluate data similarity, which is quantified as the distance between elements in different clusters. Commonly-used proximity measures include shortest distance (minimum distance) Sibson (1973), farthest neighbor (maximum distance) King (1967), group average (average distance) Sokal (1958), and Ward’s distance Ward Jr (1963), etc. The Ward’s distance is defined as the increase of total within-cluster variance after merging two clusters. Previous study Blashfield (1976) has shown that Ward’s distance performs significantly better than the other three measures in recovering original data structure. Using Ward’s distance, agglomerative hierarchical clustering on an interval sequence begins with a cluster configuration where each interval value forms its own cluster. The clustering procedure recursively chooses two clusters to merge until all interval values form a single cluster. In each recursion, the pair of merged clusters are chosen to have the minimum increase of total within-cluster
Table 3.1: Comparison of existing and proposed approach

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Limitations of existing approaches</th>
<th>Contributions of proposed approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determination of tolerance</td>
<td>Subjective selection of tolerance Ma and Hellerstein (2001), Lee et al. (2008)</td>
<td>Objective selection of tolerance</td>
</tr>
<tr>
<td>Interpretation of patterns</td>
<td>Clustering based on data similarity Jung et al. (2003)</td>
<td>Balancing trade-off between data similarity and pattern interpretability</td>
</tr>
<tr>
<td>Efficiency of pattern detection</td>
<td>Outcome sensitive to mapping Wang and Johnson (2002), Buchner and Janjarasjitt (2003); limited number of symbol types Brodzik (2007); high computational complexity Elfeky et al. (2005)</td>
<td>Flexible mapping, unlimited number of symbol types, low computational complexity</td>
</tr>
</tbody>
</table>

variance. The clustering of interval sequence will generate a symbol sequence, from which periodic patterns will be detected.

To detect periodic patterns in symbol sequence, various approaches have been proposed. Some of these approaches adopted signal processing techniques. A wavelet-based approach Wang and Johnson (2002) was developed to detect periodic patterns but it only detected patterns with period length of power of 2, i.e. 2, 4, 8, etc. Buchner and Janjarasjitt Buchner and Janjarasjitt (2003) developed a periodic pattern detection and visualization algorithm using short-time periodicity transformation, which mapped symbols into complex numbers. One limitation for this approach is that the mapping always fails to detect certain periodic pattern. This limitation was addressed in Brodzik (2007), which used a set of quaternions, i.e. extension of complex numbers to four dimensional hyper-complex numbers, to represent symbols. However, this approach was only fitted for sequences with four
different types of symbols. A convolution-based algorithm Elfeky et al. (2005) was proposed to detect periodic patterns with competitive time performance and accuracy. However, it assumed the entire sequence had only one pattern. For these approaches based on signal processing technique, one common limitation comes from existing mapping methods from symbols to numerical values. Existing mapping methods are either not applicable to cases when the number of types of symbols exceeds four, or leading to much longer sequence than the original one for analysis. Though string processing technique Cheung et al. (2005) that does not require mapping has been adopted in pattern detection Rasheed et al. (2011), the computational complexity is significantly high.

As compared to existing approaches, the proposed convolution-based template matching method possesses the unique advantages with the following features: (i) The determination of tolerance values for APPs is achieved by a hierarchical clustering of interval sequences. This avoids the subjective specification of tolerance values used in existing approaches. (ii) The proposed hierarchical clustering algorithm achieves balance between data similarity and pattern interpretability. (iii) An efficient and flexible convolution-based template matching algorithm is proposed. These features are summarized in Table 3.1.

3.3 Proposed methodology

This chapter proposes a two-step methodology to detect APPs from interval sequences. In the first step, an interval sequence shown in Figure 3.1 is clustered with a modified hierarchical clustering approach, which gives a symbol sequence that reserves the interpretable periodic patterns embedded in the interval sequence. The reserved patterns are detected in the second step, with the proposed convolution based algorithm. These two steps are presented in subsections 3.3.1 and subsection 3.3.2, respectively.
3.3.1 Hierarchical clustering for asynchronous periodicity

This research systematically and objectively sets the tolerances on similar interval values through a new hierarchical clustering approach. As aforementioned, the conventional clustering based on data similarity alone has the limitation since it is less certain on clustering the interval values that are close to the cluster boundaries. Such uncertainty can be addressed by balancing data similarity and pattern interpretability in clustering. Although it may lead to trivial reduction of data similarity within clusters when reassigning some interval values to a neighboring cluster, the resulting symbol sequence can contain much more interpretable patterns.

To implement the balancing between data similarity and pattern interpretability, a modified clustering objective function is proposed. Given an interval sequence \( \tau = (\tau_1, \tau_2, ..., \tau_{n-1}) \), the proposed objective function is formulated as the maximization of a weighted summation of four terms:

\[
\max_{K,X} J = w_1 (1 - E_1) + w_2 (1 - E_2) + w_3 I_P + w_4 U_P, \tag{3.2}
\]

where the \( K \) and \( X \) are decision variables. \( K \) represents the number of target intervals, which is equivalent to the number of clusters. \( X \) is a set that contains indicators \( x_{ij}, i = 1, 2, ..., n-1, j = 1, 2, ..., K \). \( x_{ij} \) is defined for each interval \( \tau_i \) as

\[
x_{ij} = \begin{cases} 
1, & \text{if } \tau_i \in \text{cluster } j \\
0, & \text{otherwise}
\end{cases}, \tag{3.3}
\]

where the condition "\( \tau_i \in \text{cluster } j \)” is equivalent to the condition "\( \tau_i \in [T_j - \delta_j, T_j + \delta_j] \)” used in Definition 3, with \( T_j \) as cluster center and \( \delta_j \) implicitly specified as cluster boundary. \( w_1, w_2, w_3, w_4 \in [0, 1] \) are the weight coefficients, with constraint of \( w_1 + w_2 + w_3 + w_4 = 1 \).

\( E_1 \) and \( E_2 \) evaluate the clustering configuration from a perspective of data sim-
ilarity. These two indices are defined as

\[ E_1 = \frac{E_{\text{intra}}}{SS_T}, \quad \text{and} \quad E_2 = \frac{E_{\text{inter}}}{SS_T}, \quad (3.4) \]

where \( SS_T = \sum_{i=1}^{n-1} (\tau_i - \bar{\tau})^2 \). \( \bar{\tau} \) is the average interval value, and \( \bar{\tau} = \frac{1}{n-1} \sum_{i=1}^{n-1} \tau_i \).

Suppose that the number of intervals in the \( j \)th cluster is \( n_j = \sum_{i=1}^{n-1} x_{ij} \) and the center of the \( j \)th cluster is \( \tau^{(j)} = \frac{1}{n_j} \sum_{i=1}^{n-1} x_{ij} \bar{\tau}_i \), \( E_{\text{intra}} \) and \( E_{\text{inter}} \) are defined as

\[ E_{\text{intra}} = \sum_{j=1}^{K} \sum_{i=1}^{n-1} x_{ij} (\tau_i - \tau^{(j)})^2, \quad (3.5) \]

and

\[ E_{\text{inter}} = \sum_{j=1}^{K} (\tau^{(j)} - \bar{\tau})^2, \quad (3.6) \]

respectively. \( E_{\text{inter}} \) is a non-decreasing function of the number of clusters, \( K \), and quantifies the variation among clusters. \( E_{\text{intra}} \) is a non-increasing function of \( K \) and quantifies the variation within clusters. These two indices were used as measures of cluster optimality and stopping criteria in conventional hierarchical clustering algorithm Jung et al. (2003). When all interval values belong to one cluster, \( E_{\text{intra}} = SS_T = \sum_{i=1}^{n-1} (\tau_i - \bar{\tau})^2 \) and \( E_{\text{inter}} = 0 \). When each interval value forms its own cluster, \( E_{\text{intra}} = 0 \) and \( E_{\text{inter}} = SS_T = \sum_{i=1}^{n-1} (\tau_i - \bar{\tau})^2 \). Thus, given the forms defined in Eq. (4.6), \( 0 \leq E_1, E_2 \leq 1 \).

\( I_P \) and \( U_P \) evaluate the clustering configuration from a perspective of interpretability. \( I_P \) and \( U_P \) are referred to as pattern intensity and pattern utility, respectively. These two indices are calculated from the periodic patterns detected from the symbol sequence corresponding to a given clustering configuration. As shown in Figure 3.3, \( E_1 \) and \( E_2 \) are calculated based on clustering configuration and \( I_P \) and \( U_P \) are calculated from the obtained symbol sequence. The numbers listed under curly braces show the length of the periodic segments used to calculate \( I_P \) and \( U_P \).
The clustering of an interval sequence $\tau$ creates a symbol sequence $s = (s_1, s_2, ..., s_{n-1})$, where $s_i$, $i = 1, 2, ..., n - 1$, represents the cluster index of the corresponding interval value $\tau_i$ and $s_i = S_j$, if $x_{ij} = 1$ (Definition 3). $I_P$ is defined as

$$I_P = \frac{\sum_{i=1}^{n-1} I(s_i)}{n - 1}, \quad (3.7)$$

where $I(\cdot)$ is a periodicity indicator function. For a sequence $s$, $I(s_i)$ is equal to 1 if $s_i$ belongs to a certain periodic pattern, and is equal to 0 otherwise. Consider the symbol sequence in Figure 3.3, $I(s_i) = 1$ for $i = 1, 2, ..., 6$, since the first six symbols, 'AAAAAA', show a periodic pattern. $I_P$ defined based on $I(\cdot)$ represents the proportion of symbols that belong to certain periodic patterns. The value of $I_P$ is related to the number of clusters of intervals. When each interval value forms its own cluster, $\sum_{i=1}^{n-1} I(s_i) = 0$, since there is no periodic pattern formed. And when all interval values form one single cluster, $\sum_{i=1}^{n-1} I(s_i) = n - 1$. Thus, $0 \leq I_P \leq 1$.

$U_P$, $0 \leq U_P \leq 1$, is defined as a function of average length ratio of segments with periodic patterns. Suppose $M$ such segments are detected from the sequence $s$, and each segment has a length of $l_v$, $v = 1, 2, ..., M$, the average length ratio $\bar{l}$ is defined as $\bar{l} = \frac{1}{M} \sum_{v=1}^{M} l_v/(n - 1)$, and

$$U_P = \frac{\min(\sqrt{0.5}, \sqrt{\bar{l}})}{\sqrt{0.5}} = \begin{cases} \sqrt{\bar{l}/0.5}, & \text{if } \bar{l} < 0.5, \\ 1, & \text{otherwise.} \end{cases} \quad (3.8)$$
The rationale of using $U_P$ instead of $I$ is that $I$ tends to have very small values at large number of clusters and will sharply increase when the number of clusters reduces from two to one. The increase may outweigh other terms in the objective function and thus, the optimal number of clusters may be equal to one. To avoid this, $I$ is compared with 0.5, which is the value of $I$ when sequence $s$ contains two clusters and every $s_i$ belongs to a certain periodic pattern. The reason of taking square root is to increase the gradient of $U_P$ when the number of clusters is large. This adjustment can ensure the effect of $U_P$ on the merging of clusters. The comparison between $I$ and 0.5 can also decreases the gradient of $U_P$ when the number of clusters is small and can prevent the over-weighing of $U_P$. Given the definition of $I_P$ and $U_P$, the symbol sequence in Figure 3.3 gives $I_P = \frac{15}{21}, I = \frac{1}{4}(\frac{6}{21} + \frac{3}{21} + \frac{2}{21} + \frac{4}{21}) \approx 0.179$, and $U_P = \sqrt{0.179/0.5} = 0.598$. As the symbol sequence generated from the proposed approach contains more elements involved in periodic patterns (a high value for $I_P$) and patterns of longer length (a high value for $U_P$), the obtained patterns may have better interpretability.

3.3.2 Multi-start local search algorithm for modified hierarchical clustering

Given the proposed objective function Eq. (3.2), the optimal clustering configuration is determined by solving a complex optimization problem. It is challenging to achieve the global optimal solution, because the number of ways to cluster $n - 1$ interval values into $K$ clusters can be extremely large, resulting in a huge search space. Thus, this chapter proposes a heuristic multi-start local search (MSLS) algorithm. It extends the basic local search algorithm Aarts and Lenstra (1997), which has one start point, explores only one section of the whole search space and thus, has higher possibility to converge to local optima.
The MSLS algorithm increases the chance of converging to a global optima by restarting multiple local searches from different start points. It consists of three main steps (denoted as "SS") and involves iterative local search from multiple start points, which correspond to different temporary clustering configurations.

**SS1: Initial clustering.** The algorithm starts with a conventional clustering of $\tau$, based on Ward’s distance Olson (1995). This initial clustering will create a dendrogram, the levels of which indicate different clustering configurations, as shown in Figure 3.4. These temporary clustering configurations will be used as the start points of the iterative local search for APP detection. An initial start point can be selected at any level with more than one cluster in the dendrogram, considering the trade-off between potential of global optimality (high when it is close to the bottom level of the dendrogram) and the computational efficiency (high when it is close to the top level).
**SS2**: *Iterative local searching*. Given the initial start point, a local search for agglomerative hierarchical clustering will be conducted. This local search is different from the initial clustering of SS1 in two aspects: (i) the local search is according to the index $J$ defined in the objective function Eq. (3.2), whereas the initial clustering is based on Ward’s distance; (ii) the local search starts from a given clustering configuration, whereas the initial clustering starts with each time interval value as an individual cluster. The local searching procedure recursively merges two clusters that are selected to give the maximum increase of $J$, as defined in objective function Eq. (3.2), until a single cluster is formed with all interval values. The clustering configuration $C^\#$ that gives the maximum index value, $J^\#$, is the local optimum and will be set as the temporary optimal clustering configuration, $C_{\text{temp}}^\*$, which gives $J_{\text{temp}}^\*$. The algorithm will move up to the next level of the dendrogram, as
shown in Figure 3.4, and conduct the local search to generate a new local optimum. The corresponding new $C^\#$ and $J^\#$ will be used to replace $C^\star_{temp}$ and $J^\star_{temp}$, if $J^\# \geq J^\star_{temp}$. The procedure will proceed upward along the dendrogram iteratively until the second top level in the dendrogram. A flowchart of this MSLS algorithm is illustrated in Figure 3.5.

**SS3:** *Optimal clustering formulating.* The final $C^\star_{temp}$ and $J^\star_{temp}$ will be selected as the optimal clustering configuration and the maximum objective function value, $C^\star$ and $J^\star$. This configuration is used to cluster the interval values and transform the interval sequence into a sequence of cluster symbols. As shown in Figure 3.2, symbols from an alphabet set, e.g. \{'A', 'B', 'C', ...\}, are assigned to represent different clusters in a cluster index sequence. The assignment is based on the order of occurrence of clusters in the sequence. For instance, if a cluster index sequence contains two clusters, the first and the second cluster occurred in the sequence will be denoted with symbol 'A' and 'B', respectively.

### 3.3.3 Convolution-based periodicity detection

The proposed approach to detecting periodic patterns is based on three inputs: (i) a symbol sequence $s = (s_1, s_2, ..., s_{n-1})$ of length $n - 1$; (ii) a period range, $R$, with default value $[1, \lfloor \frac{n-1}{2} \rfloor]$, where $\lfloor \rfloor$ is a floor function; and (iii) a minimum required number of repetitions for a pattern to be considered as periodic, $r_{min}$. The algorithm consists of the following steps (denoted as "DS") :

**DS1:** *Define templates.* A template sequence is defined, for a given period $p$, shown in Figure 3.6. Illustrated of the convolution based periodicity detection
\[ 1 \leq p \leq R, \text{ as} \]
\[
\mathbf{T}_p^1 = \begin{cases} 
(1, -1), & \text{if } p = 1, \\
(1, V_p, -1), & \text{if } p > 1,
\end{cases}
\] (3.9)

where \( V_p \) is a sequence of \((p - 1)\) zeros. For example, the template sequence for period \( p = 2 \) is \((1, 0, -1)\).

**DS2:** Convert symbol sequence \( s \). This conversion is implemented by uniquely mapping symbols in \( s \) with arbitrary numerical values. A new numerical sequence \( \mathbf{X}_N \) is formed for convolution calculation. For example, the symbol sequence in Figure 3.3 can be mapped to a numerical sequence \( \mathbf{X}_N \), where
\[
\mathbf{X}_N = (1, 1, 1, 1, 1, 2, 2, 2, 1, 3, 2, 2, 1, 2, 1, 2, 1, 2, 1),
\]
if the symbols ‘A’, ‘B’, and ‘C’ are mapped to 1, 2 and 3, respectively, as shown in Figure 3.6.

**DS3:** Detect periodic patterns for a given period \( p \). With a convolution operation, the numerical sequence converted in DS2 is firstly transformed to contain segments that indicate the existence of patterns with period \( p \). The segments will then be explored to find the location and the length of periodic patterns.

**DS3-1:** Perform valid convolution. Given a sequence \( \mathbf{X}_N \) of length \((n - 1)\) and a template \( \mathbf{T}_p^1 \) of length \((p + 1)\), the valid convolution results in a new sequence \( \mathbf{Y}_1 \) of length \((n - p - 1)\), where its \( i \)th \((i = 1, 2, ..., n - p - 1)\) element, denoted as \( \mathbf{Y}_{1,i} \), is calculated as
\[
\mathbf{Y}_{1,i} = \sum_{j=1}^{p+1} T_{p,(p+2)-j}^1 \cdot \mathbf{X}_{N,i+(j-1)}.
\] (3.10)

\( T_{p,(p+2)-j}^1 \) is the \([(p + 2) - j]\)th element of \( \mathbf{T}_p^1 \) and \( \mathbf{X}_{N,i+(j-1)} \) is the \([i + (j - 1)]\)th element of \( \mathbf{X}_N \). Lee et al. (2009). \( \mathbf{Y}_1 \) contains zero elements that indicate recurrence of numerical values in \( \mathbf{X}_N \), with a period of \( p \). This numerical recurrence corresponds to the symbolic recurrence in \( s \). For example, given \( \mathbf{X}_N \) and a template with period of 2, \( \mathbf{T}_2^1 = (1, 0, -1), \) \( \mathbf{Y}_1 \) is obtained as \((0, 0, 0, 0, 1, 1, 0, -1, 1, 1, -1, -1, 0, 2, -1, -1, 0, 0, 0, 0)\), as shown in Figure 3.6. Specifi-
Y_{1,17} = T_{2,3}^1 X_{N,17} + T_{2,2}^1 X_{N,18} + T_{2,1}^1 X_{N,19}
   = (-1) \cdot 1 + 0 \cdot 2 + 1 \cdot 1 = 0

Since X_{N,17} = X_{N,19} = 1, above calculation gives a value of 0, indicating the recurrence of symbol "A". Similarly, Y_{1,18} = Y_{1,19} = 0. This convolution calculation according to Eq. (3.10) naturally resembles the search for periodic patterns, as described in Definition 2. By setting T_{p,1}^1 = 1, T_{p,1+p}^1 = -1, and the other elements in T_p^1 to 0, the convolution is equivalent to comparing X_{N,i} and X_{N,i+p} only, i.e., \((-1)X_{N,i} + X_{N,i+p}\), which can be 0 if and only if \(X_{N,i} = X_{N,i+p}\). The setting of T_p^1 also ensures the flexibility of the proposed algorithm. As long as the same symbols are uniquely mapped with a certain numerical value, their recurrence with a period of p can be detected.

The detection of patterns with period p requires at least p consecutive zeros in Y_1, as it corresponds to one (v = 1 in Definition 2) repetition. In practice, it may be required that \(v \geq r_{\text{min}}\), where \(r_{\text{min}}\) is given as the minimum required number of repetitions. As shown in Figure 3.6, Y_1 has a segment of three \((p + 1 = 3)\) consecutive zeros, i.e., \((Y_{1,17}, Y_{1,18}, Y_{1,19}) = (0, 0, 0)\). These three consecutive zeros correspond to the segment 'ABABA' in s. The following sub-step is used to find such segments and locate the repetitive periodic patterns in s.

**DS3-2: Locate consecutive zero segments in Y_1.**

Denoting the starting index and the ending index of a consecutive zero segment in Y_1 as I_{S}^* and I_{E}^*, respectively, the corresponding starting index of the periodic pattern in s is I_S = I_{S}^* and the length of the segment is I_{E}^* - I_{S}^* + (p + 1). It is noted that the detected segment may not show an integer number of repetitions of the periodic pattern. For example, the segment 'ABABA' of s in Figure 3.6 contains one, but less than two, complete repetitions of 'AB'’s. In this chapter, the incomplete repetition part (i.e. the last 'A') will be discarded. As a result, the length of a periodic pattern in s can be defined based on I_S^*, I_S^* and p, as
\[ \ell = p \left\lfloor \frac{I^*_E - I^*_S + (p + 1)}{p} \right\rfloor. \] (3.11)

The ending index \( I_E \) of the periodic pattern in \( s \) is \( I_S + \ell - 1 \). When there are multiple segments of consecutive zeros with given \( p \), DS3-2 will be performed multiple times to achieve multiple pairs of \((I_S, I_E)\)'s. As illustrated in Figure 3.6, there are two segments of consecutive zeros in \( Y_1 \), given \( p = 2 \). For the first segment, \( I^*_S = 1 \) and \( I^*_E = 4 \). The corresponding starting and ending indices of the pattern (with period of 2) in \( s \) are \( I_S = 1, I_E = 1 + 2 \left\lfloor \frac{4-1+(2+1)}{2} \right\rfloor - 1 = 6 \). For the second segment, \( I^*_S = 17 \) and \( I^*_E = 19 \). The corresponding starting and ending indices of the periodic pattern in \( s \) are \( I_S = 17, I_E = 17 + 2 \left\lfloor \frac{19-17+(2+1)}{2} \right\rfloor - 1 = 20 \). Note that the last 0 in the second segment is discarded since it is not a part of a complete periodic repetition.

**DS1-DS3** will be performed iteratively for all the \( p \)'s, \( 1 \leq p \leq R \), to detect periodic patterns with different periods. These detected patterns may be redundant. Two types of redundant patterns are considered in this chapter: (i) the patterns with period as a multiple of shorter periods. For example, the segment 'AAAAAA' in Figure 3.6 can be viewed as a pattern with five(5) repetitions of 'A', or with two(2) repetitions of 'AA'. Since 'AA' is composed of a multiple of 'A', only the pattern with shorter period will be retained. (ii) the nested patterns. For example, the pattern 'BBBBB'C' in the segment 'BBBBBCBBBBC' nests a pattern 'B' with three(3) repetitions, as shown in Figure 3.2 (b). It is rational to retain the pattern with longer period. The retained non-redundant patterns will be used in calculating \( I_P \) and \( U_P \), as defined in the objective function Eq. (3.2).

The proposed APP-detection algorithm is based on valid convolution. The computational complexity for detecting patterns with period \( p \) is \( O((n - 1)\log(n - 1)) \), which is smaller than the complexity, \( O((n - 1)^2) \), of conventional detection method Rasheed et al. (2011). This advantage will ensure the applicability of the proposed method in real-world complex and computationally-demanding problems.
3.4 Case studies

The effectiveness of the proposed approach is demonstrated with both a study on numerically simulated interval sequence data and an investigation of real-world temporal operational sequence data collected from mining haul trucks. In numerical simulation study, target cluster index sequence containing ground truth patterns is assumed. Periodic patterns detected from both the proposed clustering approach and the conventional clustering approach are compared with ground truth patterns. To quantify the pattern detection performance, $I_D(s, s^T)$ is defined for a clustering result symbol sequence $s$, $s = (s_1, s_2, ..., s_n)$, based on a given target index sequence, $s^T$, $s^T = (s^T_1, s^T_2, ..., s^T_n)$, as

$$I_D(s, s^T) = \frac{\sum_{j=1}^{M} \sum_{m=B_j}^{E_j} I(s_m) \cdot \phi(s_m, s^T_m)}{\sum_{j=1}^{M} (E_j - B_j + 1)},$$

where $B_j$ and $E_j$ are the positions (indices in a symbol sequence) of the first symbol and the last symbol in the $j$th segment with periodic patterns in $s^T$, respectively. $M$ is the total number of such segments. $s_m$ and $s^T_m$ are the $m$th symbol in $s$ and $s^T$, respectively. $I(\cdot)$ is the same periodicity indicator function as defined in Eq. (4.3). Function $\phi(x, y)$ is a Kronecker operator, where

$$\phi(x, y) = \begin{cases} 
1, & \text{if } x = y, \\
0, & \text{otherwise}.
\end{cases}$$

Basically, $I_D$ is a ratio of the correctly clustered (as compared to the target index sequence) interval values that belong to periodic patterns in $s$ over the total length of all the periodic patterns in $s^T$. For example, consider two short symbol sequences, $s^T='ABABABCC'$ and $s='ACABABCC'$. $s^T$ contains two ($M = 2$) segments of periodic patterns, two(2) repetitions of 'AB' and one(1) repetition of 'C'. Thus, $B_1 = 1, E_1 = 6, B_2 = 7$, and $E_2 = 8$, resulting $I_D = \frac{4+2}{6+2} = 0.75$. When $s$ is the same as $s_T$, $I_D$ reaches the maximum value of 1.
3.4.1 Detecting APPs from simulated interval sequence

In the numerical simulation study, the target symbol sequence is set as 'ABCABCBACACAAAABACBAAABCBACAC', which contains two(2) repetitions of 'ABC', three(3) repetitions of 'A', one(1) repetition of 'CBAABC', five(5) repetitions of 'C' and one(1) repetition of 'BCBAC'. Interval sequences are simulated by linking each unique symbol to a random variable following discrete uniform distribution. There are two different settings for the parameters of the distributions: (i) clusters 'A', 'B' and 'C' are simulated by distributions $U(5, 6)$, $U(10, 13)$, and $U(48, 52)$, where $U(a,b)$ denotes a discrete uniform distribution with lower boundary at $a$ and upper boundary as $b$; (ii) clusters 'A', 'B' and 'C' are simulated by distributions $U(5, 6)$, $U(7, 12)$ and $U(48, 52)$. Given that, in setting 2, the upper boundary of cluster 'A' and the lower boundary of cluster 'B' are very close, interval values close to those boundaries are easy to be mis-clustered. Thus, it will be more difficult to correctly cluster the intervals in setting 2.

Given that interval values are generated randomly based on the specified uniform distribution, a hundred (100) simulation replications are conducted for each setting, and the mean values of performance index $I_D$ (denoted as $\bar{I}_D$) and the proportion of perfect detection (denoted as $P(I_D = 1)$) are reported in Table 3.2. It is shown that both methods can effectively detect the patterns with high $\bar{I}_D$ and $P(I_D = 1)$ under setting 1. The performance difference between the two approaches is very small. However, for the more difficult setting 2, the proposed approach has a slightly reduced performance ($\bar{I}_D = 0.9$ and $P(I_D = 1) = 0.72$) while the conventional approach performs poorly ($\bar{I}_D = 0.71$ and $P(I_D = 1) = 0.01$).

| Table 3.2: Performance index calculation in simulation study |
|-------------------------------|-------------------|-------------------|-------------------|-------------------|
|                              | Setting 1          |                  | Setting 2          |                  |
|                              | $\bar{I}_D$ | $P(I_D = 1)$ | $\bar{I}_D$ | $P(I_D = 1)$ |
| Proposed Approach            | 0.95            | 0.85            | 0.90            | 0.72            |
| Conventional Approach        | 0.95            | 0.82            | 0.71            | 0.01            |
Figure 3.7: Performance evaluation in simulation study

The detected patterns are visualized using a pattern plot in this chapter. The plot shows the segments with detected APPs. The horizontal axis represents the locations (indices) in a symbol sequence, and the vertical axis represents the periodicity (the length of a period). Segments with different APPs are illustrated with boxes labeled with Roman letters. The patterns corresponding to different labels are explained in the legend. The number of repetitions in each segment can be calculated from the length of the boxes and the periodicity. The APP detection
results from one(1) replication in setting 2 are shown in Figure 3.7, where Figure 3.7 (a) shows the results from the proposed approach, and Figure 3.7 (b) shows that from the conventional approach. In both Figure 3.7 (a) and (b), the ground truth patterns are plotted as benchmark in shaded boxes. Figure 3.7 (a) shows that the proposed method detects all the true APPs whose contents are explained in legend. In contrast, the conventional approach outputs two clusters as shown in Figure 3.7 (b), leading to incorrect patterns.

3.4.2 Detecting APPs from mining operation sequence

The proposed APP detection methodology has also been applied to a case study on event log sequence of mining haul trucks. The event log contains ’Ready’ alarms triggered and logged by sensors. With the timestamps of the alarms, a time interval sequence can be obtained and analyzed to find informative periodic patterns.

The interval sequence is illustrated in Figure 3.8 (a) (interval values measured in unit of 1,000 seconds). Given the interval sequence, both the proposed approach and the conventional approach are used to perform clusterings and obtain the cluster index sequence in Figure 3.8 (b) and Figure 3.8 (c), respectively.

The convolution based APP detection method is applied with both the proposed modified clustering (considering both data similarity and pattern interpretability) and the conventional clustering (considering data similarity only). The APPs embedded in the symbol sequences are detected with the proposed clustering and the conventional clustering. The APPs are visualized in pattern plots, as shown in Figure 3.9 (a) and (b), respectively. It is noted that the conventional method detects short and intermittent patterns, which are difficult to interpret. On the contrary, the proposed method finds more APPs, with longer period. For instance, the combination of APPs IV and I, and a series of repetitions of APP II as in Figure 3.9 (a) gives a meaningful indication of significant events in mining operations, according to a domain expert. This pattern, however, is not detected using conventional approach.
3.5 Conclusion

This research studies the detection of interval-based APPs from temporal sequence data, where periodic patterns are embedded in a sequence of intervals between events of a certain type. The unavoidable randomness brings extra disturbance to the interval values and makes the detection of APPs from real-world temporal sequences extremely difficult. Existing approaches to detecting APPs rely on subjective knowledge of magnitude of disturbance, and have limited capability to set multiple tolerances for multiple potential target interval values. To overcome these limitations, this chapter formulates the determination of tolerance values as an ag-

Figure 3.8: Interval sequence and cluster index sequence
glomerate hierarchical clustering problem. A modified objective function is proposed to determine the clustering configuration, with consideration of both data similarity and pattern interpretability. A heuristic MSLS algorithm is developed to improve the convergence to global optima. The clustering of interval values converts the interval sequence into a sequence of cluster symbols, from which, APPs are detected with a convolution-based template matching algorithm. The effectiveness of the proposed methodology is demonstrated with case studies based on temporal sequence data collected from numerical simulation and real-world mining operations.

The proposed methodology possesses some limitations. First, the proposed approach uses a weighted summation of four terms as the objective function of clustering. The weighting coefficients can be specified according to the estimation of...
the potential pattern length or interval value similarity. The clustering can also be formulated as a multi-objective optimization problem to avoid weight assignment and achieve non-dominated solutions. For the convolution-based pattern detection algorithm, one limitation is the prerequisite on period range as an input. The default range is set to $[1, \lfloor (n - 1)/2 \rfloor]$. The computation load for a long sequence will be very high due to exhaustive search within the range. In addition, the algorithm currently detects only exact repetition of symbols. A symbol sequence, such as 'ABCABDABC', is not considered as containing a periodic pattern, since it is not an exact repetition of 'ABC'. This consideration is reasonable in the APP detection of numerical time interval sequence, since the time interval values are usually significantly different from others. However, it may be beneficial to consider inexact repetition in some applications, such as DNA sequence analysis. Extended algorithm to detect this type of pattern will be developed in future.
4.1 Problem statement

Reducing equipment failure in manufacturing systems is of crucial importance for improving productivity and lowering maintenance cost. To achieve this objective, operational data and equipment health status data are collected continuously for monitoring and analysis purpose. Given the collected data, it is desirable to construct statistical models that can predict the time to equipment failure so that preventive maintenance could be scheduled in time. One such example is the prediction of mining truck failures based on the collected operational data. As aforementioned in Chapter 1, effective prediction of truck failure could ensure productivity and safety of mining operations.

For the topic of failure prediction, various methods have been proposed. Kumar and Klefsjö (1992) conducted a preliminary study of reliability statistics of a fleet of load-haul-dump machines and predicted the occurrence of failures using power law process models. Vilalta and Ma (2002) proposed a rule-based approach that predicted rare failure event based on extracted eventset patterns prior to the failure. Lin and Siewiorek (1990) developed dispersion frame technique (DFT) to analyze the time of error occurrence in near past and uses heuristic rules to predict pending failure. Salfner and Malek (2007) adopted hidden semi-Markov models for failure prediction. Apart from these methods, reliability models such as Cox proportional hazard model have been adopted for equipment failure event prediction. Li et al. (2007b) proposed a methodology that extracted failure patterns as predictors and used them in Cox model for failure prediction.

As a well-established statistical method in analyzing time to failure data, the
Cox proportional hazard model assumes that covariates are measured without error. However, this assumption can be violated in many real world applications. Recall in Chapter 1, the collected load weight of mining truck is subject to measurement errors due to inaccuracy of measurement device. In addition, the measured data contains many records with invalid values of zero tons. Existing approaches consider mainly two types of measurement errors associated with covariates. One is called classical error and the other is called Berkson error. Effects of these errors can be either additive or multiplicative. The classical error refers to the case that a measurement of variable of interest is obtained, but the true covariate value is not observed and only an erroneous measurement containing the error is obtained. The Berkson error, on the other hand, represents the case that no direct measurement of the variable of interest is available. In this case, a surrogate value is measured. The difference between the surrogate and the true covariate is referred as Berkson error.

Abundant research effort has been made to study Cox proportional hazard model with covariate measurement error. Hughes (1993) studied the effect of the classical measurement error on parameter estimation. A relationship between the naive estimates ignoring measurement error and true coefficient value under different levels of error variances is obtained through simulation. The simulation outcome suggested that the occurrence of classical errors leads to biased estimates. In the single covariate case, the bias is demonstrated as attenuated magnitude of the coefficient estimate. To address the biased estimate, many of the existing approaches to handling measurement error are extended or adopted to analyze Cox model. The regression calibration technique, which replaces the covariate of interest by its conditional expectation given the observed covariate, was proposed by Prentice (1982). The advantage of this method is its simplicity, as after obtaining the conditional expectation used in replacement, the model can be fit using existing software designed for proportional hazard model. However, this model does not guarantee asymptotically unbiasedness and only works when the failure event is very rare. The simulation-extrapolation (SIMEX) procedure proposed by Cook and Stefanski (1994) represented another mainstream technique in dealing with measurement er-
ror. The key idea is to simulate new data by adding increasing amounts of noises to the measured erroneous covariates and estimating the regression coefficients on these simulated data. Then the expectation of the estimator is modeled as a function of measurement error variance and extrapolation is used to obtain estimate under the case of no measurement error. The simulation study showed that SIMEX approach produced estimators that are nearly asymptotically unbiased for nonlinear regression models. The other group of methods, represented by research work from Nakamura (1992), proposed a corrected score function so that even using the erroneous covariate still results in an asymptotically unbiased estimate. However, for Cox model estimation, only approximate corrections can be made. This approach has been further studied by Huang and Wang (2000) who used replicate measurements on covariates to estimate the variance of measurement error and by Tsiatis and Davidian (2001) whose method allowed time-dependent covariates. The likelihood-based approach proposed in Hu et al. (1998) differed from the above-mentioned approaches in that the unobserved covariate is treated as independent normal random effects and it uses a full likelihood function consisting of joint density functions of observed covariate and unobserved true covariate. The model estimation can be easily achieved by adapting existing software functions used for fitting nonlinear mixed effect models.

In contrast to the variety of methods proposed for Cox model with classical measurement error, the research of estimation of regression coefficient under the mixture of classical errors and Berkson errors is relatively limited. Reeves et al. (1998) pioneered the research work of mixture of errors and discussed how the errors affect the estimates from logistic regression in an epidemiological study. Approximations to logistic function using standard normal cumulative distribution function is adopted and a formula that links the naive estimates with the true regression coefficient is proposed. Mallick et al. (2002) proposed Bayesian methods for semi-parametric estimation approach in an analysis where the response variable is binary and covariates are measured with mixture of errors. In contrast to Reeves et al. (1998) which assumed normal distribution for true covariate, the true covariate was also modeled
A study considering the correlated structure of the Berkson error in the mixture of errors was conducted by Li et al. (2007a). The study also focused on binary response in logistic regression and discussed both Bayesian methods using Markov chain Monte Carlo and Monte Carlo expectation and maximization method. Carroll et al. (2007) proposed a non-parametric regression approach that used kernel estimation and characteristic functions of Berkson and classical error. The estimation procedure of Cox proportional hazard model under the mixture of errors, however, has not been reported in existing research literature.

This chapter proposed the estimation method for proportional hazard model under the mixture of additive classical and Berkson errors on covariates. The method firstly links the unobserved true covariate with the observed one using linear regression models. The error term in this regression model is treated as a latent variable. To maximize likelihood that contains latent variable, a Monte Carlo expectation maximization (MCEM) approach is proposed. The Monte Carlo E-step in the MCEM facilitates the calculation of posterior mean, whose distribution does not have closed form. According the property of MCEM algorithm, the obtained estimate will be asymptotically unbiased under certain regularity conditions.

The rest of this chapter is organized as follows. Section 4.2 explains the proposed approach that allows the estimation of Cox model coefficients under the mixture of classical and Berkson error. Subsection 4.2.1 gives the definition of the mixture of classical and Berkson error and how the true covariate and observed covariate can be linked with linear regression. Subsection 4.2.2 details the model inference using MCEM method under the mixture of errors. A simulation study is included in section 4.3 to demonstrate the effectiveness of the proposed approach. Section 4.4 summarizes the proposed approach and concludes the chapter.
4.2 Proposed approach

4.2.1 Modeling mixture of measurement error

The formal definition of mixture of classical and Berkson errors is firstly explained. As aforementioned, the mixture of error represents the situation that the direct measurement of the true covariate of interest is not available, and a surrogate is measured with error at the same time. In mathematical formulation, if the true covariate is denoted as $x_i$, $i = 1, 2, ..., N$, where $i$ is observation index, $N$ is the sample size, and $w_i$ is the observed erroneous measurement of the surrogate $s_i$, the relationship between these covariates is formulated as

$$x_i = s_i + e_i^b$$
$$w_i = s_i + e_i^c$$

(4.1)

where $e_i^b$ represents the Berkson error and $e_i^c$ represents the classical error. Assume $e_i^c \sim N(0, \sigma_c^2)$, $e_i^b \sim N(0, \sigma_b^2)$, $s_i \sim N(\mu, \tilde{\sigma}^2)$, the relationship between the true covariate $x_i$ and the observed covariate $w_i$, can be expressed using a simple linear regression, as explained in Reeves et al. (1998).

To obtain the regression equation, both $x_i$ and $w_i$ are firstly centered to mean zero by defining two new variables $x_i^*$ and $w_i^*$ as

$$x_i^* = x_i - E(x_i) = x_i - \mu,$$
$$w_i^* = w_i - E(w_i) = w_i - \mu,$$

A simple linear regression model linking $x_i^*$ and $w_i^*$ is

$$x_i^* = E(x_i^*|w_i^*) + \epsilon_i = \gamma w_i^* + \epsilon_i$$

(4.2)

The property of simple linear regression has proved that

$$\gamma = \frac{Cov(x_i^*, w_i^*)}{Var(w_i^*)} = \frac{Cov(s_i + e_i^b - \mu, s_i + e_i^c - \mu)}{Var(s_i + e_i^c - \mu)}.$$ 

(4.3)
Assuming the Berkson and classical errors are uncorrelated, then $\text{Cov}(s_i + e^b_i - \mu, s_i + e^c_i - \mu) = \text{Var}(s_i) = \tilde{\sigma}^2$. Eq.(4.3) becomes

$$\gamma = \frac{\text{Var}(s_i)}{\text{Var}(s_i + e^c_i - \mu)} = \frac{\tilde{\sigma}^2}{\tilde{\sigma}^2 + \sigma^2_c}$$

(4.4)

Therefore the regression equation Eq.(4.2) can be written as

$$x^*_i = E(x^*_i | w^*_i) + \epsilon_i = \frac{\tilde{\sigma}^2}{\tilde{\sigma}^2 + \sigma^2_c} w^*_i + \epsilon_i$$

(4.5)

The variance of $\epsilon_i$ can be calculated as

$$\text{Var}(\epsilon_i) = \text{Var}(x^*_i - \frac{\tilde{\sigma}^2}{\tilde{\sigma}^2 + \sigma^2_c} w^*_i)$$

$$= \tilde{\sigma}^2 + \sigma^2_b + \frac{(\tilde{\sigma}^2)}{\tilde{\sigma}^2 + \sigma^2_c}^2 (\tilde{\sigma}^2 + \sigma^2_c) - 2 \text{Cov}(x^*_i, \frac{\tilde{\sigma}^2}{\tilde{\sigma}^2 + \sigma^2_c} w^*_i)$$

$$= \tilde{\sigma}^2 + \sigma^2_b + \frac{(\tilde{\sigma}^2)^2}{\tilde{\sigma}^2 + \sigma^2_c} - 2 \frac{(\tilde{\sigma}^2)^2}{\tilde{\sigma}^2 + \sigma^2_c}$$

$$= \tilde{\sigma}^2 + \sigma^2_b - \gamma^2 (\tilde{\sigma}^2 + \sigma^2_c)$$

This shows that $\epsilon_i \sim N(0, \sigma^2_i)$, $\sigma^2_i = \tilde{\sigma}^2 + \sigma^2_b - \gamma^2 (\tilde{\sigma}^2 + \sigma^2_c)$. The relationship between $x_i$ and $w_i$ becomes

$$x_i - \mu = \frac{\tilde{\sigma}^2}{\tilde{\sigma}^2 + \sigma^2_c} (w_i - \mu) + \epsilon_i = \gamma (w_i - \mu) + \epsilon_i$$

and can be simplified as

$$x_i = \mu + \gamma (w_i - \mu) + \epsilon_i,$$

(4.6)

While this equation is derived with univariate covariate, it can be extended easily to a scenario that multiple covariates are error-prone. Suppose that $x_i$, $s_i \sim N(\mu, \tilde{\Sigma})$, $e^b_i \sim N(0, \Sigma_b)$, $e^c_i \sim N(0, \Sigma_c)$, and $w_i$ correspond to true covariate, surrogate, Berkson error, classical error and observed covariate of dimension $p$, respectively,
the multivariate counterpart of Eq.(4.6) is

\[ x_i = \mu + \Gamma (w_i - \mu) + \epsilon_i, \]  

(4.7)

where \( \Gamma = \bar{\Sigma} (\bar{\Sigma} + \Sigma_\epsilon)^{-1}, \epsilon_i \sim N(0, \Sigma_\epsilon), \) and \( \Sigma_\epsilon = (\bar{\Sigma} + \Sigma_b) - \Gamma \bar{\Sigma}. \)

4.2.2 Model inference under mixture of error

The Cox proportional hazard model has been widely used in survival analysis. The model can be readily adopted to model the failure time of equipment given its working conditions, which are used as covariates. The model quantifies the failure time with the hazard rate function \( h(t) \), which has a multiplicative structure as

\[ h(t_i) = h_0(t_i) \exp(x_i' \beta_x + z_i' \beta_z), \]  

(4.8)

where \( i = 1, 2, ..., N \), \( N \) is the sample size. \( h(t_i) \) is the hazard rate for \( i \)th failure occurred at time \( t_i \), given its error-prone covariate \( x_i \) and error-free covariate \( z_i \). The hazard rate at time \( t_i \) quantifies the instantaneous probability of failure given that the subject still survives at time \( t_i \). This hazard rate is formulated as a product between the baseline hazard rate \( h_0(t_i) \) and the effect of covariate \( x_i \) and \( z_i \), i.e. \( \exp(x_i' \beta_x + z_i' \beta_z) \). For covariates, this chapter considers the time-invariant ones, where covariate \( x_i \) and \( z_i \) are not changing along time. The function \( h_0() \) quantifies the hazard rate when covariates are at their benchmark value equal to zero. \( \beta_x \) and \( \beta_z \) are unknown regression coefficients that need to be estimated from the data.

The procedure of estimating \( \beta_x \) and \( \beta_z \) when no measurement errors exist is firstly reviewed. Based on the Eq.(4.8) the complete log-likelihood assuming \( x_i \) is known and no censoring of observations can be written as:

\[ l(\beta_x, \beta_z) = \sum_{i=1}^{N} \{ \log(h_0(t_i)) + (x_i' \beta_x + z_i' \beta_z) - H_0(t_i) \exp(x_i' \beta_x + z_i' \beta_z) \}, \]  

(4.9)

Since \( h_0(t_i) \) and \( H_0(t_i) \) are also unknown, they need to be estimated through the
maximization of \( l(\beta_x, \beta_z) \). The usual approach to this maximization problem is to formulate the profile likelihood, which firstly finds the optimal \( h_0(t_i) \) and \( H_0(t_i) \) by assuming \( h_0(t_i) \) as piece-wise constant and treating \( \beta_x \) and \( \beta_z \) as given. Then the optimal \( h_0(t_i) \) and \( H_0(t_i) \), which are now functions of \( \beta_x \) and \( \beta_z \), are plugged in Eq.(4.9) and the partial log-likelihood is formed as

\[
l_P(\beta_x, \beta_z) = \sum_{i=1}^{N} \{ x_i' \beta_x + z_i' \beta_z - \log \sum_{j \in R(t_i)} \exp(x_j' \beta_x + z_j' \beta_z) \} - N
\]

The maximum likelihood estimates of \( \beta_x \) and \( \beta_z \) are obtained through solving the equation that sets the partial derivative of \( l_P(\beta_x, \beta_z) \) with respect to \( \beta_x \) and \( \beta_z \) to zero. Once the estimates \( \hat{\beta}_x \) and \( \hat{\beta}_z \) are obtained, the function \( h_0() \) and \( H_0() \) can be estimated as

\[
\hat{h}_0(t) = \frac{1}{\sum_{j \in R(t)} \exp(x_j' \beta_x + z_j' \beta_z)}, \quad \text{and} \quad \hat{H}_0(t) = \sum_{t_j \leq t} \hat{h}_0(t_j),
\]

where \( R(t) \) represents the risk set at time \( t \), i.e. \( R(t) = \{ j | t_j \geq t, j = 1, 2, ..., N \} \).

When the mixture of classical and Berkson errors are present in covariate \( x_i \), the true covariate \( x_i \) can no longer be observed and the likelihood function Eq.(4.9) can not be maximized directly. Note that the true covariate can be expressed as a linear function of observed covariate \( w_i \) using Eq.(4.7). Plugging this equation in Eq.(4.9) results in

\[
l(\beta_x, \beta_z | \epsilon_1, \epsilon_2, ..., \epsilon_N, \mu, \Sigma)
= \sum_{i=1}^{N} \{ \log(h_0(t_i)) + (|\mu + \Gamma(w_i - \mu) + \epsilon_i|' \beta_x + z_i' \beta_z) \\
- H_0(t_i) \exp(|\mu + \Gamma(w_i - \mu) + \epsilon_i|' \beta_x + z_i' \beta_z) \} \}
\]

where \( \epsilon_1, \epsilon_2, ..., \epsilon_N \) are latent variables. To address the model identifiability issue, this chapter assumes that the variance (i.e. \( \Sigma_c \)) of classical error on covariate \( x_i \) is known. The other nuisance parameters, including \( \mu, \tilde{\Sigma}, \Sigma_b \) need to be estimated.
from the data. Therefore the complete log-likelihood function can be written as

\[
l(\beta_x, \beta_z, \mu, \Sigma, \Sigma_b) = f(\{t_i\}, \{w_i\} | \beta_x, \beta_z, \{\epsilon_i\}, \mu, \Sigma, \Sigma_b) \]

(4.13)

\[
= \log \{f_t(\{t_i\} | \{w_i\}, \beta_x, \beta_z, \{\epsilon_i\}, \mu, \Sigma, \Sigma_b) \} \]

(4.14)

\[
\approx \log \{f_t(\{t_i\} | \{w_i\}, \beta_x, \beta_z, \{\epsilon_i\}) f_w(\{w_i\} | \mu, \Sigma, \Sigma_b) \} \]

(4.15)

\[
= \log[f_t(\{t_i\} | \{w_i\}, \beta_x, \beta_z, \{\epsilon_i\})] + \log[f_w(\{w_i\} | \mu, \Sigma, \Sigma_b)], \]

(4.16)

where \(\{t_i\}, \{w_i\}, \{\epsilon_i\}\) represent the set of all \(N\) failure time, observed error-prone covariate, and latent variables, respectively. The \(f(\{t_i\}, \{w_i\}) | \beta_x, \beta_z, \{\epsilon_i\}, \mu, \Sigma, \Sigma_b)\) is thus the joint density function of \(\{t_i\}\) and \(\{w_i\}\). The simplification from Eq.(4.14) to Eq.(4.15) relies on the fact that \(\{w_i\}\) is independent of coefficients \(\beta_x, \beta_z\) and \(\{\epsilon_i\}\) and a simplifying assumption that the error of estimating \(\mu, \Sigma, \Sigma_b\) solely based on \(\{w_i\}\) is negligible. Therefore, the obtained equation Eq.(4.16) indicates that the estimation of \(\mu, \Sigma, \Sigma_b\) will be based on \(\{w_i\}\) and can be performed independently of \(\beta_x, \beta_z\). The usual approach is to estimate \(\mu, \Sigma, \Sigma_b\) first with a linear random effect model and then plug in these estimates in the maximization of \(\log[f_t(\{t_i\} | \{w_i\}, \beta_x, \beta_z, \{\epsilon_i\})]\) for estimating \(\beta_x, \beta_z\). Assuming now the values of \(\mu, \Sigma, \Sigma_b\) are obtained, estimation of \(\beta_x, \beta_z\) is achieved
by maximizing the log-likelihood function

\[
\log[f_t(\{t_i\}|\{w_i\}, \beta_x, \beta_z, \{\epsilon_i\})] = \log[\prod_{i=1}^{N} f_t(t_i|w_i, \beta_x, \beta_z, \epsilon_i)]
\]

\[
= \log[\prod_{i=1}^{N} f_t(t_i|w_i, \beta_x, \beta_z, \epsilon_i)]
\]

\[
= \sum_{i=1}^{N} \log[f_t(t_i|w_i, \beta_x, \beta_z, \epsilon_i)]
\]

\[
= \sum_{i=1}^{N} \{ \log(h_0(t_i)) + (|\mu + \Gamma(\epsilon_i - \mu)| + \epsilon_i, \beta_x + z_i, \beta_z) - H_0(t_i)exp(|\mu + \Gamma(\epsilon_i - \mu) + \epsilon_i, \beta_x + z_i, \beta_z) \}
\]

(4.17)

By treating \( \epsilon_i \) as latent variable, the inference based on expectation-maximization (EM) algorithm is proposed to estimate the coefficient \( \beta_x \) and \( \beta_z \). The EM algorithm proceeds iteratively with two steps. The expectation step, i.e. E-step, computes the expectation of the log-likelihood with respect to the latent variable, based on its conditional distribution given the response and current estimate. The maximization step, i.e. M-step, performs the maximization of the expected log-likelihood so that an updated estimate is obtained. The algorithm iterates until the convergence of the estimate. The MCEM algorithm, which replaces the calculation of E-step with Monte Carlo integration, is adopted due to the lack of analytical form of conditional distribution involving hazard function.

Specifically, the MCEM algorithm proceeds with the following steps:

1. Generation of initial estimates for \( \beta_x, \beta_z \). These initial values are generated by first fitting a Cox model that ignores the measurement errors in \( x_i \). The obtained estimates are denoted as \( \hat{\beta}_x^{(0)}, \hat{\beta}_z^{(0)} \). The estimates for baseline hazard and cumulative baseline hazard function, \( \hat{h}_0^{(0)}(t) \) and \( \hat{H}_0^{(0)}(t) \), can be obtained by plugging \( \hat{\beta}_x^{(0)}, \hat{\beta}_z^{(0)} \) in Eq. (4.10).

2. Calculation of the conditional expected log-likelihood with respect to \( \epsilon_i, i = \)
According to Bayes theorem, the conditional distribution can be formatted as

\[
E[(\beta_x, \beta_z | \epsilon_i, \epsilon_2, ..., \epsilon_N, \mu, \Sigma)]
\]

\[
= \sum_{i=1}^{N} \left\{ \log(h_0(t_i)) + ([\mu + \Gamma(w_i - \mu) + \epsilon_i]^{\prime} \beta_x + z_i^{\prime} \beta_z) \right\}
\]

(4.18)

\[
= \sum_{i=1}^{N} \left\{ \log(h_0(t_i)) + ([\mu + \Gamma(w_i - \mu) + E(\epsilon_i)]^{\prime} \beta_x + z_i^{\prime} \beta_z) - H_0(t_i)exp([\mu + \Gamma(w_i - \mu) + \epsilon_i^{\prime} \beta_x + z_i^{\prime} \beta_z]) \right\}
\]

(4.19)

It is noted that the conditional expectation of log-likelihood involves calculation of \(E(\epsilon_i)\) and \(E(exp(\epsilon_i^{\prime} \beta_x))\). These expectation are with respect to \(\epsilon_i\) which has the distribution \(f(\epsilon_i | t_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)})\), i.e.

\[
E(\epsilon_i) = \int \epsilon_i f(\epsilon_i | t_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) d\epsilon_i
\]

and

\[
E(exp(\epsilon_i^{\prime} \beta_x)) = \int exp(\epsilon_i^{\prime} \beta_x) f(\epsilon_i | t_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) d\epsilon_i
\]

According to Bayes theorem, the conditional distribution can be formatted as

\[
f(\epsilon_i | t_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) = \frac{f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) f(\epsilon_i | \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)})}{\int f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) f(\epsilon_i | \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) d\epsilon_i}
\]

\[
= \frac{f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) f(\epsilon_i)}{\int f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) f(\epsilon_i) d\epsilon_i}
\]

\[
\propto f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) f(\epsilon_i),
\]

where

\[
f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) = \hat{h}_0^{(k)}(t_i) exp([\mu + \Gamma(w_i - \mu) + \epsilon_i]^{\prime} \hat{\beta}_x^{(k)} + z_i^{\prime} \hat{\beta}_z^{(k)})
\]

\[
\exp(-\hat{H}_0^{(k)}(t_i)exp([\mu + \Gamma(w_i - \mu) + \epsilon_i]^{\prime} \hat{\beta}_x^{(k)} + z_i^{\prime} \hat{\beta}_z^{(k}))},
\]

1, 2, ..., N using its distribution conditioned on response and current coefficient estimates. Suppose the current iteration index is \(k\), then given \(\hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}\), the conditional expected log-likelihood is

\[
E[l(\beta_x, \beta_z | \epsilon_1, \epsilon_2, ..., \epsilon_N, \mu, \Sigma)]
\]

\[
= \sum_{i=1}^{N} \left\{ \log(h_0(t_i)) + ([\mu + \Gamma(w_i - \mu) + \epsilon_i]^{\prime} \beta_x + z_i^{\prime} \beta_z) \right\}
\]

(4.18)

\[
= \sum_{i=1}^{N} \left\{ \log(h_0(t_i)) + ([\mu + \Gamma(w_i - \mu) + E(\epsilon_i)]^{\prime} \beta_x + z_i^{\prime} \beta_z) - H_0(t_i)exp([\mu + \Gamma(w_i - \mu) + \epsilon_i^{\prime} \beta_x + z_i^{\prime} \beta_z]) \right\}
\]

(4.19)
and \( f(\epsilon_i) \) is the marginal density function for measurement error. Computational difficulty arises from the integration \( \int f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) f(\epsilon_i) d\epsilon_i \), which includes the complex function \( f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) \). Since it is impossible to obtain the analytical form of \( f(\epsilon_i | t_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) \), the Markov Chain Monte Carlo (MCMC) integration is adopted. Specifically, a random walk Metropolis algorithm is used to generate the samples from the posterior distribution \( f(\epsilon_i | t_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) \). The advantage of using this algorithm is that it now only requires the calculation of \( f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) f(\epsilon_i) \) to generate the samples without the computation of \( \int f(t_i | \epsilon_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) f(\epsilon_i) d\epsilon_i \). Suppose at \( k \)th iteration, \( M \) samples are generated for each \( f(\epsilon_i | t_i, \hat{\beta}_x^{(k)}, \hat{\beta}_z^{(k)}) \), and are denoted as \( \epsilon^{(k)}_{i,1}, \epsilon^{(k)}_{i,2}, ..., \epsilon^{(k)}_{i,M} \). The MCMC integration can be achieved with these samples as

\[
E(\epsilon_i) = \frac{1}{M} \sum_{m=1}^{M} \epsilon^{(k)}_{i,m} 
\]

(4.20)

\[
E(\exp(\epsilon_i \beta_x)) = \frac{1}{M} \sum_{m=1}^{M} \exp([\epsilon^{(k)}_{i,m}]' \beta_x).
\]

(4.21)

3. Maximization of expected log-likelihood. Based on Eq.(4.20), (4.21) and (4.19), the expected log-likelihood value can be obtained. To maximize it with respect to \( \beta_x, \beta_z \), the profile likelihood approach is adopted. The maximization of Eq.(4.19) is converted to maximization of

\[
l^*_P(\beta_x, \beta_z) = \sum_{i=1}^{N} \{[\mu + \Gamma(w_i - \mu) + \frac{1}{M} \sum_{m=1}^{M} \epsilon^{(k)}_{i,m}]' \beta_x + z'_j \beta_z \}
\]

\[
- \log \sum_{j \in R(t_i)} \left[ \frac{1}{M} \sum_{m=1}^{M} \exp([\epsilon^{(k)}_{j,m}]' \beta_x) \exp([\mu + \Gamma(w_i - \mu)]' \beta_x + z'_j \beta_z) \right]
\]

The maximizer of \( l^*_P(\beta_x, \beta_z) \) becomes the updated estimate \( \hat{\beta}_x^{(k+1)} \) and \( \hat{\beta}_z^{(k+1)} \).
which are used to obtain updated $\hat{h}_0^{(k+1)}(t)$ as

$$
\hat{h}_0^{(k+1)}(t) = \frac{1}{\sum_{j \in R(t)} \left[ \frac{1}{M} \sum_{m=1}^{M} \exp([\mathbf{e}_{j,m}]^{(k)} \hat{\beta}_x^{(k+1)})] \exp([\mathbf{\mu} + \mathbf{\Gamma}(\mathbf{w}_i - \mathbf{\mu})]^{(k)}) \hat{\beta}_x^{(k+1)} + z_j^{(k+1)} \hat{\beta}_z^{(k+1)}\right]},
$$

and $\hat{H}_0^{(k+1)}(t)$ is obtained by plugging $\hat{h}_0^{(k+1)}(t)$ in Eq.(4.10). Given the $\hat{\beta}_x^{(k+1)}$, $\hat{\beta}_z^{(k+1)}$, $\hat{h}_0^{(k+1)}(t)$ and $\hat{H}_0^{(k+1)}(t)$, the next iteration of Monte Carlo E-step is performed with generation of new MCMC samples.

The algorithm iterates until the convergence of the estimates. The standard deviation of estimates can be obtained from the inverse of Hessian matrix of the likelihood function.

4.3 Simulation study

A simulation study is conducted to demonstrate the effectiveness of the proposed estimation approach. The conducted experiment compares the performance of proposed approach with the naive estimate ignoring errors under different settings for classical and Berkson error variance. In the simulation study, the hazard rate is assumed to depend on a continuous covariate $x_i, i = 1, 2, ..., 100$ with the following relationship

$$
\hat{h}(t_i) = h_0(t_i)exp(\beta x_i),
$$

where the baseline hazard rate function $h_0(t) = 0.05, \beta = 1$. The covariate $x_i$ consists of a mixture of 10 groups, and each group contains 10 true covariate values. A hundred failure time without censoring is collected corresponding to 100 covariate values from all 10 groups. The group means, denoted as $s_i$, follow normal distribution $N(\mu, \sigma_{bg}^2)$ with $\mu = 3, \sigma_{bg}^2 = 1$. The individual covariate is assumed $N(\mu, \sigma_{bg}^2 + \sigma_{wg}^2)$, where $\sigma_{wg}^2 = (\lambda \mu)^2$ with $\lambda$ being a predefined ratio.

To simulate the mixture of classical and Berkson error, it is assumed that the 100 true covariate values are not observed, only the corresponding group mean are mea-
sured as surrogates and used to approximate the individual covariate values. Thus, the Berkson error variance is equal to $\sigma_{wg}^2$. In addition, the classical errors exist in measuring the group means. The classical error follows $N(0, (0.01\mu)^2)$. The experiment runs 50 rounds of simulation with $\lambda = 0.05, 0.1, 0.2, 0.4$, respectively. For each $\lambda$, the performance of the proposed estimation approach is compared to naive estimation that directly uses observed erroneous covariates. The performance is measured by checking the unbiasedness of the estimates. The results are summarized in Table 4.1. The mean and standard deviation of estimates from the two estimation approaches in 50 rounds of simulation are recorded. Based on the collected estimates, a t-test is performed to check whether the estimates have population mean equal to the true coefficient value $\beta = 1$. As the table shows, when $\lambda = 0.05$, the $p$-values indicate that both the proposed approach and the naive estimates produce unbiased estimation outcome. However, as $\lambda$ grows to 0.1 or 0.2, the naive estimate fails to give unbiased estimate while the proposed approach outperforms by producing unbiased estimates. The corresponding $p$-values are highlighted with bold font. However, when the error variance grows too large ($\lambda = 0.4$), which corresponds to a Berkson error variance of 1.44 for covariates mean equal to 3, the performance of proposed approach deteriorates as well.

<table>
<thead>
<tr>
<th>Table 4.1: Performance comparison under different error variance settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>$\lambda = 0.05$</td>
</tr>
<tr>
<td>Stdev</td>
</tr>
<tr>
<td>p-value</td>
</tr>
<tr>
<td>$\lambda = 0.1$</td>
</tr>
<tr>
<td>Stdev</td>
</tr>
<tr>
<td>p-value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\lambda = 0.2$</th>
<th>Proposed method</th>
<th>Naive estimate</th>
<th>Proposed method</th>
<th>Naive estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda = 0.4$</td>
<td>0.9727199</td>
<td>0.9307209</td>
<td>0.1538688</td>
<td>0.1372857</td>
</tr>
<tr>
<td>Stdev</td>
<td>0.1284738</td>
<td>0.1238890</td>
<td>0.1538688</td>
<td>0.1372857</td>
</tr>
<tr>
<td>p-value</td>
<td><strong>0.3094</strong></td>
<td>5.289e-07</td>
<td>&lt;2.2e-16</td>
<td>&lt;2.2e-16</td>
</tr>
</tbody>
</table>
4.4 Conclusion

Effective prediction of time to failure in manufacturing systems provides invaluable
opportunity for operation supervisors to take preventive measures before equipment
fails. The Cox proportional hazard model, which has been introduced into failure
prediction application, assumes that covariates of the model are free of measurement
erors. This assumption, however, is often violated in many real world applications.
The errors that affect the model fitting can be categorized mainly into two groups:
classical error and Berkson error. Abundant approach has been proposed to reduce
the asymptotic bias of estimates when covariates contain classical error alone. For
scenarios where a mixture of classical and Berkson error occurs, no existing approach
has been proposed for Cox model parameter estimation. This chapter focuses on
the Cox model inference under such scenarios.

The proposed method links the true covariate with the observed erroneous mea-
surement using linear regression technique. Through representing the true covariate
as a function of observed values and latent error term, the maximum likelihood
estimate is obtained using an EM algorithm. The Markov chain Monte Carlo in-
tegration is adopted in the expectation step of the EM algorithm as the posterior
distribution of the latent error does not have an analytical form. A simulation study
is conducted to demonstrate the effectiveness of the proposed approach with small
to moderate level of Berkson error variance.

The proposed parametric method has distribution assumption for true covari-
ates, classical error and Berkson errors, all of which are normally distributed. The
nuisance parameters related to these normal distributions need to be given before-
hand or estimated directly from data. If these parameters have to be estimated
from the data, a simplifying assumption that allows separate estimation of these
parameters from the fitting of Cox model is needed. For future work, further in-
vestigation has to be conducted in quantifying the effect of such simplification. In
simulation study, it is noted that the proposed method fails to produce unbiased
estimate when the Berkson error variance becomes large ($\lambda = 0.4$). Thus, the other
future research direction is on improving the performance of the proposed approach in the large error variance case.
CHAPTER 5

CONCLUSIONS AND FUTURE RESEARCH

The research presented in this dissertation is concluded by summarizing the original contributions. Potential future studies are also discussed in this chapter.

5.1 Conclusion

The performance of a manufacturing system depends on its four elements: operator, machines, computer system and material handling system. To ensure the performance of these elements, operational data containing various aspects of information are collected for monitoring and analysis purpose. This dissertation focuses on the operator performance evaluation and machine failure prediction using collected operational data. This research work is motivated by the challenges in analyzing these data. These challenges include the complex relationship among the variables, the latent and implicit information of special interest to failure prediction, and the fact the data are often contaminated with outliers, missing and erroneous measurements. To address these challenges, this dissertation has made the following original contributions:

- To compare performance of different operators, a methodology is proposed based on the robust ZIP regression modeling and multiple comparisons technique. The methodology enables the comparison by quantifying and removing the complex of machines and other impacting factors on the performance metric using a regression model. The constructed regression model considers practically the distribution of NoD (number of defects). Since the number of defect will be strictly controlled to extremely low level, the NoD data often contain excessive number of zeros, which make the Poisson distribution inadequate to model the counts of NoD. Moreover, the NoD data may contain
outliers, whose values are significantly larger than that of majority of data. The outliers may be either valid values representing a true distribution of the variable of interest or invalid values as a result of certain mistakes, e.g., typos during recording the data. Robust analytical methods have to be developed for modeling data with these outliers. To reduce the impact of excessive zeros and outliers on model fitting, robust zero-inflated Poisson (ZIP) model is adopted. The Pearson residuals of the robust ZIP regression model are used as the independent inputs to the ANOMR/ANOMRV charts for performance comparison. As the Pearson residuals have non-normal distributions, existing methods such as ANOM/ANOMV chart cannot be applied. In stead, the ANOMR/ANOMRV charts are used for comparison of residuals grouped by operator ID. The estimated model coefficients are also used in multiple hypothesis tests to identify under-performing machines. This identification enables preventive maintenance to avoid further performance deterioration and reduce machine downtime. The identification of both under-performing operators and under-performing machines will help reduce NoD level and improve product quality and thus performance of the manufacturing system. The effectiveness of the methodology is demonstrated with a simulation and a real-world case study.

- To detect interval-based asynchronous periodic patterns (APPs) from operational data sequence, an algorithm combining a modified hierarchical clustering step and a convolution-based template matching step is proposed. The modified clustering step overcomes the drawback of existing pattern detection approaches, i.e., using subjective tolerance to address disturbance. It also enhances the interpretability of the extracted patterns with the added new terms in the clustering objective function. These added terms result in the extraction of longer and more patterns from the sequence. The clustering step effectively extracts APPs and reserves them in a symbol sequence. This step transforms the problem of detecting interval-based patterns into that of de-
detecting symbol-based periodic patterns, for which a variety of existing methods can be used. Compared with existing symbol-based periodic pattern detection method, the proposed one efficiently detects the patterns and allows a flexible mapping scheme. Both simulation and case study with real data have been conducted to demonstrate the effectiveness of the algorithm.

- To predict the machine failure based on the error-prone covariates, a new method is proposed for statistical inference of Cox proportional hazard model whose covariates have a mixture of classical and Berkson error. The method assumes normal distributions for the true covariates, the classical error and Berkson error. A linear regression model is constructed to link the true but unobserved covariates with observed erroneous ones. This linear regression model contains error terms whose distribution can be estimated or known from external data sources. The estimation of Cox model coefficients is achieved with an EM algorithm that treats the error terms as latent variables. The Markov Chain Monte Carlo simulation is used for calculating the conditional expectation with respect to error terms, whose posterior distributions have no closed form. The conducted simulation shows that compared to naive estimate, the proposed method achieves more accurate estimation under small to moderate level of error variance.

5.2 Future work

This dissertation presents the innovative research in addressing the problem of operator performance evaluation and failure prediction through the analysis of operational data. The proposed methods, while have been demonstrated effective through simulation or real-world case study, can be further improved in future research work. One limitation of the methodology proposed in the operator performance evaluation is the choice of parameter values, e.g. size of set $H_I$ in FAST-LTS algorithm. Currently the methodology adopts empirical values from existing literature. However, better parameter values may exist. A data-dependent selection method of these pa-
rameters will be desired. Thus, future research will focus on developing systematic methods of deriving optimal parameters. In addition, the proposed methodology does not fit the applications with small sample size. Further investigation will be also conducted on this topic.

For APP detection problem, the proposed algorithm uses a weighted combination of four terms as the objective function in the modified hierarchical clustering. These weights quantify the importance of each term and are currently set equal. Alternative solution to this multi-objective optimization problem is to use non-dominated solutions. The convolution-based pattern detection algorithm currently requires the period range of interest as an input. The default range is set to $[1, \lfloor g/2 \rfloor]$ for a sequence of length $g$. This means when an exhaustive search is conducted in a very large period range, the computational load will be intensive. Future work will focus on finding candidate periods to narrow down period range. In addition, the algorithm currently detects only exact repetition of symbols. Extended algorithm to detect patterns that contains symbol insertion and deletion will be developed in future.

For the developed Cox model inference method, the nuisance parameters related to distributions of true covariate, classical and Berkson error need to be given or estimated directly from data. A simplifying assumption that allows separate estimation of these parameters from the fitting of Cox model is currently used. More work can be conducted in quantifying the effect of such simplification. Simulation outcome also indicates that improving the performance of the proposed approach in the large error variance case is another research direction.
REFERENCES


