Advanced Alarm Monitoring Based on Alarm Data

by

Wenkai Hu

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Control Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Alarm systems are critical assets of modern industrial plants to assist operators in managing plant upsets and hazardous situations. A good alarm system must detect abnormalities and warn the operators promptly and yet at the same time not mislead, overload or distract the operators. However, in many actual cases, alarm systems often function poorly, and distract and overload operators due to nuisance alarms and alarm floods. Consequently, operators might be confounded by these problems, and be less alert to true alarms. To ensure process safety and improve efficiency of alarm systems, this thesis focuses on the development of advanced alarm management techniques, specifically using alarm data.

Four problems have been solved. First, a new method is developed to detect and quantify correlated alarms in the presence of occurrence delays, which are identified as the main causes leading to erroneous conclusions from existing methods. The effectiveness of the method is ensured by a statistical test based on the distribution of occurrence delays and a signal conversion mechanism of generating continuous-valued pseudo alarm sequences. Second, in order to assist prediction of alarm floods and prevention of their negative consequences, an accelerated local alignment method is proposed to find similar alarm flood sequences, which are very likely caused by the same root cause. To improve the computational efficiency and accuracy, three novel strategies are incorporated, including the priority-based similarity scoring strategy, the set-based pre-matching mechanism, and the modified seeding-extending steps. Third, to identify abnormality propagation paths and detect root causes, a causality inference method based on binary-valued alarm data is developed as an alternative in the absence of process data. A modified transfer entropy (TE) and a direct transfer entropy (DTE) are formulated based on the two characteristics of alarm signals, namely, the random occurrence delays and the mutual independence of alarm occurrences. Lastly, a data-driven technique is proposed to discover association rules of mode-dependent alarms from historical Alarm & Event (A&E) logs. The proposed method can help process engineers in discovering consequential nuisance alarms, and configuring state-based alarming strategies.

The effectiveness and applicability of the proposed methods are validated by case studies using real industrial alarm data. Using the proposed methods, interesting patterns can be extracted from the alarm data, and used to improve alarm monitoring by reducing nuisance alarms, addressing alarm floods, and tracking the propagation of abnormalities.

Preface

The research conducted in Chapters 2-4 of this thesis was part of an international research collaboration, with Dr. Jiandong Wang at Peking University, and research in Chapters 4-5 was part of collaboration, with Dr. Sirish L. Shah from the Department of Chemical and Materials Engineering at the University of Alberta. The ideas in Chapters 2 and 4 were from joint discussions with Dr. Jiandong Wang and Dr. Sirish L. Shah, respectively. The ideas in Chapters 3 and 5 were my original ideas. The algorithms, technical derivations, and case studies were my original work, as well as the introduction in Chapter 1.

- Chapter 2 has been published as Hu, W., Wang, J., & Chen, T., "A new method to detect and quantify correlated alarms with occurrence delays," *Computers & Chemical Engineering*, 80, 189-198, 2015.

- Chapter 3 has been published as Hu, W., Wang, J., & Chen, T., "A local alignment approach to similarity analysis of industrial alarm flood sequences," *Control Engineering Practice*, 55, 13-25, 2016. A short version has been published as Hu, W., Wang, J., & Chen, T., "Fast sequence alignment for comparing industrial alarm floods," In *Proceedings of 9th IFAC Symposium on Advanced Control of Chemical Processes*, Whistler, Canada, 48(8), 647-652, 2015.

- Chapter 4 has been submitted for publication as Hu, W., Wang, J., Chen, T., & Shah, S. L., "Cause and effect analysis of industrial alarm signals using modified transfer entropies," *Control Engineering Practice*, 2016.

- Chapter 5 has been submitted for publication as Hu, W., Chen, T., & Shah, S. L., "Discovering association rules of mode-dependent alarms from alarm and event logs," *IEEE Trans. Control Systems Technology*, 2016.

Acknowledgements

I would like to express my most sincere gratitude to those who have provided support during my Ph.D. program in the past four years. Most of all, I would like to thank my supervisor, Professor Tongwen Chen, for all his efforts and suggestions in helping as well as supervising me for my Ph.D. research. I have always been inspired by his academic achievements and supervision abilities, which have been driving me to concentrate on my research work and make continuous efforts to excel. It is from the University of Alberta, with the supervision of Professor Tongwen Chen, I have learned how to conduct research, write academic articles, and think in depth and with foresight.

I would also like to thank Dr. Xiang Chen (University of Windsor), Dr. Sirish L. Shah, Dr. Qing Zhao, and Dr. Mahdi Tavakoli for their comments and suggestions in the final defense of my Ph.D. degree. In addition, to complete the research work in this dissertation, some collaborators, including Dr. Sirish L. Shah and Dr. Jiandong Wang (Peking University), have also contributed their ideas and help. Thanks to their insightful suggestions, the research work in this dissertation was conducted smoothly and good results were achieved.

Further, I would also like to thank the current and former group members for their kindness and encouragement, with which I had an unforgettable life at the University of Alberta. Specifically, I would like to thank Ahmad W. Al-Dabbagh, Shiqi Lai, Muhammad S. Afzal, and Cen Guo (Tsinghua University), with whom I have some pieces of collaborative research work published in conference proceedings or in preparation for peer-reviewed journal articles.

At last, I would like to thank all my family members, especially my parents and my wife for their love and support. My parents have always been doing whatever they can to support me without reservation since I came to the world, so that I am able to have a good education and the chance to conduct my Ph.D. research. Meanwhile, I would like to express my special gratitude to my wife for accompanying me in experiencing the cold winters in Edmonton and countless happy moments. Just one day before my final Ph.D. defense, my wife and I received a lovely and healthy son, who is the root cause of my busy life at the end of my Ph.D. program and also the reason of my deep happiness. Thus, these moments have lots of meanings to me, because of the upgrading of my role, not just in academia, but also in life.

In the past four years, I have really gained a lot, including research abilities, friendship, love, and fruit of love. Again, I would like to thank all my family members, teachers, and friends for their love, kindness and support.

Contents

1 Introduction 1

1.1 Background 1

1.1.1 Alarm Systems and Alarm Data 1

1.1.2 Current Status of Alarm Management 4

1.2 Literature Survey 6

1.2.1 Methods to Minimize Nuisance Alarms 6

1.2.2 Techniques to Analyze and Inhibit Alarm Floods . . . 8

1.2.3 Causality Inference for Complex Chemical Processes . . 10

1.3 Thesis Contributions 12

1.4 Thesis Organization 13

2 Detection and Quantification of Correlated Alarms with Occurrence Delays 15

2.1 Overview 15

2.2 Occurrence Delays 16

2.2.1 Alarm Signals 16

2.2.2 Definition of Occurrence Delays 17

2.2.3 Mechanism to Generate Pseudo Alarm Signals 21

2.3 Detection of Correlated Alarms 24

2.3.1 Estimation of the Correlation Delay 24

2.3.2 Statistical Test 26

2.3.3 Detection Procedure 30

2.4 Industrial Case Studies 32

2.4.1 Case I: Correlated Alarms 32

2.4.2 Case II: Uncorrelated Alarms 35

2.5 Summary 37

3 Similarity Analysis of Industrial Alarm Flood Sequences 38

3.1 Overview 38

3.2 Similarity Scores for Alarm Floods 39

3.2.1 Representations of Alarm Floods 39

3.2.2 Alarm Priority-Based Similarity Scoring Strategy . . . 41

3.3 Local Alignment of Alarm Floods 42

3.3.1 Main Steps of the BLAST 43

3.3.2 Set-Based Pre-Matching Mechanism 44

3.3.3 The Seeding Step 46

3.3.4 The Extending and Backtracking Steps 48

3.3.5 Time Ambiguity Tolerance 53

3.3.6 The Steps of the Local Alignment Algorithm 55

3.3.7 Parameter Determination 57

3.4 Industrial Case Studies 59

3.4.1 Case I 59

3.4.2 Case II 61

3.5 Summary 64

4 Cause and Effect Analysis of Industrial Alarm Signals Using Modified Transfer Entropies 66

4.1 Overview 66

4.2 Transfer Entropy of Alarm Data 67

4.2.1 Transfer Entropy of Alarm Signals 67

4.2.2 Detection of Direct Causality 72

4.3 Causality Inference 73

4.3.1 Significance Test 73

4.3.2 Number of Alarm Occurrences 77

4.3.3 Causality Inference Based on NTEs 79

4.3.4 Detection of Direct Causality Based on NDTEs 81

4.4 Case Studies 83

4.4.1 Case I: Numerical Case Study 83

4.4.2 Case II: Industrial Case Study 86

4.5 Summary 89

5 Discovering Association Rules of Mode-Dependent Alarms from Alarm and Event Logs 91

5.1 Overview 91

5.2 Preliminaries on Mode Dependent Alarms 92

5.2.1 State-Based Alarming in Practice 92

5.2.2 Operating Modes and Consequential Alarms 93

5.3 Problem Formulation 95

5.3.1 Mathematical Definitions 95

5.3.2 Problem Formulation 97

5.4 Methodology 99

5.4.1 Frequent Patterns of Operating Modes 99

5.4.2 Mode-Dependent Alarms for Single Operating Mode . 103

5.4.3 Mode-Dependent Alarms for Multiple Operating Modes 105

5.4.4 Detection Procedures 107

5.5 Industrial Case Studies 108

5.5.1 Case Study I 109

5.5.2 Case Study II 111

5.6 Summary 116

6 Conclusions and Future Work 117

6.1 Conclusions 117

6.2 Future Work 118

Bibliography 121

List of Tables

1.1 An example of a structured A&E log. 3

2.1 Calculated correlation statistics for Example 2b. 24

2.2 Estimation of the correlation delay using different approaches for Example 3. 26

3.1 An industrial example of an alarm flood sequence. 40

3.2 Alarm priorities in alarm systems. 41

3.3 Basic similarity scores 42

3.4 Alarm floods X and Y with time and priority information. . . 46

3.5 Basic components of different sequences. 47

3.6 The matched alarm pairs. 48

3.7 Extending score matrix H in the forward extension based on the seed $\tilde{Z}(1) =< 2, 1, 3, 4 >$. 52

3.8 Extending score matrix H in the backward extension based on the seed $\tilde{Z}(1) =< 2, 1, 3, 4 >$. 52

3.9 Extending score matrix H of $\tilde{Z}(1)$ in the forward extension with time stamp information. 54

3.10 The alignment accuracy measured by the average number of the matched pairs $\bar{C}(X, Y)$ and the total computation time T in querying $\mathbb{B}$ for X_i $(i = 1, ..., 10)$. $L(X_i)$ indicates the number of alarms contained in X_i. $N(Y)$ denotes the number of relevant alarm floods Y found in $\mathbb{B}$. 61

4.1 Normalized transfer entropies (From row to column). 84

4.2 Normalized transfer entropies (From row to column). 88

4.3 Detection of direct causalities based on NDTEs. 89

5.1 Operating modes of some commonly used equipment. 94

5.2 Number of frequent mode patterns based on different F_{th}. 111
5.3 Number of association rules based on different F_{th}. 111
5.4 Association rules detected for Mode #68. 113

List of Figures

1.1 Alarm system dataflow [47]. 2

1.2 The cause and effect analysis for industrial processes via different resources. 10

2.1 Process signals in Example 1: (a) $x_p(t)$ (solid) with x_{tp} (dash), (b) $y_p(t)$ (solid) with y_{tp} (dash) for the case that $\omega(t)$ is absent, (c) $y_p(t)$ (solid) with y_{tp} (dash) for the case that $\omega(t)$ is present. 19

2.2 Alarm signals in Example 1: (a) $x_a(t)$, (b) $y_a(t)$ for the case that $\omega(t)$ is absent, (c) $y_a(t)$ for the case that $\omega(t)$ is present, (d) $y'_a(t)$ for the case that $\omega(t)$ is present 20

2.3 The histograms of occurrence delays without (a) and with (b) the presence of $\omega(t)$. 20

2.4 Example 2a: (a) $x_a(t)$ and $y_a(t)$, (b) $x_g(t)$ and $y_g(t)$. 23

2.5 Histogram of φ in 1000 Monte Carlo simulations for uncorrelated alarms with $p_r = 0.005$ and $q_r = 360$. 29

2.6 μ_φ as a function of p_r and q_r. 30

2.7 σ_φ as a function of p_r and q_r. 31

2.8 Flowchart for detecting correlated alarms. 32

2.9 Samples of $x_a(t)$ (top) and $y_a(t)$ (bottom) for Case I. 33

2.10 The sequence of ROD (top) and the histogram of ROD (bottom) for Case I. 34

2.11 The estimated Sorgenfrei coefficients (top) and Jaccard coefficients (bottom) for different values of τ for Case I. 34

2.12 Samples of $x_a(t)$ (top) and $y_a(t)$ (bottom) for Case II. 35

2.13 The sequence of ROD (top) and the histogram of ROD (bottom) for Case II. 36

2.14 The estimated Sorgenfrei coefficients (top) and Jaccard coefficients (bottom) for different values of τ for Case II. 36

3.1 The calculation of the extending score matrix H in the gapped extension. The circled numbers 1, 2, and 3 indicate the first, second, and third ways to calculate the element $H_{i,j}$ associated with eqns. (3.18), (3.19), and (3.20). 49
3.2 Graphic illustration of the gapped extension. 52
3.3 Flowchart of the proposed local alignment algorithm. 56
3.4 Average number of matched pairs $\bar{C}(X, Y)$ versus K and U. . 58
3.5 Average computation time $\bar{T}$ versus K and U. 59
3.6 Simplified schematic diagram of the oil conversion plant. . . . 60
3.7 Performance statistics of the proposed local alignment algorithm (dot) and the modified SW algorithm (circle-dash) for the query of X to 125 object alarm floods: (a) the number of matched pairs; (b) the computation time. 62
3.8 Simplified schematic diagram of the hydrogen heater (the dashed lines indicate data links). 63
3.9 The best sequence alignment between the query alarm flood X and the 117th Y from $\mathbb{B}$. 64

4.1 A numerical example for predicting Y based on the history of X under the influence of random occurrence delays. 69
4.2 Distribution of NTEs calculated from 10,000 pairs of surrogate alarm signals X^s and Y^s: (a) the histogram is skewed right and multimodal; (b) the normal probability plot deviates from the red straight line. 74
4.3 Significance thresholds γ_T of NTEs versus K in 1000 Monte Carlo simulations. 76
4.4 Results of 1000 Monte Carlo simulations for the case without causal relation: (a) NTEs (blue line) and corresponding significance thresholds (red line); (b) correctness of results ("True" for correct estimate while "False" for incorrect estimate). . . . 77
4.5 An example of correlated alarms and the distribution of random occurrence delays: a) the alarm signal X, c) the alarm signal Y, b) the occurrence delays, and d) the histogram of occurrence delays. 78
4.6 Average NTEs v.s. numbers of alarm occurrences for different p. 79
4.7 Flowchart for the causality inference based on NTEs. 80

4.8 Flowchart for the detection of direct causalities based on the NDTEs. 82
4.9 Alarm occurrence signals: (a) the alarm signal X_1; (b) the alarm signal X_2; (c) the alarm signal X_3. 84
4.10 NTEs versus time lags. 85
4.11 Causal maps: (a) the causal map of all information flow paths; (b) the causal map of direct information flow paths 85
4.12 Alarm occurrence signals over 3.5 days time period. 86
4.13 Trends of transfer entropies versus time lags. 87
4.14 Causal map of information flow paths. 88
4.15 Causal map of direct information flow paths. 89

5.1 Sketch of a lube oil pump (PI is an indicator and the pump is controlled by an operator via a DCS (Distributed Control System)). 92
5.2 Switching between "Running" and "Closure" modes of a lube oil pump (blue line) and the transition of alarm states (dashed red line). Two boxed plots show zoomed details when the pump started or stopped, and the corresponding alarm states. . . . 93
5.3 An example of an operating mode variable (a) and an alarm occurrence variable (b). 96
5.4 Example of the mode-dependent alarm on a time axis. 98
5.5 Example of mode-dependent alarms of multiple modes on a time axis. 98
5.6 Graphical example for producing the time vector of the multimode variable $\tilde{X}^{(2)} = \{x_1, x_2\}$ based on $\mathbf{t}^{(x_1)}$ and $\mathbf{t}^{(x_2)}$. 106
5.7 Flow chart for the procedures to determine association rules of mode-dependent alarms. 109
5.8 The time difference for each event of operating mode and the nearest alarm occurrence following it. 110
5.9 The mean of Chi-squared values for all association rules based on different F_{th}. 112
5.10 The frequencies $\xi(x_i \to y_j)$ and Chi-squared values Λ for the top four hundred alarms and "Mode #68". (The dashed lines in the upper plot and lower plot indicate the frequency threshold $F_{th} = 10$ and the Chi-square threshold 3.84, respectively.) . . . 112

5.11 Occurrences of “Mode #68” and the five alarms on the time axis. (The vertical lines indicate the time instants when the mode or alarm occurred.) 113
5.12 Time differences between the occurrences of each mode-alarm pair. 114
5.13 Occurrences of “Mode #161”, “Mode #179”, “Mode #188”, and “Alarm #293” on the time axis. 115
5.14 Time differences between the occurrence of “Alarm #293” and the co-occurrence of “Modes #161”, “Modes #179”, and “Modes #188”. 115

List of Symbols

S_{jacc}	Jaccard Coefficient
S_{sorg}	Sorgenfrei Coefficient
r_{xy}	Correlation Between Two Alarm Variables
λ	Occurrence Delay
λ_r	Real Occurrence Delay
φ	Lag Factor
$< x_1, x_2, ..., x_M >$	Alarm Flood Sequence
$\phi(p_l)$	Match Score for Priority p_l
ζ	Mismatch Score
δ	Gap Penalty
Σ	Alphabet of Unique Alarms
$S_{\text{set}}(X, Y)$	Set-Based Similarity Index Between Alarm Floods
$S(\tilde{X}, \tilde{Y})$	Similarity Score Between Sequences $\tilde{X}$ and $\tilde{Y}$
$T_{X \to Y}$	Transfer Entropy (TE)
$NT_{X \to Y}$	Normalized Transfer Entropy (NTE)
$D_{X \to Y \mid Z}$	Direct Transfer Entropy (DTE)
$ND_{X \to Y \mid Z}$	Normalized Direct Transfer Entropy (NSTE)
γ_T	Significance Threshold of NTE
γ_D	Significance Threshold of NDTE

$\mathbb{D}$	Event Database
$\mathbb{N}^+$	Domain of Natural Numbers
$\xi(x_i \to y_j)$	Frequency of x_i Being Followed by y_j
$\mathcal{L}$	Collection of Frequent Mode Patterns
Λ	Logarithmic Likelihood Statistic

List of Acronyms

A&E	Alarm & Event
AAD	Averaged Alarm Delay
ALM	Alarm Occurrences
BLAST	Basic Local Alignment Search Tool
BPCS	Basic Process Control System
DCS	Distributed Control System
DTE	Direct Transfer Entropy
DTW	Dynamic Time Warping
EEMUA	Engineering Equipment and Materials Users' Association
FAR	False Alarm Rate
HMI	Human-Machine Interface
HSP	High Scoring Segment Pair
IEC	International Electrotechnical Commission
IQR	Inter-Quartile Range
ISA	International Society of Automation
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MAR	Missed Alarm Rate
NDTE	Normalized Direct Transfer Entropy
NTE	Normalized Transfer Entropy

ODE	Occurrence Delay Based Estimation
P&ID	Piping and Instrumentation Diagram
ROD	Real Occurrence Delay
RTN	Return-To-Normal
SCADA	Supervisory Control and Data Acquisition System
SDG	Signed Directed Graph
SIS	Safety Instrumented System
SW	Smith-Waterman
TCD	Temporal Causal Diagrams
TE	Transfer Entropy
TFPG	Timed Failure Propagation Graphs

Chapter 1

Introduction

1.1 Background

Modern large-scale industrial facilities, e.g., oil refineries, pharmaceutical factories, and power plants, usually consist of a large number of actuating elements, metering devices, measuring transducers, and control modules, which are tightly interconnected. Abnormalities in such integrated processes usually propagate through plant devices, causing enormous losses, such as out-of-spec product quality, process upsets, and even serious environmental releases or human injuries. In view of this and to be in regulatory compliance with respect to safe process operation, the management issues of abnormal situations via process monitoring, fault diagnosis, and alarm management have been receiving increasing attentions from both academia and industrial communities [24, 60, 61, 96]. To assist operators in perceiving near misses and managing hazardous situations, alarm systems are commonly used to monitor process measurement, equipment status, and safety conditions. To design, implement, and maintain industrial alarm systems, standards and guidelines, such as EEMUA-191 [28], ISA-18.2 [47], and IEC-62682 [46], have been developed and widely accepted by industrial practitioners. This section provides the background of industrial alarm systems, and the current status of alarm management.

1.1.1 Alarm Systems and Alarm Data

Alarm systems are critical assets of process industries to notify operators of abnormal process conditions or equipment malfunctions. In the days of hardwired controls and alarms, the installation of new alarms was expensive due to the high cost of hardware and limited space on the control wall [38, 85]. Over the past decades, with the progress of digital technologies, the Distributed Control Systems (DCS) and the Supervisory Control and Data Acquisition (SCADA)systems have gradually predominated. As a result, the configuration of control strategies and the installation of alarms have become effortless tasks. Specifically for alarms, the computerized scrolling list has replaced the lightboxes, and the installation of alarms can be easily conducted using softwares. A typical modern alarm system consists of three major components, as shown in Fig. 1.1 [47]: (1) The Basic Process Control System (BPCS) and the Safety Instrumented System (SIS) generates alarms based on the measurements of process conditions and certain logic; (2) the Human-Machine Interface (HMI), such as a computer screen or an annunciator panel, delivers alarm information to operators; (3) the alarm log formats and stores the alarm messages and other related events. The advanced alarm applications, external systems, and alarm historian are also important for an effective alarm system.

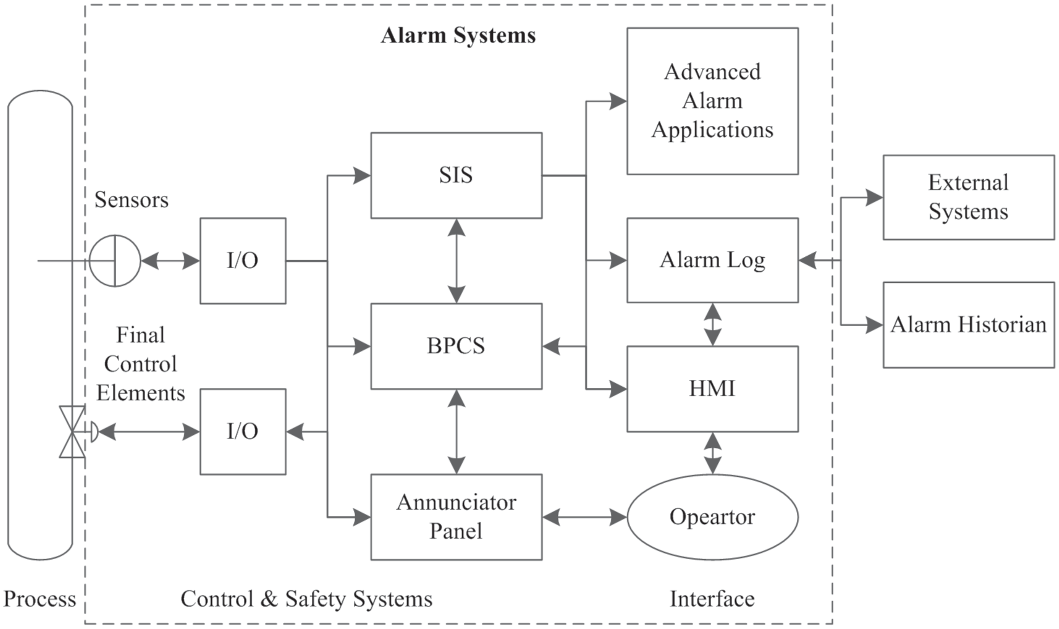


Figure 1.1: Alarm system dataflow [47].

In a typical alarm system, the alarm signals are generated as visible and/or audible notifications to indicate equipment malfunctions, process deviations, or abnormal conditions. For the sake of offline examination or analysis, alarm signals are usually formatted as textural data and stored in an Alarm & Event (A&E) log. Generally, the alarm data in an A&E log consists of two parts: the configuration attributes, e.g., the tag name, alarm type, priority and location, and the realtime messages, e.g., alarm occurrences, return-to-normal instants, and their time stamps [38, 49, 57]. In addition to the alarm messages, the operator related events, such as the operator acknowledgements, adjustment of controllers, and changes to alarms are also available in many A&E logs. An example of a structured A&E log in an oil plant is shown in Table 1.1.

Table 1.1: An example of a structured A&E log.

Time stamp	Type	Tag	Identifier	State	Priority	Location	Description
26/02/2016 08:17:23	Alarm	F1A2	CFN	ALM	High	Unit A	The status of the valve for feed gas supply
26/02/2016 08:17:41	Alarm	P2C7	PVBAD	ALM	Low	Unit A	Exceeding the range of measuring instrument
26/02/2016 08:18:05	Alarm	P5C1	CFN	ALM	High	Unit A	The status of the calculated variable for product 1
26/02/2016 09:33:39	Alarm	L2V2	PVHI	ALM	Critical	Unit B	The water level is above its high alarm limit
26/02/2016 10:41:33	Alarm	F1A2	CFN	RTN	High	Unit A	The status of the valve for feed gas supply
26/02/2016 10:49:52	Change	VALV2		OPEN		Unit B	Open a drainage valve at the outlet of a tank
26/02/2016 10:55:17	Alarm	L2V2	PVHI	RTN	Critical	Unit B	The water level is below its high alarm limit
26/02/2016 11:34:05	Alarm	P2C7	PVBAD	RTN	Low	Unit A	Within the range of measuring instrument
26/02/2016 12:21:01	Alarm	P5C1	CFN	RTN	High	Unit A	The status of the calculated variable for product 1
26/02/2016 14:30:12	Change	PUMP1		STOP		Unit C	Stop a pump that supplies lube oil
26/02/2016 14:31:33	Alarm	PI01	PVLO	ALM	Low	Unit C	The discharge pressure is below its low alarm limit
26/02/2016 14:32:15	Alarm	FI01	PVLO	ALM	Low	Unit C	The discharge flow is below its low alarm limit
26/02/2016 19:55:12	Change	PUMP1		START		Unit C	Start a pump that supplies lube oil
26/02/2016 19:55:50	Alarm	FI01	PVLO	RTN	Low	Unit C	The discharge flow is above its low alarm limit
26/02/2016 19:56:17	Alarm	PI01	PVLO	RTN	Low	Unit C	The discharge pressure is above its low alarm limit

The header of each column represents an attribute while each row records the textual messages along with time. “Time stamp” indicates when an event occurs. “Type” indicates whether the event is an alarm or an operator action. “Tag” is a name allotted to label a specific device or process. “Identifier” represents the alarm type, e.g. CFN (changing from normal), PVLO (exceeding the low alarm limit), and PVBAD (out of measuring range). In the “State” column, we can know whether an alarm occurs (ALM) or returns to normal (RTN) and what actions the operators have made. “Priority” represents the relative importance assigned to an alarm, and “Area” denotes the physical location. For instance, “F1A2” indicates the gas flow and “CFN” is an applicable identifer of “F1A2”; they jointly make a unique alarm tag “F1A2.CFN” that is located in Unit A and configured with a high priority. The alarm is binary-valued with two states, namely, “ALM” and “RTN”. “F1A2.CFN” was annunciated at 08:17:23 and returned to normal at 10:41:33 on 26/02/2015. Moreover, the interactions between alarms and operator actions can be observed from the A&E log. For instance, the alarm “L2V2.PVHI” returned to normal after the drainage valve “VALV2” was opened. In Unit C, it looks like that the low pressure alarm “PI01.Low” and low flow rate alarm “FI01.Low” were annunciated after pump labeled as “PUMP1” stopped, and returned to normal subsequently after it restarted.

The A&E log is an important resource that enables the offline analysis. Accordingly, interesting patterns can be discovered from historical data and used for the online purpose, so as to help operators with alarm monitoring. The work in this thesis is purely alarm data based.

1.1.2 Current Status of Alarm Management

A good alarm system must promptly detect abnormalities and warn the operators promptly while does not mislead, overload or distract the operators [35, 78]. However, in many actual cases, alarm systems often poorly function and suffer from common major problems, such as nuisance alarms, standing alarms, and alarm floods [13, 38, 87]. Consequently, operators might be confounded by these problems, and be less alert to true alarms, which may compromise process safety and production efficiency. The standards and guidelines [28, 46, 47] suggest that the acceptable average alarm rate and peak alarm rate should be no more than 1 and 10 over a 10 min period for each operator, respectively. However, the reality of alarm management in process industries is far worse than these benchmarks as revealed in [78, 85]. The number of alarms presented to operators is often much higher than that can be effectively managed, and most of these alarms are of little value to operators [13, 35, 73]. For example, the average alarm rate per day in a European refinery was as high as 14,250 [64]. A survey in [12] reported average alarm rates ranged from 2 to 33 over a 10 min period in different plants. Consequently, operators were distracted from critical alarms, and this was identified as the leading cause of many accidents.

Nuisance alarms, such as chattering alarms, fleeting alarms, standing alarms, and redundant alarms have been identified as the main problems contributing to the inefficiency of alarm systems. Chattering alarms are typical nuisance alarms caused by measurement noises and disturbances [28]. They make fast transitions between alarm and non-alarm states in a short span of time without operators’ response. An alarm that repeats three or more times per minute is regarded as the worst case [47]. A fleeting alarm is regarded as a sub-category of nuisance alarms similar to chattering [38]; it appears and clears too quickly for the operator to respond, but does not necessarily repeat. ISA-18.2 [47] declares that at the rationalization stage, alarm justification and prioritization should ensure that an alarm does not duplicate another alarm that is designed for the same abnormality. However, massive interacting components make redundancy almost inevitable. As a result, alarms could be either redundant or highly overlapping in indicating the presence of the same abnormality [78]. The monitoring burden of operators is increased due to redundant alarm annunciators; these redundant alarms should be detected and suppressed. Meanwhile, related alarms could be grouped and presented to operators as symptoms to detect abnormalities [78]. Therefore, the detection and quantification of correlated alarms are useful in improving alarm management.

Another major culprit overloading operators is the alarm flood, which refers to a situation during which the alarm rate is too high, exceeding the ability of industrial plant operators to manage occurred alarms in a prompt manner. The industrial standard ANSI/ISA-18.2 [47] says that an alarm flood begins when 10 or more alarms occur within 10-minute time period until the alarm rate drops below five alarm occurrences in 10 min. Alarm floods could be caused by many factors, e.g., abnormal situations, improper alarm system design, and operating state transitions [38]. In practice, alarm floods should be limited to less than 1% of the total time period that an industrial alarm system is in operation [28]. However, alarm floods are quite common in the existing alarm systems [9]. In the presence of alarm floods, a large amount of annunciated alarms may not be manageable by operators. As a result, critical alarms may be overlooked, and various negative consequences could arise due to lack of responses to these critical alarms. The consequences include making a dangerous process situation much worse, increasing the risk of process upsets, and even deteriorating the system performance, and leading to serious accidents. Moreover, when the alarm rate is too high, operators have no choice but to ignore many of the annunciated alarms. In this case, the designed functionality of alarm systems is partially or even completely lost. Thus, the equation "floods = incidents = loss" is likely to be valid [9]. As an example, 275 alarms had to be handled by two operators during the 10.7

minutes before the explosion accident occurred in the Texaco Refinery at Milford Haven [40]; apparently, the operators failed in handling these alarms due to such high alarm rate. Hollifield & Habibi [38] (page 75) reported that the average number of alarm floods per day in 8 weeks was 14.6, and the percentage of time for alarm systems staying at an alarm flood situation was 50.3%. In such situations, alarms burst out within a short period, making operators distracted and confused; thereby, wrong actions were taken and critical alarms were missed.

Motivated by above problems of industrial alarm management, the work in this thesis targets at developing advanced techniques based on alarm data, to reduce nuisance alarms, address alarm floods, track the propagation of abnormalities, and eventually avoid alarm overloading and provide operators with decision supports.

1.2 Literature Survey

This section presents a detailed literature survey on recent advances of alarm management techniques for tackling nuisance alarms, analyzing alarm floods, and detecting causal relations.

1.2.1 Methods to Minimize Nuisance Alarms

As revealed in [93], nuisance alarms are major culprits for alarm overloading. To reduce nuisance alarms and mitigate alarm overloading, a variety of alarm reduction techniques have been proposed, e.g., the combined use of plant connectivity and alarm logs [79], the filtering of false alarms using a dynamic fault tree [83], a proactive alarm reduction system [50], a distributed parallel alarm management strategy [64], and an alarm analytic tool [45]. To evaluate the performance of alarm systems, the false alarm rate (FAR), missed alarm rate (MAR) and averaged alarm delay (AAD) are proposed in [1, 95]. The first two metrics measure the accuracy in detecting normal and abnormal conditions, and the AAD denotes the alarm latency or promptness. To minimize FAR and MAR, filters, deadbands and delay-timers are most commonly used tools [18, 38]. Good alarm configuration and efficient alarm system design are the preferential steps in preventing nuisance alarms. To deal with chattering alarms, redundant alarms, and consequential alarms, a variety of more advanced techniques have been proposed. A quantitative measure based on run length distributions was introduced by Naghoosi et al. [69] and Kondaveeti et al. [58] to detect chattering alarms. The removal of chattering alarms can be easily achieved using delay timers [58, 91]. To deal with chattering alarms caused by oscillating processes, Wang & Chen [90] presented an online detection and removal approach.

In addition, correlated alarms could be overlapping in indicating the same abnormality [78]. To detect correlated alarms and quantify their similarities, some alarm data based techniques have been developed. Yang et al. [98] analyzed the difference between the cross correlation of process signals and that of alarm signals, and combined the results of cross correlation with process connectivity information to design alarm limits. Noda et al. [74] proposed a scheme called event correlation analysis to quantify the relationship between alarms and operating actions, and to identify correlated alarms and unnecessary operating actions. Kondaveeti et al. [57] grouped similar alarms together based on Jaccard similarity coefficients in an alarm similarity map. Yang et al. [99] transformed binary alarm sequences to continuous-valued pseudo alarm sequences and clustered correlated alarms based on the cross correlation of pseudo alarm sequences. Yang et al. [100] compared different similarity coefficients for binary-valued alarm signals, and detected correlated alarms based on the Sorgenfrei coefficient and the distribution of correlation delays. As a result, redundant alarms can be eliminated and highly correlated alarms be grouped.

In addition to the above techniques for reducing nuisance alarms, the logic based alarming techniques, such as the dynamic suppression and state-based alarming, are also effective in improving the performance of alarm systems and prevalent in practice [28, 38, 47]. It is not uncommon to see that a process or a piece of equipment has several operating states, as evident during the startup and shutdown of a unit, the partial, full or closure of a device (such as a pump or a compressor), and the synthesis of different products [38]. In such cases, the alarms can be minimized by dynamically modifying alarm parameters or suppressing alarms based on certain conditions [51, 75, 102]. In such cases, state-based alarming is being used more and more frequently in practice and could be a critical technique in control systems of the next generation as indicated by Jerhotova et al. [51]. According to a test implemented in a nuclear power plant simulator, a state-based alarm system showed high usability in helping operators [71]. Bhaumik et al. [10] discussed how a mode-based solution was deployed in an oil sand extraction plant for effective alarm management, which kept unnecessary alarms out of annunciators under some specific operating states. In addition to this, Hollifield & Habibi [38] and Wang et al. [93] also reported successful applications of state-based alarming in chemical plants and power units.

1.2.2 Techniques to Analyze and Inhibit Alarm Floods

In order to reduce the occurrence of alarm floods or alleviate the severity of alarm floods, much of recent research work has been carried out on alarm system rationalization. Plant connectivity and alarm logs were combined to reduce the number of alarms by grouping alarm messages associated with a common root cause [79]. Based on the dependencies between fault events and the precedence of alarm messages, a dynamic fault tree was developed to generate filtering rules for false alarms [83]. To remove alarm floods or to mitigate their effects, two advanced alarm handling techniques were presented: the state-based alarming reduces alarm messages by suppressing alarms for different process states and the alarm load shedding strategy displays only the most critical alarms during alarm floods [51]. To avoid alarm floods during transition of operating states, an alarm management strategy was provided by dynamically changing alarm limits based on a priori knowledge of transitions [102].

Because an alarm flood is usually the result of a primary event and its consequential events [87], an alarm flood is often composed of a series of consequential alarms. Thus, the guideline EEMUA-191 [13, 28] recommended to group consequential alarms to reduce the number of alarms during alarm floods. To avoid alarm flooding and find out root alarms, Wang et al. [92] proposed a method to identify consequential alarms with the combination of alarm similarity analysis and process variable causality inference. Based on the concept of frequent pattern mining, a criteria-based alarm flood pattern recognition method was utilized to reduce alarm overloading during alarm floods by identifying alarm sequences of causally dependent notifications and displaying them as a single piece of information [88]. Most-frequent alarm sequences and causal alarms consolidating the alarm sequences were identified to redesign alarm systems for reducing alarm floods [29, 30]. Based on the pattern mining techniques, sequential alarms were found and used for alarm rationalization to remove redundant alarms, identify bad actors, and establish an effective alarm system [20, 59]. To help operators to concentrate on most important alarms during alarm floods, an alarm prioritization system was p-resented to prioritize alarms by calculating the severity of each alarm based on fuzzy-logic rules [31]. Some new alarm presentation techniques were proposed to help operators in understanding the alarm floods by showing alarms in time series together with short alarm descriptions [62]. Towards improving operator responses during alarm floods, Bullemer et al. [14] developed alternative alarm presentation techniques to enable operators to strategically view subsets of alarms rather than the whole list.

Recently, the prediction and prevention of alarm floods have been drawing more attentions. An abnormal event usually causes a series of alarms in a chronological order, and the same type of abnormalities usually lead to similar alarm flood sequences. Thus, by comparing an incoming alarm sequence with potentially similar alarm floods in the historical database, it is possible to achieve an early warning of abnormalities, predict the cause of the incoming alarm flood, and take proactive operational actions to prevent the occurrence of an alarm flood and its negative consequences. To the best of our knowledge, the contemporary studies on the prediction and prevention of alarm floods are limited to the very first step of finding similar alarm flood sequences, which are different from above-mentioned extracting consequential alarms. Ahmed et al. [3] exploited a dynamic time warping (DTW) algorithm to find common sequences among alarm floods. Cheng et al. [17] developed a modified Smith-Waterman (SW) algorithm for local sequence alignment of alarm floods with incorporation of time stamp information. In addition, Charbonnier et al. [15] proposed a pattern-matching approach based on alarm lists rather than sequences. By incorporating the orders of alarms, Charbonnier et al. [16] utilized the Needleman-Wunsch algorithm to compare alarm floods and achieved fault isolation by extracting fault templates.

1.2.3 Causality Inference for Complex Chemical Processes

The detection of abnormality root causes is challenging unless the propagation of plant-wide abnormalities through a complex web of interconnected material and information flow paths is explicitly known. The cause-effect reasoning provides an effective way to investigate abnormality propagation paths [24]. Various techniques for cause and effect analysis of industrial processes have been studied [7, 8, 11, 19, 25, 26, 68, 86, 96, 97]. These techniques can be categorized into graph-based methods and data-driven approaches, as depicted in Fig. 1.2. Once an incident, disturbance, or oscillation occurs and propagates through process units, the task of cause-effect reasoning is to discover a causal map and find the root causes.

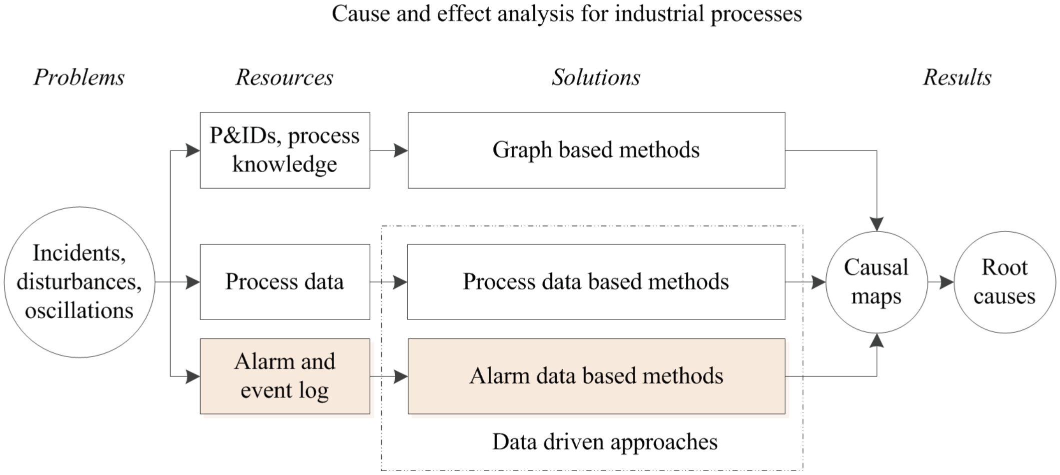


Figure 1.2: The cause and effect analysis for industrial processes via different resources.

Graphical causality inference has different forms of graphic models, such as structural graphs, qualitative graphs, and causal probabilistic graphs [11]. In [96], a qualitative fault detection and hazard analysis method was proposed based on the Signed Directed Graph (SDG), which is a typical type of qualitative graphical model describing process variables and their causal relations by nodes and arcs, respectively. In [22], a Bayesian method was proposed for the induction of probabilistic networks that provide insights into probabilistic dependencies among multiple variables. To diagnose problems associated with the power transmission lines, Mahadevan et al. [65] developed a Temporal Causal Diagrams (TCD) and a refinement of the Timed Failure Propagation

Graphs (TFPG) to represent faults and their propagation paths. For the monitoring and fault isolation of complex dynamic systems, Mosterman et al. [67] exploited parsimonious topological models to analyze the transient behavior in response to faults in a qualitative framework. The graph-based methods share some common advantages, e.g., they are easily implemented and interpreted. However, the establishment of graphic models requires the availability of an up-to-date Piping and Instrumentation Diagram (P&ID), process knowledge, and engineering experience [97]. The collection of such information is time and resource intensive, making the detection of root causes inefficient, especially for large scale systems [25, 96].

In contrast, data driven methods are effective for many cases [77]. The existing data driven methods for causality inference are mainly based on various metrics such as the Granger causality [34] and the transfer entropy (TE) [8] obtained from historical data samples of process variables [7, 8, 19, 25, 26, 68, 86]. The modified Kullback-Leibner information distance and modified causal dependency have been applied to identify fault propagation paths [19]. The transfer entropy approach for continuous chemical processes was utilized to construct causal maps [8]. A combination of P&ID information and adjacency matrices was used to identify the root cause of plant-wide oscillations in a benchmark industrial case study [52]. The spectral envelope was used to identify the root cause of the abnormality on the same benchmark problem [53]. To find direct dependencies between process variables, a direct transfer entropy was proposed in [25]. To circumvent the strong assumption of a well-defined probability distribution, a transfer 0-entropy approach was proposed in [26] and [70]. In addition to this, the method based on lag-adjusted correlation combined with the use of connectivity information and causal measurement also showed promising results [7, 86]. The data-driven methods do not require process knowledge or P&IDs, and thus could be broadly used in many cases. However, they have some common limitations. Sufficiently large amount of data sets, especially data sets representing abnormal behaviors and covering multiple operating regions, are required. Such data sets sometimes may not be available in practice. The high computational complexity is another limitation that restricts the implementation of some data-driven approaches. For example, the computational complexity of TE is exponentially increasing with the number of cause and effect variables [8].

It is noteworthy that the above approaches are mainly based on process knowledge and continuous-valued process data, while the binary-valued alarm data is rarely examined for cause-effect reasoning. Since alarms are indications of abnormalities, a natural question is: can one identify abnormality propagation paths based on alarm data? Rather than capturing the directional connections, [100], [99] and [41] proposed systematic methods to detect the correlation between alarm signals. It is not uncommon to have a single root cause triggering a series of alarms that lead to alarm floods, and in this vein the work in [30] attempted to find frequent and common sequential alarm patterns from alarm logs. By combining the process topology and alarm logs, the interrelation rules were obtained for reducing alerts [79]. However, none of these methods have been able to satisfactorily identify propagation flow paths of abnormalities based on binary-valued alarm data. To our best knowledge, the work in [101] is the first attempt to capture the causal relations of industrial alarm signals using the transfer entropy approach. The work in this thesis provides a more complete study than [101], and is based on a different form of alarm signals.

1.3 Thesis Contributions

To ensure process safety and improve the efficiency of alarm systems, this thesis develops a variety of advanced alarm management techniques based on alarm data. The major contributions in this thesis that distinguish it from other work are summarized as follows:

1. Proposed a new method to detect correlated alarms and quantify their correlation levels. Specifically, the correlation between two alarm signals are determined by a statistical test based on the distribution of real occurrence delays. The correlation level is then calculated based on pseudo sequences obtained by super-imposing a Gaussian kernel to binary alarm signals.

2. Proposed a new local alignment algorithm to find similar alarm flood sequences, based on the basic local alignment search tool (BLAST) formulated in [4, 5]. The computational efficiency and alignment accuracy are improved owing to three novel elements, namely, the priority-based

similarity scoring strategy, the set-based pre-matching mechanism, and the modified seeding-extending steps.

3. Proposed a causality inference method based on binary-valued alarm data to identify abnormality propagation paths and detect root causes, in the absence of continuous-valued process data.

4. Defined and formulated a modified transfer entropy (TE) and a direct transfer entropy (DTE) as the basic statistical metrics, driven by two characteristics of alarm signals, namely, the random occurrence delays and the mutual independence of alarm occurrences.

5. Proposed a data-driven method to discover association rules of mode-dependent alarms from historical A&E logs, to help process engineers with the configuration of state-based alarming strategy.

6. Formulated the determination of mode-dependent alarms as a hypothesis testing problem and develop a modified a-priori algorithm to find frequent patterns of operating modes. Using the proposed method, the mode-dependent alarms for either a single operating mode or multiple operating modes are discovered.

1.4 Thesis Organization

The remainder of the thesis is organized as follows.

In Chapter 2, a new method to detect correlated alarms is presented. Section 2.1 gives an overview of the research work in this chapter. Section 2.2 defines the occurrence delay and presents the mechanism to obtain pseudo alarm signals. Section 2.3 proposes an algorithm to estimate the correlation delay and a novel statistical test to detect correlated alarms. Industrial case studies are provided in Section 2.4 to validate the obtained results. Section 2.5 summarizes the results and presents some conclusions.

In Chapter 3, a new local alignment algorithm to find similar alarm flood sequences is presented. Section 3.1 gives an overview of the research work in this chapter. Section 3.2 defines a priority-based similarity scoring strategy. The novel local alignment algorithm is proposed in Section 3.3. Section 3.4 presents industrial case studies to illustrate the effectiveness of the proposed algorithm and make a comparison with the modified SW algorithm. Section 3.5 summarizes the results and presents some conclusions.

In Chapter 4, a causality inference method based on binary-valued alarm data is presented. Section 4.1 gives an overview of the research work in this chapter. Section 4.2 defines the TE and DTE between alarm signals. In Section 4.3, a significance test, minimum requirement of alarm occurrences, and procedures of causality inference are presented. The effectiveness of the proposed method is illustrated by numerical and industrial case studies in Section 4.4, followed by a summary in Section 4.5.

In Chapter 5, a data-driven method to detect association rules of mode-dependent alarms is presented. Section 5.1 gives an overview of the research work in this chapter. Section 5.2 discusses the sate-based alarming in practice, categories of operating states, and alarm state transitions. Section 5.3 formulates the problem of the detection of mode dependent alarms. In Section 5.4, systematic strategies to detect the association rules of mode-dependent alarms are presented. The practicability and effectiveness of the proposed methods are illustrated by application to an industrial case in Section 5.5, followed by a summary in Section 5.6.

In Chapter 6, the concluding remarks and some promising directions of future work are presented.

Chapter 2

Detection and Quantification of Correlated Alarms with Occurrence Delays*

2.1 Overview

In this chapter, a new method to detect and quantify correlated alarm signals is studied. The development of this method is motivated by two drawbacks revealed in our industrial experience in the application of the method proposed in [100] to detect correlated alarms: (i) The estimated Sorgenfrei coefficients (as well as Jaccard coefficients in [57]) sometimes are quite small even for two alarm signals that have been known to be physically related; (ii) the distribution of correlation delays requires a large amount of data samples to get a sufficient number of alarm activations, which sometimes is a demanding requirement in practice. This study analyzes the causes of the two drawbacks and proposes corresponding solutions. To be precise, the contributions of this study is to propose a new method to detect correlated alarms and quantify their correlation level. The proposed method is composed of three parts. First, the so-called occurrence delay is defined and identified as the main cause of the first drawback mentioned above; in order to tolerate the presence of occurrence delays, a mechanism is proposed to generate pseudo alarm signals. Second, a novel approach is given to estimate the correlation delay in order to formulate real occurrence delays (ROD), and is shown to perform better than the

*A version of this chapter has been published as: Hu, W., Wang, J., & Chen, T. (2015). A new method to detect and quantify correlated alarms with occurrence delays. *Computers & Chemical Engineering*, 80, 189-198.

counterparts based on maximizing Sorgenfrei and Jaccard coefficients. Third, a statistical test based on the ROD is proposed to determine whether two alarm signals are correlated or not; this test requires a much smaller number of alarm activations so that the second drawback discussed earlier is resolved.

2.2 Occurrence Delays

This section first introduces alarm signals, and defines the occurrence delays caused by random noises. Next, a mechanism to generate pseudo alarm signals is presented to alleviate the effects of occurrence delays on the detection of correlated alarms.

2.2.1 Alarm Signals

Alarm signals can be represented in two forms. One formulation is that an alarm signal takes the value of 1 only at the time instant when its associated process signal goes into the abnormal state from the normal state. To be precise, if $x_p(t)$ is a discrete-time process signal, then the corresponding alarm signal $x_a(t)$ is generated as

$$x_a(t) = \begin{cases} 1, & \text{if } x_p(t) \notin \mathcal{T} \;\&\; x_p(t-1) \in \mathcal{T} \\ 0, & \text{otherwise} \end{cases}, \tag{2.1}$$

where $\mathcal{T}$ is the normal range of $x_p(t)$. Taking the generation of high alarms as an example, if a real-valued constant x_{tp} is the high-alarm trip point, then $\mathcal{T}$ stands for the normal range $(-\infty, x_{tp}]$. Another formulation of alarm signals is that an alarm signal takes the value of 1 throughout the time duration when the associated process signal is in the abnormal state, i.e.,

$$\tilde{x}_a(t) = \begin{cases} 1, & \text{if } x_p(t) \notin \mathcal{T} \\ 0, & \text{otherwise} \end{cases}. \tag{2.2}$$

Yang et al. [100] (Section II-B therein) investigated the two formulations for the purpose of detecting correlated alarms, and recommended the usage of $x_a(t)$ in (2.1), which is adopted in this study. Note that if the return to normal time instants in $\tilde{x}_a(t)$ need to be considered, the proposed algorithm is equally applicable after adding extra '1's to $x_a(t)$ in (2.1) at these time instants.

Chattering alarms are the most common nuisance alarms caused by measurement noises and disturbances [28]. They make fast transitions between alarm and non-alarm states in a short span of time without operators' response. Removing chattering alarms is a prerequisite for alarm flood analysis [17]. In the detection of correlated alarms, chattering alarms play a role as 'noise', while the alarms activated by actual abnormal conditions serve as 'signal'. If chattering alarms are dominant, then the signal-to-noise level is too low to detect correlated alarms. Therefore, it is necessary to remove chattering alarms before detecting correlated alarms. The detection and removal of chattering alarms have been studied recently by Naghoosi et al. [69], Wang & Chen [90], and Wang & Chen [91]. In particular, an m-sample delay timer is an effective way to remove chattering alarms, and can be represented as

$$x'_a(t) = \begin{cases} 1, & \text{if } \tilde{x}_a(t-m+1:t) = 1 \text{ and } x'_a(t-1) = 0 \\ 0, & \text{if } \tilde{x}_a(t-m+1:t) = 0 \text{ and } x'_a(t-1) = 1 \\ x'_a(t-1), & \text{otherwise} \end{cases}, \quad (2.3)$$

where $\tilde{x}_a(t-m+1:t)$ is a short notation for the set $\{\tilde{x}_a(t-m+1), \cdots, \tilde{x}_a(t)\}$. In the sequel, we assume that chattering alarms have been effectively removed by using the delay timer with the designed m as proposed by Wang & Chen [91].

2.2.2 Definition of Occurrence Delays

This subsection defines the occurrence delays that have significant effects on the detection of correlated alarms.

As revealed in (2.1), the occurrence of an alarm is associated with an event that the corresponding process signal runs into the abnormal state. Thus, the correlation between process signals is the essence leading to correlated alarms. In general, the relation between two correlated process signals $x_p(t)$ and $y_p(t)$ can be described as

$$y_p(t) = f(x_p(t-\tilde{\tau})) + \omega(t) \quad (2.4)$$

where the symbol $f(\cdot)$ stands for a functional linear/nonlinear relationship, the constant $\tilde{\tau}$ is the time delay, and the signal $\omega(t)$ is a disturbance additive to $y_p(t)$.

Suppose that the occurrence of an abnormality in $x_p(t)$ causes an alarm activation of $x_a(t)$, i.e., $x_a(t)$ changes from the non-alarm state 0 to the alarm state 1 at a particular time instant k. Because $x_p(t)$ and $y_p(t)$ are connected as that in (2.4), the abnormality in $x_p(t)$ may be propagated to $y_p(t)$, leading to an alarm activation in $y_a(t)$. Then, three terms are defined as follows:

Definition 1. If the disturbance $\omega(t)$ is absent, and an abnormality always causes the alarm signals $x_a(t)$ and $y_a(t)$, associated with two correlated process signals $x_p(t)$ and $y_p(t)$, changing from 0 to 1 at the time instants k and $k+\tau$ respectively for a positive integer k, then the time difference τ is defined as the correlation delay between $x_a\left(t\right)$ and $y_a\left(t\right)$. If $\omega(t)$ is present, then an abnormality may cause $x_a\left(t\right)$ and $y_a\left(t\right)$ respectively changing their values from 0 to 1 at the time instants k and $k+\lambda\left(k\right)$, where $\lambda\left(k\right)$ is a random variable. Here $\lambda\left(k\right)$ is referred to as the occurrence delay between $x_a\left(t\right)$ and $y_a\left(t\right)$. The sequence

$$\lambda_r\left(k\right) := \lambda\left(k\right) - \tau \tag{2.5}$$

is defined as the real occurrence delay (ROD) between $x_a\left(t\right)$ and $y_a\left(t\right)$.

It is worthy noting that owing to the possible presence of nonlinearity in $f\left(\cdot\right)$, the correlation delay τ does not have to be the same as $\tilde{\tau}$ in (2.4).

Example 1. This example is used to illustrate the presence of occurrence delays. The process signal $x_p(t)$ is generated as

$$x_p(\tilde{l}_{m-1}+1:\tilde{l}_m) = \begin{cases} \mathcal{N}(\mu_1,\sigma_1), & \text{if } x_s(m)=0 \\ \mathcal{N}(\mu_2,\sigma_2), & \text{if } x_s(m)=1 \end{cases}. \tag{2.6}$$

Here $x_p(t)$ is composed of M segments, and the length of the m-th segment for $m=1,2,\cdots,M$ is a uniform random variable l_m in the range $[L_1,L_2]$ for two positive integers L_1 and L_2; thus, $x_p(\tilde{l}_{m-1}+1:\tilde{l}_m)$ represents the samples of $x_p(t)$ over a range $t\in[\tilde{l}_{m-1}+1:\tilde{l}_m]$ with $\tilde{l}_{m-1}:=\sum_{i=1}^{m-1} l_i$ and $\tilde{l}_m=\tilde{l}_{m-1}+l_m$. In the m-th segment, $x_p(t)$ is in the normal or abnormal state, determined by a Bernoulli random variable $x_s(m)\sim\mathcal{B}(1,p)$, where p is the probability of abnormal state. The symbol $\mathcal{N}(\mu,\sigma)$ stands for the Gaussian distribution with mean μ and standard deviation σ. The disturbance $\omega(t)\sim\mathcal{N}(0,\sigma_\omega)$ is another Gaussian white noise being independent of $x_p(t)$. The process signal $y_p(t)$ is obtained from (2.4), where the process function $f(\cdot)$ is a first-order dynamic digital filter,

$$f(z)=\frac{0.5}{z-0.5}, \tag{2.7}$$

and the time lag between $x_p(t)$ and $y_p(t)$ is $\tilde{\tau}=60$. In this example, the parameters in (2.6) are selected as $M=100$, $L_1=30$, $L_2=90$, $\mu_1=5$, $\mu_2=10$, $\sigma_1=0.2$, $\sigma_2=0.2$, and $p=0.25$. Assume that the alarms $x_a(t)$ and $y_a(t)$ are generated from $x_p(t)$ and $y_p(t)$ as that in (2.1) by using high-alarm trip points $x_{tp}=7$ and $y_{tp}=7$.

If the noise $\omega(t)$ is absent, i.e., $\sigma_\omega = 0$, then $x_p(t)$ and $y_p(t)$ are respectively presented in Figs. 2.1-(a) and (b), while the alarm signals $x_a(t)$ and $y_a(t)$ are respectively given in Figs. 2.2-(a) and (b). The distribution of occurrence delays is shown in Fig. 2.3-(a). All the 25 occurrence delays are the same, namely, $\lambda(k) = 61$ for $k = 1, 2, \cdots, 25$.

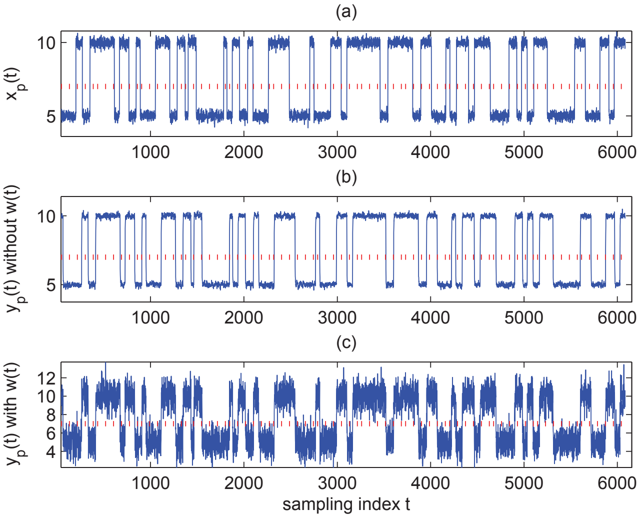


Figure 2.1: Process signals in Example 1: (a) $x_p(t)$ (solid) with x_{tp} (dash), (b) $y_p(t)$ (solid) with y_{tp} (dash) for the case that $\omega(t)$ is absent, (c) $y_p(t)$ (solid) with y_{tp} (dash) for the case that $\omega(t)$ is present.

If $\omega(t)$ is present with $\sigma_\omega = 1$, then $y_p(t)$ becomes the one in Fig. 2.1-(c), while $y_a(t)$ is shown in Fig. 2.2-(c). There are many chattering alarms in $y_a(t)$; using the m-sample delay timer in (2.3) with $m = 5$ can effectively remove all the chattering alarms, as shown by the resulting alarm signal $y'_a(t)$ in Fig. 2.2-(d). The resulting 25 occurrence delays are not equal, as shown by the histogram of these occurrence delays in Fig. 2.3-(b). Note that the usage of the delay timer introduces some extra time delay (about 5 samples) to the occurrence delays. □

The presence of occurrence delays has significant effects on the existing methods to detect correlated alarms, which are based on similarity measures of binary sequences such as Jaccard and Sorgenfrei coefficients [57, 100]. The

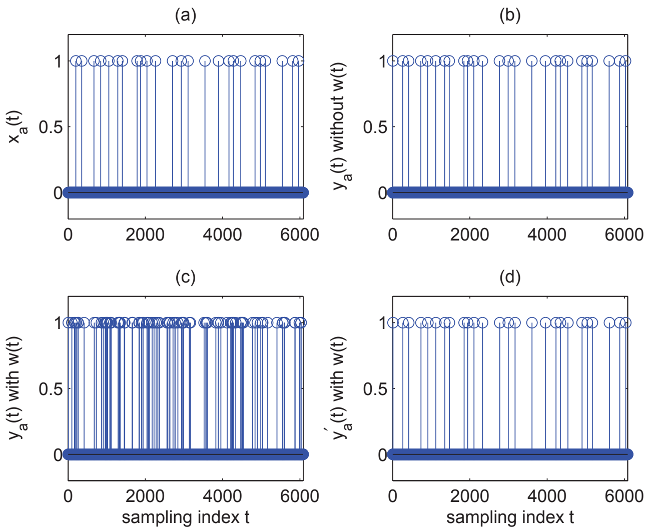


Figure 2.2: Alarm signals in Example 1: (a) $x_a(t)$, (b) $y_a(t)$ for the case that $\omega(t)$ is absent, (c) $y_a(t)$ for the case that $\omega(t)$ is present, (d) $y'_a(t)$ for the case that $\omega(t)$ is present

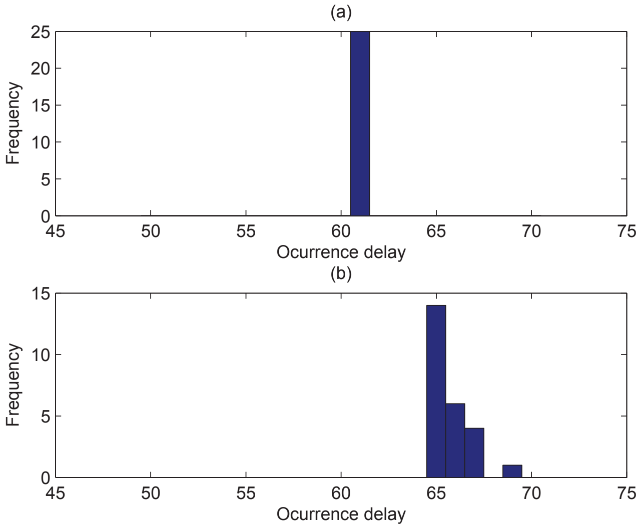


Figure 2.3: The histograms of occurrence delays without (a) and with (b) the presence of $\omega(t)$.

definitions of Jaccard and Sorgenfrei coefficients are

$$S_{\text{jacc}} = \frac{C}{N_x + N_y - C}, \tag{2.8}$$

$$S_{\text{sorg}} = \frac{C^2}{N_x N_y}. \tag{2.9}$$

Here C is the number of 1's appeared simultaneously in $x_a(t)$ and $y_a(t)$, and N_x and N_y are the total numbers of 1's in $x_a(t)$ and $y_a(t)$, respectively. The effectiveness of these coefficients requires an alignment of the time instants that $x_a(t)$ and $y_a(t)$ are activated to take the value 1. However, if occurrence delays are present, as illustrated in Example 1, then there is no way to achieve an alignment without errors, so that the methods based on Jaccard and Sorgenfrei coefficients may lead to erroneous conclusions. In particular, the calculated Jaccard or Sorgenfrei coefficient could be very low even for the case that two alarm sequences are indeed correlated. This issue had been noticed previously and two solutions were proposed [57, 99]. One solution is to pad extra 1's at both sides of each 1 in $x_a(t)$ [57]. That is, if $x_a(t) = 1$, then extra 1's are padded to $x_a(t)$ to have $x_a(t - \delta : t + \delta) = 1$ for a positive integer δ. This solution apparently introduces extra 1's and leads to biased estimates of Jaccard and Sorgenfrei coefficients; thus, it is not adopted here. Another solution is to transform the alarm signal $\tilde{x}_a(t)$ in (2.2) into the so-called pseudo alarm signals [99]. We will adopt the similar idea for $x_a(t)$ in (2.1), with the details to be presented in the next subsection.

2.2.3 Mechanism to Generate Pseudo Alarm Signals

This subsection is on the mechanism to generate pseudo alarm signals in order to tolerate the presence of occurrence delays.

First, the generation of pseudo alarm signals is to deal with the RODs $\lambda_r(k)$ in (2.5) so that it should be independent to the correlation delay τ. Thus, $x_a(t)$ or $y_a(t)$ needs to be shifted by τ samples properly in order to set the correlation delay after shifting to be zero. Without loss of generality, we choose to shift $y_a(t)$, i.e.,

$$y_{a,s}(t) := y_a(t - \tau). \tag{2.10}$$

After the data shifting, the sequence of $y_{a,s}(t)$ needs to be padded with zeros to have the same length with that of $x_a(t)$. Thus, two alarm sequences are formulated $X_a := \{x_a(t)\}_{t=1}^{N}$ and $Y_{a,s} := \{y_{a,s}(t)\}_{t=1}^{N}$, respectively.

Second, we obtain pseudo alarm signals for the alarm sequences X_a and $Y_{a,s}$ as follows. A kernel density estimation method is used to convert the binary alarm signal $x_a(t)$ to a real-valued pseudo alarm signal denoted as $x_{a,p}(t)$. If the Gaussian kernel function is used [82], then $x_{a,p}(t)$ is formulated as

$$x_{a,p}(t) = \sum_{i=1}^{N_x} e^{-\frac{(t-t_i)^2}{2h^2}}, \tag{2.11}$$

where the alarm quantity N_x is the total number of 1's in X_a, t_i is the time index that $x_a(t)$ takes the value 1, and h is the bandwidth of the Gaussian kernel function. Note that the difference between $x_{a,p}(t)$ and the pseudo alarm signals in Yang et al. [99] is that $x_a(t)$ in (2.1) is used, instead of $\tilde{x}_a(t)$ in (2.2). As shown in [100] (Section II-B therein), $x_a(t)$ is more suitable for the detection of correlated alarms than $\tilde{x}_a(t)$, so that generating pseudo alarm signals based on $x_a(t)$ is a more reasonable choice, too. Analogously to $x_{a,p}(t)$ in (2.11), the pseudo alarm signal for $y_{a,s}(t)$ in (2.10) is

$$y_{a,p}(t) = \sum_{j=1}^{N_y} e^{-\frac{(t-t_j)^2}{2h^2}}, \tag{2.12}$$

where N_y is the total number of 1's in $Y_{a,s}$, and t_j is the time index that $y_{a,s}(t)$ takes the value 1.

The bandwidth h in (2.11) and (2.12) is a critical parameter that affects the subsequent detection of correlated alarms. The selection of h should satisfy two conditions: 1) h should be large enough to cover the nonzero RODs between possibly correlated alarm signals; 2) h should not be too large so as to avoid the possible overlapping in $x_{a,p}(t)$ or $y_{a,p}(t)$ between adjacent 1's in $x_a(t)$ or $y_{a,s}(t)$. Thus, h is subject to such a tradeoff. An empirical choice of h is the sample mean of all nonzero RODs,

$$h = \frac{1}{L} \sum_{l=1}^{L} \tilde{\lambda}_r(l) \tag{2.13}$$

where $\tilde{\lambda}_r(l)$'s for $l = 1, 2, \cdots, L$ are the nonzero RODs of $\lambda_r(k)$ in (2.5).

Now, it is ready to calculate Pearson's correlation coefficient that is a widely-used statistics to measure the correlation between two random variables [54]. That is, Pearson's correlation coefficient between pseudo alarm sequences $X_{a,p} := \{x_{a,p}(t)\}_{t=1}^{N}$ and $Y_{a,p} := \{y_{a,p}(t)\}_{t=1}^{N}$ is

$$r_{xy} = \frac{1}{N-1} \sum_{t=1}^{N} \left(\frac{x_{a,p}(t) - \bar{x}_{a,p}}{S_x} \right) \left(\frac{y_{a,p}(t) - \bar{y}_{a,p}}{S_y} \right) \tag{2.14}$$

where $\bar{x}_{a,p}$ ($\bar{y}_{a,p}$) and S_x (S_y) are the sample mean and standard deviation of $x_{a,p}(t)$ ($y_{a,p}(t)$), respectively. The overlapped area between two pseudo sequences $X_{a,p}$ and $Y_{a,p}$ is able to alleviate the effects of occurrence delays, and thus to reflect the correlation between alarm signals more accurately than the similarity measures for binary alarm signals such as Jaccard and Sorgenfrei coefficients. Two numerical examples are presented here to support this statement.

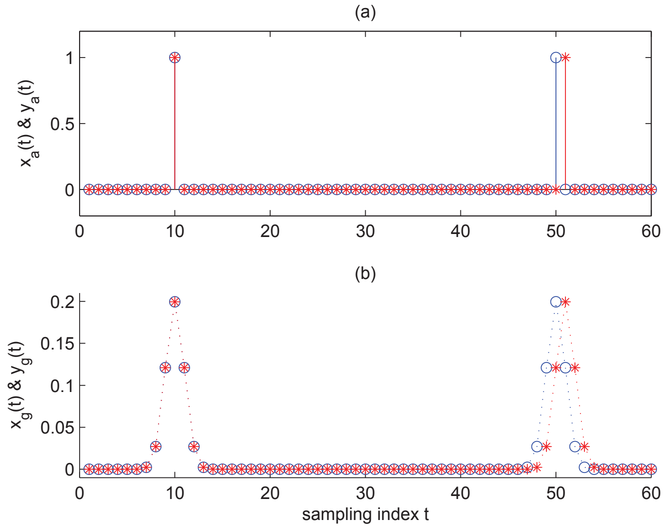


Figure 2.4: Example 2a: (a) $x_a(t)$ and $y_a(t)$, (b) $x_g(t)$ and $y_g(t)$.

Example 2a. This is an illustrative example with only two occurrences of alarms in $x_a(t)$ and $y_a(t)$ as shown in Fig. 2.4-(a). The first pair of alarms occurs at the time instant $t = 10$ for both $x_a(t)$ and $y_a(t)$, while the second pair of alarms occurs at $t = 50$ for $x_a(t)$ and $t = 51$ for $y_a(t)$, respectively. If the one-sample time difference between the second pair of alarms is a ROD, then $x_a(t)$ and $y_a(t)$ are regarded to be completely correlated with a correlation measure about 1. However, Jaccard and Sorgenfrei coefficients are respectively calculated from (2.8) and (2.9) as $S_{\text{jacc}} = 1/(2+2-1) = 0.3333$ and $S_{\text{sorg}} = 1/(2 \cdot 2) = 0.25$, which are far away from 1. If pseudo alarm signals are generated, as shown in Fig. 2.4-(b), Pearson's correlation coefficient r_{xy} is equal to 0.8745. Thus, pseudo alarm signals can effectively alleviate the negative effects of RODs on the detection of correlated alarms.

Example 2b. The configuration is the same as Example 1, except that the original correlation delay $\tau = 61$ (see Fig. 2.3-(a)) has been removed. 1000

Table 2.1: Calculated correlation statistics for Example 2b.

Signal type	Statistic	Mean	Std
Pseudo alarm signals	Pearson's correlation coefficient	0.8532	0.0290
Binary alarm signals	Jaccard coefficient	0.4780	0.0569
	Sorgenfrei coefficient	0.4271	0.0487

Monte Carlo simulations are implemented. In each simulation, the noise $\omega(t)$ is randomly generated with the standard deviation $\sigma_\omega = 1.4$. To generate pseudo signals, the bandwidth h is calculated using (2.13). The value of h is changing with the mean 1.01 and standard deviation 0.04 as calculated from the simulations. Table 2.1 compares the sample means and standard deviations of Pearson's correlation coefficient based on pseudo alarm signals, and Jaccard and Sorgenfrei coefficients based on alarm signals. Considering the linear relationship between $x_p(t)$ and $y_p(t)$, the alarm signals $x_a(t)$ and $y_a(t)$ should be highly correlated to each other and the correlation statistics should be close to 1. It is obvious that Pearson's correlation coefficients are much higher than Jaccard and Sorgenfrei coefficients, and are consistent with the fact that $x_a(t)$ and $y_a(t)$ are indeed correlated. Hence, the utilization of pseudo alarm signals is effective in avoiding false detection in the presence of occurrence delays.

2.3 Detection of Correlated Alarms

This section presents a new approach to estimate the correlation delay, and a statistical test to determine whether two alarm signals are correlated or not. Next, the procedures of the proposed method to detect and quantify correlated alarms are summarized.

2.3.1 Estimation of the Correlation Delay

In Section 2.2.3, the correlation delay τ is set to zero by shifting data samples before generating pseudo alarm signals. This subsection provides a new approach to estimate τ.

Based on the definition of the ROD $\tilde{\lambda}(k)$ in (2.5), it is obvious that $\tilde{\lambda}(k)$ should be close to zero for correlated alarms. This leads us to estimate the correlation delay τ as

$$\hat{\tau} = \arg\min_{\tau} \bar{\lambda}_r(\tau), \tag{2.15}$$

where

$$\bar{\lambda}_r(\tau) = \frac{1}{K}\sum_{k=1}^{K} |\lambda(k) - \tau|. \tag{2.16}$$

Here $\lambda(k)$'s for $k = 1, 2, \cdots, K$ are the occurrence delays between $x_a(t)$ and $y_a(t)$. Based on (2.15), a so-called occurrence delay based estimation (ODE) algorithm is proposed, with the computer pseudo codes listed as Algorithm 1. The ODE algorithm provides an estimate of the correlation delay τ. If τ is positive, it is said that $y_a(t)$ follows $x_a(t)$ by τ samples; otherwise, $y_a(t)$ proceeds $x_a(t)$ by $|\tau|$ samples.

Algorithm 1 Occurrence delay based estimation (ODE) algorithm

```
Input Argument #1: Alarm sequences $X_a := \{x_a(t)\}_{t=1}^{N}$ and $Y_a := \{y_a(t)\}_{t=1}^{N}$;
Input Argument #2: $L$ the maximum value of the correlation delay;
for $i = -L$ to $L$ do
    if $i < 0$ then
        $S_1 = Y_a(1 : N + i)$; $S_2 = X_a(1 - i : N)$;
    else
        $S_1 = X_a(1 : N - i)$; $S_2 = Y_a(1 + i : N)$;
    end if
    $a_1 =$ find $(S_1 == 1)$; $a_2 =$ find $(S_2 == 1)$;
    $q_1 =$ length $(a_1)$; $q_2 =$ length $(a_2)$;
    if $q_1 < q_2$ then
        for $k = 1$ to $q_1$ do
            $\lambda_r(k) = \min(|a_1(k) - a_2|)$;
        end for
    else
        for $k = 1$ to $q_2$ do
            $\lambda_r(k) = \min(|a_2(k) - a_1|)$;
        end for
    end if
    $\bar{\lambda}_r(i) =$ mean $(\lambda_r)$;
end for
$\hat{\tau} = \arg_i \min \bar{\lambda}_r(i)$.
```

Estimating the correlation delay τ is also a necessary step in the existing methods of detecting correlated alarms based on Jaccard and Sorgenfrei coefficients [57, 100]. For instance, the estimation of τ based on Sorgenfrei coefficient is [100],

$$\hat{\tau} = \arg\max_{\tau} S_{\text{sorg}}(\tau), \tag{2.17}$$

where $S_{\text{sorg}}(\tau)$ takes the value of Sorgenfrei coefficient in (2.9) between $x_a(t)$ and $y_a(t+\tau)$. Thus, $\hat{\tau}$ in (2.17) is based on the best alignment of 1's in alarm signals to have the largest value of Sorgenfrei coefficient, while $\hat{\tau}$ in (2.15) is to make the mean of RODs close to zero. Therefore, the proposed approach to estimate τ is very different from the existing approaches in [57] and [100].

Example 3. The configuration is the same as in Example 1. Parameters M, L_1, L_2, μ_1, μ_2, σ_1, σ_2, x_{tp} and y_{tp} are the same as in Example 1, while other parameters $\tilde{\tau}$, p, and σ_ω, are randomly determined in each simulation, in the ranges $\tilde{\tau} \in [-80, 80]$, $p \in [0.15, 0.5]$, and $\sigma_\omega \in [2, 5]$. 10000 Monte Carlo simulations are implemented. The ODE algorithm and the approaches by maximizing Sorgenfrei and Jaccard coefficients, e.g., that in (2.17), are applied to estimate the correlation delay $\tau = 61$ between $x_a(t)$ and $y_a(t)$. Table 2.2 presents the mean absolute errors (MAE) and mean absolute percentage errors (MAPE) from the 10000 Monte Carlo simulations. It is obvious that the ODE algorithm performs much better than the other two approaches. The difference arises from a fact that the presence of occurrence delays introduces errors in the alignment of alarm activations, and leads to the erroneous estimates of τ in the approaches based on Jaccard and Sorgenfrei coefficients; in contrast, such an alignment of alarm activations is not required in the ODE algorithm.

Table 2.2: Estimation of the correlation delay using different approaches for Example 3.

Approaches	MAE	MAPE %
The ODE algorithm	0.5752	2.4981
Estimation based on Jaccard coefficients	1.2385	7.6430
Estimation based on Sorgenfrei coefficients	1.1811	7.5249

2.3.2 Statistical Test

This subsection provides a statistical test to tell whether two alarm signals are correlated or not.

The Pearson's correlation coefficient r_{xy} in (2.14) based on pseudo alarm signals $x_{a,p}(t)$ and $y_{a,p}(t)$ is able to quantify the level of correlation; however,

r_{xy} is not suitable for determining whether the alarm signals $x_a(t)$ and $y_a(t)$ are correlated, because the distributions of $x_{a,p}(t)$ and $y_{a,p}(t)$ are unavailable so that a threshold for testing $r_{xy} \neq 0$ is rather difficult to be determined.

If alarm signals $x_a(t)$ and $y_a(t)$ are correlated, then the estimated correlation delays $\hat{\tau}$'s in (2.15) for different sampled sequences of $x_a(t)$ and $y_a(t)$ should be concentrated to a small interval around the actual correlation delay τ; in contrast, if $x_a(t)$ and $y_a(t)$ are uncorrelated, then $\hat{\tau}$'s in (2.15) are distributed randomly. A similar observation has been presented in [100], where a statistical test is proposed on whether the distribution of $\hat{\tau}$ in (2.17) is concentrated to a small interval. However, such a statistical test has a drawback for practical applications, namely, multiple estimates of $\hat{\tau}$ have to be obtained. A recommendation therein is to have at least 10 estimates of $\hat{\tau}$, each of which requires data samples of $x_a(t)$ and $y_a(t)$ having more than 30 alarm activations. As a result, the statistical test in [100] requires the data samples containing more than 300 alarm activations, which sometimes is hard to meet in practice. Hence, it is desirable to design a statistical test that has less demanding requirements on the data samples of $x_a(t)$ and $y_a(t)$.

A significant test is presented in [7] to determine whether two process signals $x_p(t)$ and $y_p(t)$ are linearly correlated, solely based on one single estimate of cross-correlation between $x_p(t)$ and $y_p(t)$. This significant test is certainly applicable to pseudo alarm signals $x_{a,p}(t)$ and $y_{a,p}(t)$ here. That is, if r_{xy} in (2.14) satisfies the following inequality (2.18), then $x_{a,p}(t)$ and $y_{a,p}(t)$ are claimed to be correlated,

$$r_{xy} \geq r_{th} := \mu_{r_{xy}} + 3\sigma_{r_{xy}}, \tag{2.18}$$

where $\mu_{r_{xy}}$ and $\sigma_{r_{xy}}$ are the sample mean and standard deviation of r_{xy}, respectively; an empirical estimate of the threshold r_{th} is

$$r_{th} = 1.85N^{-0.41} + 2.37N^{-0.53}, \tag{2.19}$$

where N is the data length of $x_{a,p}(t)$ and $y_{a,p}(t)$. It is obvious that r_{th} only relates to the data length N. However, the detected alarm correlation based on $x_{a,p}(t)$ and $y_{a,p}(t)$ is highly associated with the distribution of alarm points in $x_a(t)$ and $y_a(t)$. If the empirical estimate in (2.19) is directly applied, the significance threshold is always the same no matter how the alarm rate and alarm count of $x_a(t)$ and $y_a(t)$ change. Therefore, a more accurate estimate of the significance threshold should be specified by incorporating the distribution information of alarm data.

We adopt the similar idea of the significant test in (2.18) to design a statistical test for the ROD $\tilde{\lambda}(k)$ in (2.5) between $x_a(t)$ and $y_a(t)$. A critical observation is that for correlated alarms, the RODs $\tilde{\lambda}(k)$ are close to zero, while $\tilde{\lambda}(k)$ could be very large for uncorrelated alarms. To be consistent with the estimate $\hat{\tau}$ in (2.15), a lag factor is defined,

$$\varphi = \min_{\tau} \bar{\lambda}_r(\tau), \tag{2.20}$$

where $\bar{\lambda}_r(\tau)$ is given in (2.16). Since φ is always positive and is close to zero if $x_a(t)$ and $y_a(t)$ are correlated, a one-sided hypothesis test should be applied. Analogously to (2.18), $x_a(t)$ and $y_a(t)$ are claimed to be correlated if the following inequality holds,

$$\varphi < \varphi_{th} := \mu_\varphi - 3\sigma_\varphi \tag{2.21}$$

where μ_φ and σ_φ are respectively the mean and the standard deviation of φ. Under the null hypothesis that $x_a(t)$ and $y_a(t)$ are uncorrelated, we find that μ_φ and σ_φ can be modelled as functions of two factors, namely, the alarm probability rate denoted as p_r and the alarm quantity denoted as q_r,

$$\begin{aligned} \mu_\varphi &= a_\mu p_r^{b_\mu} q_r^{c_\mu} \\ \sigma_\varphi &= a_\sigma p_r^{b_\sigma} q_r^{c_\sigma} \end{aligned} \tag{2.22}$$

The two functions can be estimated empirically. For instance, Fig. 2.5 shows the histograms of φ in 1000 Monte Carlo simulations for uncorrelated alarms with a pair of p_r and q_r. To identify the parameters a_μ, b_μ, c_μ and $a_\sigma, b_\sigma, c_\sigma$, the values of p_r and q_r are changed in the ranges $[0.0003, 0.02]$ and $[20, 1000]$ with certain step sizes, respectively. For each pair of p_r and q_r, 1000 Monte Carlo simulations are implemented to estimate μ_φ and σ_φ. The estimated values of μ_φ and σ_φ are presented in Figs. 2.6 and 2.7, respectively. By the standard least-squares based curve-fitting method, the functions in (2.22) are estimated as

$$\begin{aligned} \mu_\varphi &= 0.4390 p_r^{-1.0010} q_r^{0.0163}, \\ \sigma_\varphi &= 0.7886 p_r^{-0.9900} q_r^{-0.4855}. \end{aligned} \tag{2.23}$$

As shown in Figs. 2.6 and 2.7, the estimated functions in (2.23) fit the samples of (p_r, q_r, μ_φ) and $(p_r, q_r, \sigma_\varphi)$ very well. Therefore, the threshold φ_{th} in (2.21)

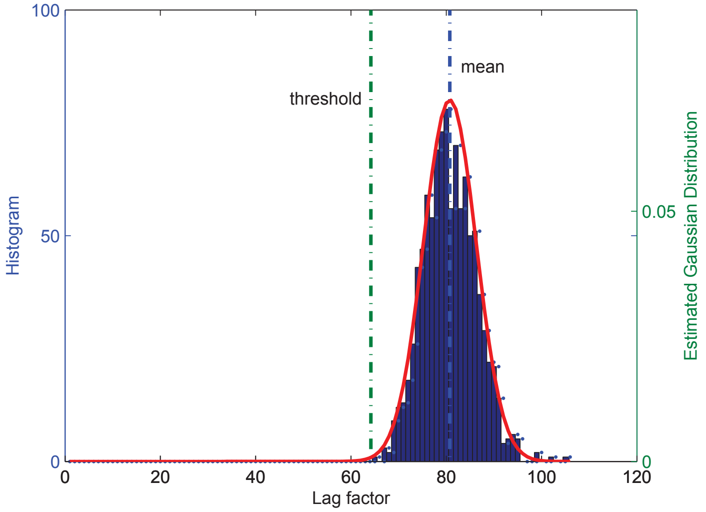


Figure 2.5: Histogram of φ in 1000 Monte Carlo simulations for uncorrelated alarms with $p_r = 0.005$ and $q_r = 360$.

can be calculated for one set of $x_a(t)$ and $y_a(t)$ from (2.23); the values of p_r and q_r can be obtained from $x_a(t)$ and $y_a(t)$ as

$$q_r = \min(N_x, N_y),\ p_r = \frac{q_r}{N}, \tag{2.24}$$

where N_x and N_y are the alarm quantities in $x_a(t)$ and $y_a(t)$, respectively, and N is the data length of $x_a(t)$ and $y_a(t)$. If (2.21) is satisfied, then the RODs are claimed to be close to zero, and the alarm signals $x_a(t)$ and $y_a(t)$ are believed to be correlated. Then, the level of correlation can be measured by Pearson's correlation coefficient r_{xy} in (2.14) based on pseudo alarm signals $x_{a,p}(t)$ and $y_{a,p}(t)$.

The numbers of alarm quantities N_x in $x_a(t)$ and N_y in $y_a(t)$ cannot be too small, in order to make the statistical test in (2.21) reliable. The functions in (2.23) can be used to determine the least number of alarm quantities. In general, the threshold φ_{th} is positive; however, if the alarm quantity q_r is too small, φ_{th} could drop below zero. Based on (2.21) and (2.23), we can find the relation between q_r and p_r to achieve $\varphi_{th} = 0$ as

$$q_r = 28.6936\ p_r^{0.0219}. \tag{2.25}$$

The alarm probability rate p_r is a real value in the range $[0, 1]$. In practice, after removing chattering alarms, p_r should be less than 0.05, namely, 3 alarm activations per minute, which is a well-accepted benchmark for chattering alarms [47]. Eq. (2.25) yields $q_r = 26.8715$ for $p_r = 0.05$. Hence, we choose $q_r = 27$ as a threshold of trusty, i.e., both of $x_a(t)$ and $y_a(t)$ should contain at least 27 alarm activations to ensure that the statistical test is reliable. In contrast, as mentioned earlier in this subsection, the statistical test in [100] requires at least 300 alarm activations. Therefore, the proposed statistical test in (2.21) is much less demanding on the data samples of $x_a(t)$ and $y_a(t)$.

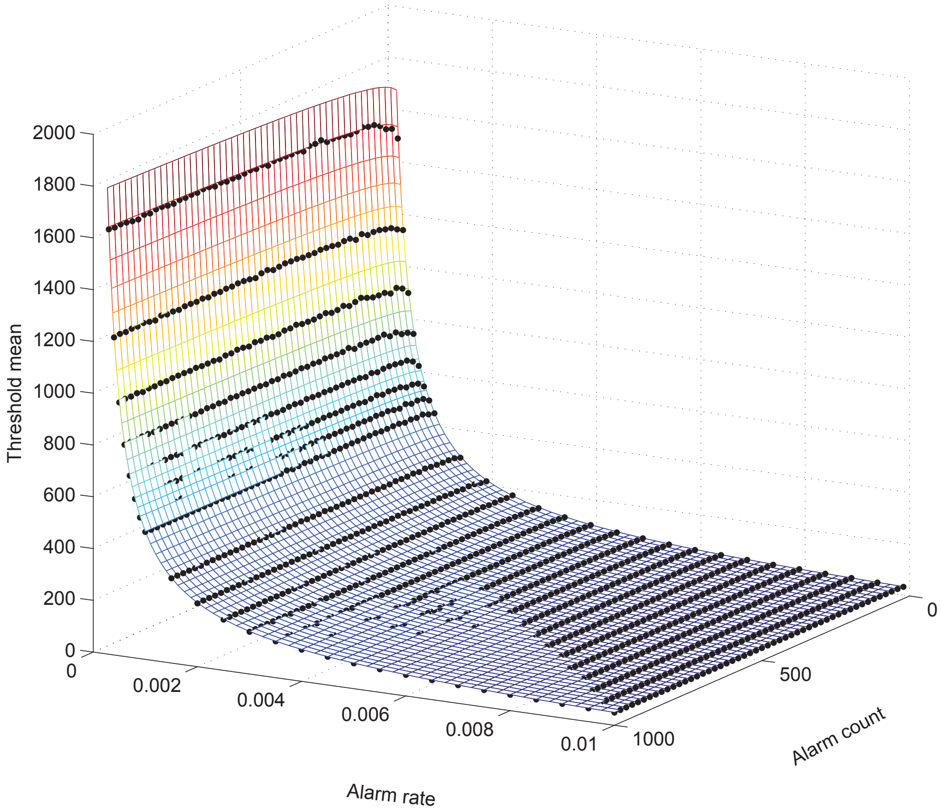


Figure 2.6: μ_φ as a function of p_r and q_r.

2.3.3 Detection Procedure

The complete procedure of the proposed method to detect and quantify correlated alarms is composed of the following steps, and is depicted in Fig. 2.8.

1. Obtain two alarm sequences $X_a := \{x_a(t)\}_{t=1}^N$ and $Y_a := \{y_a(t)\}_{t=1}^N$ with at least 27 alarm activations in each sequence, and remove chattering

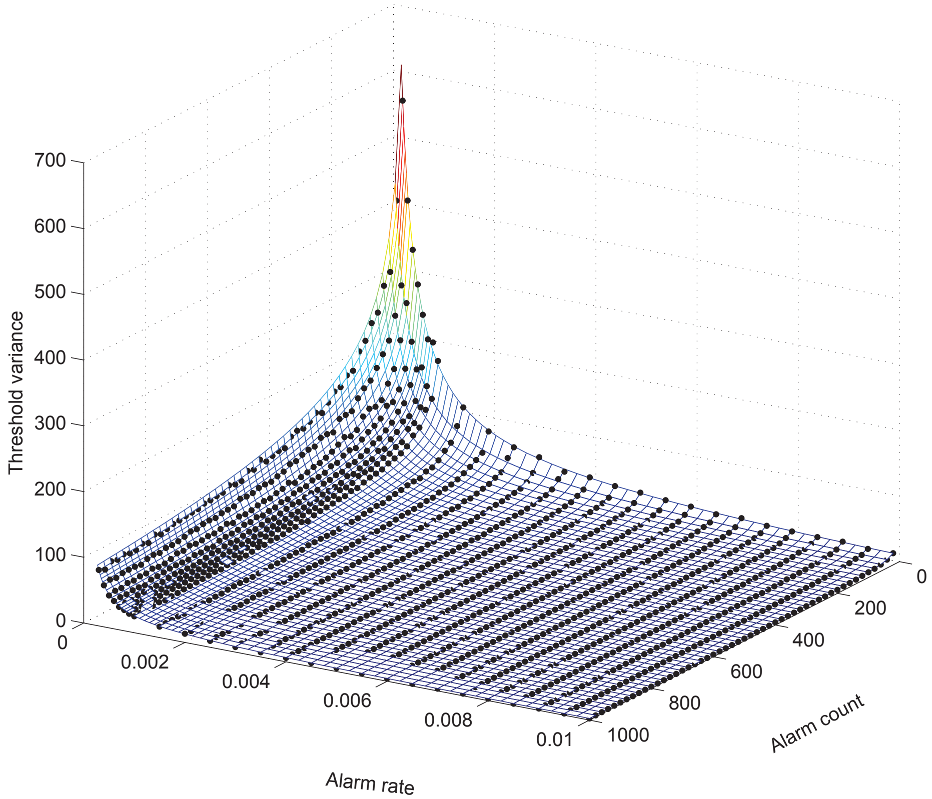


Figure 2.7: σ_φ as a function of p_r and q_r.

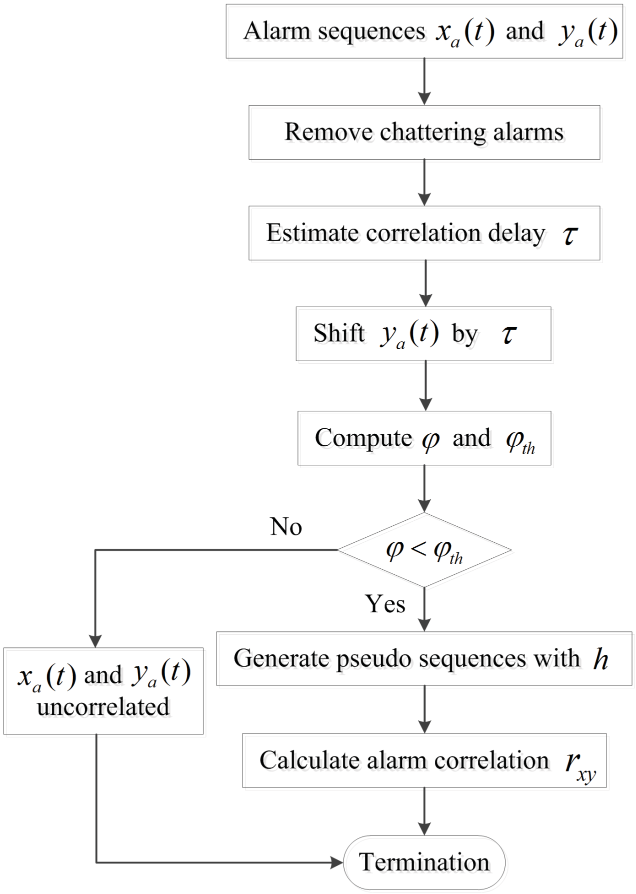
alarms by the delay timer in (2.3) if necessary;

2. Estimate the correlation delay τ between X_a and Y_a using the ODE algorithm in (2.15), and remove τ by shifting $y_a(t)$ to reach $y_{a,s}(t)$ in (2.10) to formulate $Y_{a,s} := \{y_{a,s}(t)\}_{t=1}^{N}$;

3. Perform the statistical test in (2.21) based on X_a and $Y_{a,s}$ to determine whether $x_a(t)$ and $y_a(t)$ are correlated, where the threshold φ_{th} is calculated using the empirical functions of μ_φ and σ_φ in (2.23) with the parameters p_r and q_r in (2.24). If (2.21) holds, then $x_a(t)$ and $y_a(t)$ are claimed to be correlated, and we proceed to Step 4 to quantify the correlation level; otherwise, $x_a(t)$ and $y_a(t)$ are regarded to be uncorrelated and the detection procedure is terminated;

4. Transform X_a and $Y_{a,s}$ respectively into the pseudo alarm sequences $X_{a,p} := \{x_{a,p}(t)\}_{t=1}^{N}$ and $Y_{a,p} := \{y_{a,p}(t)\}_{t=1}^{N}$ using the kernel bandwidth

h in (2.13), and calculate Pearson's correlation coefficient r_{xy} in (2.14).



Alarm sequences $x_a(t)$ and $y_a(t)$
Remove chattering alarms
Estimate correlation delay τ
Shift $y_a(t)$ by τ
Compute φ and φ_{th}
$\varphi < \varphi_{th}$
No
Yes
$x_a(t)$ and $y_a(t)$ uncorrelated
Generate pseudo sequences with h
Calculate alarm correlation r_{xy}
Termination

Figure 2.8: Flowchart for detecting correlated alarms.

2.4 Industrial Case Studies

This section provides two industrial case studies as the representative examples of many successful applications of the proposed method. The case studies show that the proposed method performs well in detecting correlated or uncorrelated alarms.

2.4.1 Case I: Correlated Alarms

Historical samples of two alarm signals $x_a(t)$ and $y_a(t)$ are obtained from an oil plant in Canada, as shown in Fig. 2.9. The data length is $N = 241556$ with the sampling interval 1 sec (about 67 hours). Alarm quantities of $x_a(t)$

and $y_a(t)$ are $N_x = 653$ and $N_y = 242$, respectively. The m-sample delay timer in (2.3) is exploited to remove chattering alarms in $x_a(t)$, where $m = 20$ is used as recommended in [90]; the alarm quantity in $x_a(t)$ is reduced to $N_x = 391$. Using the ODE algorithm with the searching range $\tau \in [-100, 100]$ seconds, the correlation delay τ is estimated to be $\hat{\tau} = 28$. Fig. 2.10 shows the sequence of the ROD $\tilde{\lambda}_r(k)$ and the estimated distribution of ROD in the form of a histogram; in particular, the RODs are found to be mainly located between -30 and 30.

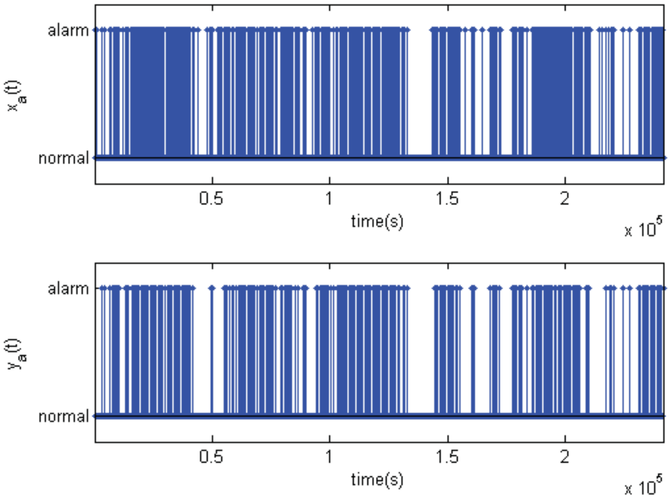


Figure 2.9: Samples of $x_a(t)$ (top) and $y_a(t)$ (bottom) for Case I.

After shifting $y_a(t)$ backward by $\hat{\tau} = 28$ samples, the lag factor is calculated to be $\varphi = 14.591$. Since $y_a(t)$ contains fewer alarm points, the threshold $\varphi_{th} = 329.121$ is calculated by using μ_φ and σ_φ in (2.23) based on the alarm quantity of $y_a(t)$, i.e., $q_r = N_y = 242$ and $p_r = N_y/N = 0.001$. Because $\varphi < \varphi_{th}$, $x_a(t)$ and $y_a(t)$ are claimed to be correlated. Next, the pseudo alarm signals $x_{a,p}$ in (2.11) and $y_{a,p}(t)$ in (2.12) are generated, with the kernel bandwidth h in (2.13) equal to 15.026. Finally, Pearson's correlation coefficient is calculated to be $r_{xy} = 0.571$. As a comparison, the correlated alarm detection methods based on Jaccard coefficients S_{jacc} in (2.8) and Sorgenfrei coefficients S_{sorg} in (2.9) [57, 100] are implemented based on $x_a(t)$ and $y_a(t)$. For instance, Sorgenfrei coefficients S_{sorg} in (2.9) are calculated for different values of τ for $x_a(t)$ and $y_a(t + \tau)$, and the maximum value of the calculated Sorgenfrei coefficients is taken as the final estimate. The calculated Sorgen-

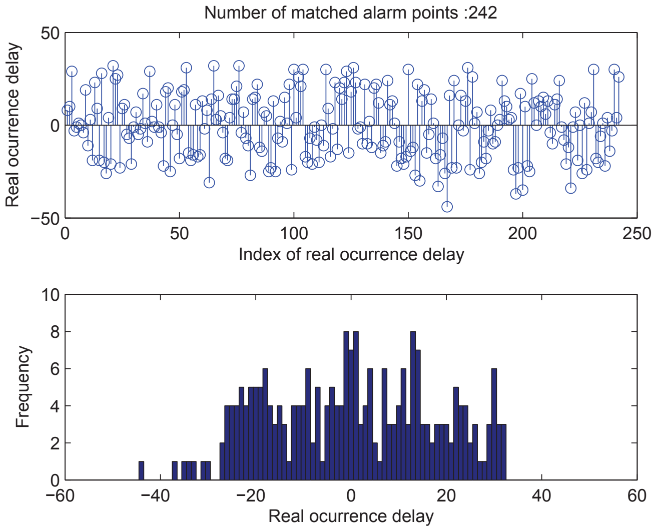


Figure 2.10: The sequence of ROD (top) and the histogram of ROD (bottom) for Case I.

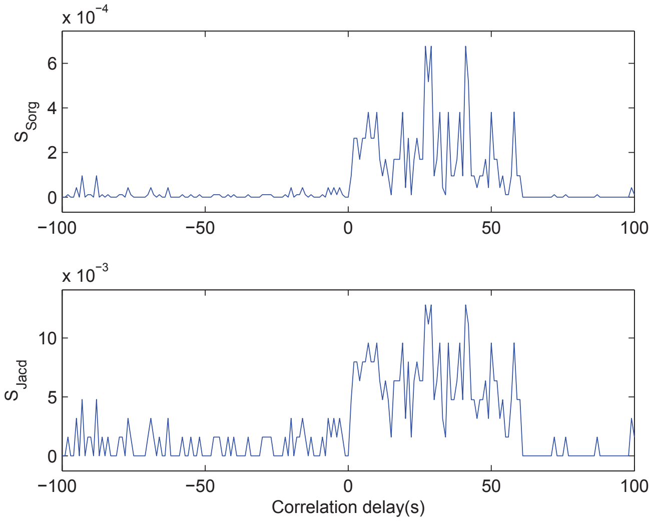


Figure 2.11: The estimated Sorgenfrei coefficients (top) and Jaccard coefficients (bottom) for different values of τ for Case I.

frei and Jaccard coefficients are shown in Fig. 2.11. Their maximum values are $S_{\text{sorg}} = 0.000676$ and $S_{\text{jacc}} = 0.0128$. The thresholds of Sorgenfrei and Jaccard coefficients to separate independent and correlated alarm signals are respectively equal to 1/16 and 1/7 [100] (Proposition 1 therein); thus, the calculated Sorgenfrei and Jaccard coefficients are smaller than their thresholds, respectively, so that $x_a(t)$ and $y_a(t)$ are claimed to be uncorrelated. This erroneous results are caused by the presence of the occurrence delays, which introduce errors in the alignment of alarm activations causing such small values of Sorgenfrei and Jaccard coefficients. In contrast, the proposed method gives a relatively higher correlation coefficient. In fact, the process signals $x_p(t)$ and $y_p(t)$ associated with $x_a(t)$ and $y_a(t)$ are physically connected: $y_p(t)$ is the flow rate of the fuel gas in a burner which has two supplies from cell A and cell B; $x_p(t)$ is the flow rate of the feed fuel gas from cell B. Thus, $x_a(t)$ and $y_a(t)$ are expected to be correlated.

2.4.2 Case II: Uncorrelated Alarms

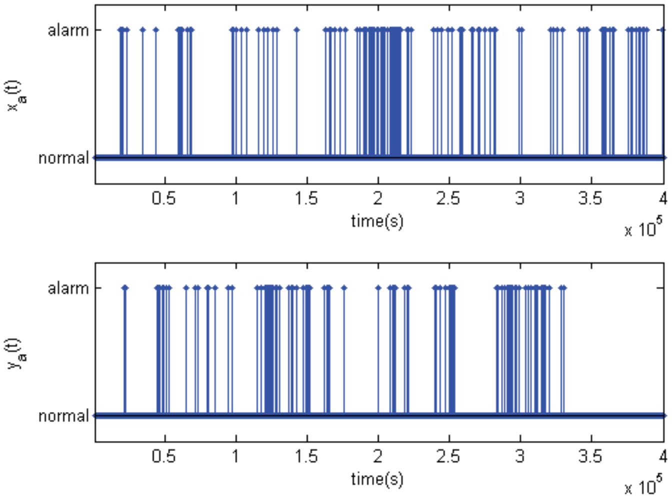


Figure 2.12: Samples of $x_a(t)$ (top) and $y_a(t)$ (bottom) for Case II.

To validate the performance of the proposed method for uncorrelated alarms, historical samples of two alarm signals $x_a(t)$ and $y_a(t)$ are obtained from a refinery plant in Alberta, Canada, as shown in Fig. 2.12. The data length is $N = 400000$ with the sampling interval 1 sec (about 111 hours). Alarm quantities of $x_a(t)$ and $y_a(t)$ are $N_x = 152$ and $N_y = 127$, respectively. No chattering alarms are detected. Using the ODE algorithm with the searching range $\tau \in [-300, 300]$ seconds, the correlation delay τ is estimated to be $\hat{\tau} = 227$. Fig. 2.13 shows the sequence of the ROD $\tilde{\lambda}_r(k)$ and the estimated distribution of ROD in the form of histogram; in particular, the RODs are found to span a broad range from -12587 to 11599. Observing the RODs, we can see that even the correlation delay has been removed, most alarm points in $y_a(t)$ cannot be well matched with the alarm points in $x_a(t)$.

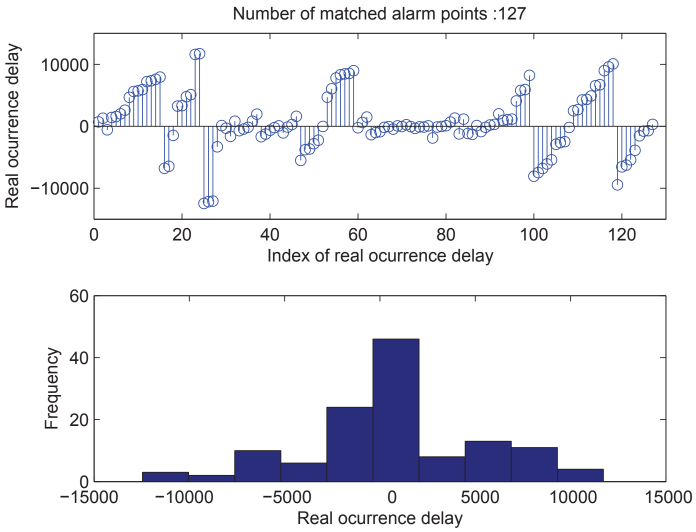


Figure 2.13: The sequence of ROD (top) and the histogram of ROD (bottom) for Case II.

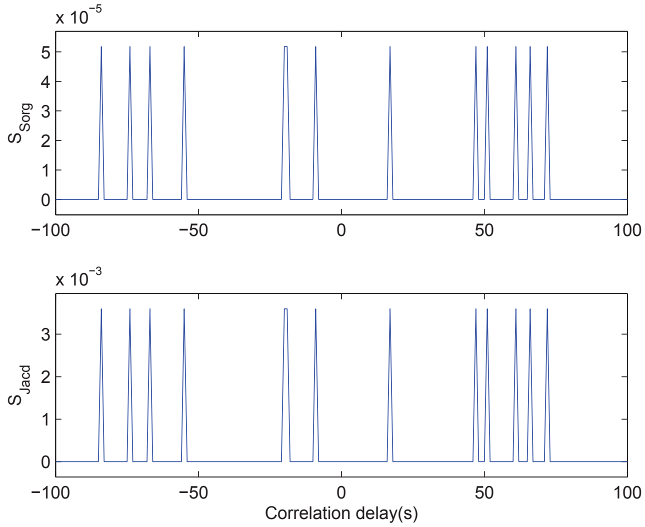


Figure 2.14: The estimated Sorgenfrei coefficients (top) and Jaccard coefficients (bottom) for different values of τ for Case II.

After removing the correlation delay, the lag factor is calculated to be $\varphi = 3352.244$. Since $y_a(t)$ contains fewer alarm points, the threshold $\varphi_{th} = 853.474$ is calculated from (2.23) based on the alarm quantity of $y_a(t)$, i.e., $q_r =$

$N_y = 127$ and $p_r = N_y/N = 3.177 \times 10^{-4}$. Because $\varphi > \varphi_{th}$, $x_a(t)$ and $y_a(t)$ are believed to be uncorrelated and the computation is thereby terminated. As a comparison, the correlated alarm detection methods based on Jaccard coefficient S_{jacc} in (2.8) and Sorgenfrei coefficient S_{sorg} in (2.9) are employed based on $x_a(t)$ and $y_a(t)$. The calculated Sorgenfrei and Jaccard coefficients are shown in Fig. 2.14. Their maximum values are $S_{\text{sorg}} = 5.180 \times 10^{-5}$ and $S_{\text{jacc}} = 3.597 \times 10^{-3}$; thus, $x_a(t)$ and $y_a(t)$ are claimed to be uncorrelated. This is consistent with the conclusion obtained using the proposed method. Considering the fact that the process signals associated with $x_a(t)$ and $y_a(t)$ are located at different and disconnected units, the conclusion that $x_a(t)$ and $y_a(t)$ are uncorrelated is reasonable.

2.5 Summary

The study in this chapter proposed a new method to detect correlated alarms and quantify the correlation level. First, the occurrence delay was defined in Definition 1 and was shown to be the main cause of erroneous small values of Sorgenfrei and Jaccard coefficients that require the alignment of alarm activations. To tolerate the presence of occurrence delays, the binary alarm signals were transformed to continuous-valued pseudo alarm signals via (2.11) and (2.12), so that the Pearson's correlation coefficient r_{xy} in (2.14) can be used to quantify the correlation level. Second, the ODE algorithm was provided to estimate the correlation delay τ in order to be removed from occurrence delays to reach the RODs. The ODE algorithm was shown to perform better than the existing methods based on Sorgenfrei and Jaccard coefficients. This is also owing to the presence of occurrence delays causing errors in the alignment of alarm activations. Third, a statistical test based on the RODs was proposed to detect correlated alarms. That is, if (2.21) is satisfied, then the alarm signals $x_a(t)$ and $y_a(t)$ are claimed to be correlated. A rule of thumb is to have at least 27 alarm activations to make the statistical test reliable, while the counterparts in literature requires at least 300 alarm activations. Thus, the proposed statistical test is much less demanding on sample lengths of $x_a(t)$ and $y_a(t)$. Numerical examples were provided to support the obtained results, and industrial case studies showed that the proposed method performed well in detecting correlated alarms and uncorrelated ones.

Chapter 3

Similarity Analysis of Industrial Alarm Flood Sequences *

3.1 Overview

As the state-of-art of similarity analysis of industrial alarm floods, the existing methods suffer from the following limitations: (i) The DTW and SW algorithms in [3, 17] align all alarms even if two alarm floods share almost no common alarm tags; such a computation should be avoided. (ii) The computation complexity of the SW algorithm is too high, which may prevent it from online prediction of upcoming alarm floods. (iii) As important attributes of alarm variables, alarm priorities have not been considered by the SW algorithm yet, while it is an intuitively reasonable choice to weigh more on alarms with higher priorities in the similar alarm sequence alignment. Motivated by addressing the above limitations, this chapter proposes a new local alignment algorithm to find similar alarm flood sequences, based on the basic local alignment search tool (BLAST) formulated in [4, 5]. Comparing with the modified SW algorithm, the proposed algorithm is much faster in computation and provides a higher alignment accuracy for similar alarm sequences. These improvements are owing to the following three novelties: (i) A set-based prematching mechanism is introduced to exclude the comparison between alarm floods with few common alarm tags, and to exclude irrelevant alarm tags in

*A version of this chapter has been published as: Hu, W., Wang, J., & Chen, T. (2016). A local alignment approach to similarity analysis of industrial alarm flood sequences. *Control Engineering Practice*, 55, 13-25. A short version has been published as: Hu, W., Wang, J., & Chen, T. (2015). Fast sequence alignment for comparing industrial alarm floods. In *Proceedings of 9th IFAC Symposium on Advanced Control of Chemical Processes*, Whistler, Canada, 647-652.

order to avoid their distractions on the subsequent alarm sequence alignment. (ii) The seeding and extending steps of the conventional BLAST are adapted for alarm floods, where only regions of high similarities are preserved, so that the searching space is reduced significantly. (iii) A priority-based similarity scoring strategy is developed so that the proposed algorithm is more sensitive to alarms having higher alarm priorities.

3.2 Similarity Scores for Alarm Floods

This section introduces the mathematical representations of alarm floods, and defines a priority-based similarity scoring strategy for comparing two alarm floods.

3.2.1 Representations of Alarm Floods

Alarm floods are composed of a series of chronologically sorted alarms, each of which generally includes a variety of attributes, such as the tag name, alarm identifier, time stamp, alarm priority, and process description [58]. The tag name is the label of a process variable (including both analog and digital variables) associated with an alarm; the alarm identifier describes the alarm type, e.g. PVHI indicates an analog variable exceeding a high alarm limit. A tag name and an identifier jointly compose a unique alarm tag. For instance, if the alarm identifier PVHI is applicable to the tag name 1PT01, then 1PT01.PVHI is a unique alarm tag. The time stamp marks the time instant that an alarm occurs or clears. The alarm priority indicates the importance of an alarm. Hence, an alarm flood X can be described as

$$X =< x_1, x_2, ..., x_M >, \tag{3.1}$$

where the symbol $< \cdot >$ indicates a sequence, the length M is the total number of occurred alarms in X, and the element x_i indicates the i-th alarm occurred in the chronological order. We represent x_i by a tuple with three attributes,

$$x_i = (e_i, t_i, p_i). \tag{3.2}$$

Here e_i is the alarm tag of x_i, and t_i and p_i are the corresponding time stamp and alarm priority, respectively. For the ease of computation, the alarm tag e_i

is better in a numerical form so that we map all distinct alarm tags in words to numerical symbols. Thus, a numerical alphabet can be constructed as

$$\Sigma = \{1, 2, ..., V\}, \tag{3.3}$$

where V represents the size of the alphabet, equal to the total number of distinct alarm tags.

Table 3.1 shows an industrial example of an alarm flood: the first column lists the alarm tags in the chronological order, the second column gives the numerical symbols of alarm tags, the third column indicates the time stamps, and the last column presents the alarm priorities.

Table 3.1: An industrial example of an alarm flood sequence.

Alarm tag	Symbol (e_i)	Time stamp (t_i)	Priority (p_i)
T09.PVLO	9	10/5/2013 02:20:01	Low
T05.PVHI	5	10/5/2013 02:20:55	Low
T06.PVLO	6	10/5/2013 02:22:02	Low
T02.PVHI	2	10/5/2013 02:22:42	High
T01.LOLO	1	10/5/2013 02:23:35	High
T03.PVHI	3	10/5/2013 02:24:22	Low
T04.PVLO	4	10/5/2013 02:25:11	Low
T03.PVHI	3	10/5/2013 02:27:02	Low
T04.PVLO	4	10/5/2013 02:27:03	Low
T07.PVLO	7	10/5/2013 02:27:03	Emergency
T04.PVLO	4	10/5/2013 02:29:05	Low
T05.LOLO	5	10/5/2013 02:30:05	Low

In the analysis of alarm floods, a troublesome issue is the presence of chattering alarms, which occur and clear frequently in a short time span without operators' response. The chattering alarms may lead to false detection of alarm floods. Hence, a prerequisite step for alarm flood analysis is to remove chattering alarms. The detection and removal of chattering alarms have received increasing attentions recently [58, 69, 90, 91]; in particular, the m-sample delay timer with m equal to 20, as recommended by Rule 3a in [91], is applied to remove chattering alarms in the sequel.

3.2.2 Alarm Priority-Based Similarity Scoring Strategy

Alarm priorities indicate the relative importance of alarm tags, based on the seriousness of consequences of ignoring alarms, and the allowable response time to handle abnormalities associated with alarms [47]. Three or four priorities are often adopted, possibly labeled by different names as shown in Table 3.2. As a good practice guide, the numbers of alarm tags assigned with three priorities from lower to higher are recommended to have the 80%/15%/5% distribution [47]. Higher priorities are configured to a smaller number of alarm tags, and they are expected to occur rarely; however, their occurrences indicate the presence of severe abnormalities. In contrast, lower priorities are assigned to most alarms associated with less severe abnormalities. Thus, it is reasonable that the algorithm to be proposed for similar alarm sequence alignment of alarm floods is more sensitive to the occurrences of alarms with higher priorities.

Table 3.2: Alarm priorities in alarm systems.

Alarm priority	List 1	List 2	List 3
Priority 1 (p_1)	Emergency	Emergency	Critical
Priority 2 (p_2)	High	High	Warning
Priority 3 (p_3)	Medium	Low	Advisory
Priority 4 (p_4)	Low		

A basic similarity score $s(x_i, y_j)$ measures the similarity of two alarms $x_i = (e_i^x, t_i^x, p_i^x)$ and $y_j = (e_j^y, t_j^y, p_j^y)$ in two alarm floods X and Y. Depending on whether x_i and y_i have the same alarm tag, the calculation of $s(x_i, y_j)$ is formulated as

$$s(x_i, y_j) = \left\{ \begin{array}{cl} \phi(p_i^x) \text{ (or } \phi(p_j^y)) & \text{if } e_i^x = e_j^y \\ \zeta & \text{if } e_i^x \neq e_j^y \end{array} \right. , \tag{3.4}$$

Here the match score $\phi(\cdot)$ takes positive value and is a function of alarm priority while the mismatch score ζ is a negative value. For a priority list $\{p_1, p_2, ..., p_L\}$ in a descending order of importance, the match score $\phi(p_l)$ should be monotonically deceasing with respect to l, e.g.,

$$\phi(p_l) = \alpha + \beta(L - l), \tag{3.5}$$

where α and β are two positive constants. For the gapped alignment [17, 84], x_i may be aligned with a gap "-" rather than a different alarm tag; in this case, the basic similarity score is equal to a constant gap penalty $\delta < 0$, i.e.,

$$s(x_i, -) = \delta. \tag{3.6}$$

In order to prefer a gapped alignment instead of the alignment of mismatched alarm tags, the mismatch score and the gap penalty should satisfy the following condition,

$$\zeta < 2\delta < 0. \tag{3.7}$$

In this study, we use the linear match score function in (3.5) with parameters $\alpha = 3$ and $\beta = 1.5$, the mismatch score $\zeta = -2.5$, and the gap penalty $\delta = -1$, as summarized in Table 3.3, for all illustrative examples and industrial case studies to be presented later. The choices of the match score, mismatch score, and gap penalty are not unique. The distance between $\phi(p_l), l = 1, 2, ..., L$ is based on the importance of alarm priorities. For instance, given three priorities “Emergency”, “High”, and “Low”, the scores based on Table 3.3 are 6, 4.5, and 3, which indicate that the importance of an “Emergency” alarm is equal to that of two “Low” alarms, and the importance of two “High” alarm is equal to that of three “Low” alarms. The distance between $\phi(p_l), l = 1, 2, ..., L$, can be changed based on the experts’ experience on their assessment of relative importance of different alarm priorities in alarm management. Generally, if $\zeta < 2\delta < 0$ and $\phi(p_l) > |\zeta|, l = 1, 2, ..., L$, are satisfied, the choices of $\phi(p_l)$, ζ, and σ have no significant effects on the final results.

Table 3.3: Basic similarity scores

Item	Symbol	Score
Match for p_l	$\phi(p_l)$	$3 + 1.5(L - l)$
Mismatch	ζ	-2.5
Gap penalty	δ	-1

3.3 Local Alignment of Alarm Floods

In this section, the main idea of the basic local alignment search tool (BLAST) is firstly revisited. Next, a new local alignment algorithm is proposed for similar alarm sequence alignment of alarm floods.

3.3.1 Main Steps of the BLAST

The BLAST locates similar segments between a query sequence and object sequences [4, 5]. The BLAST is composed of two main steps, namely, the seeding and extending steps. In the seeding step, both the query and object sequences are broken into short words having a fixed size. A lookup table is built for all possible assemblies. By indexing short words in the lookup table, similar segments of the query and object sequences are quickly located. The pairs of words with similarity scores exceeding a threshold are kept as seeds while others are discarded. Next, in the extending step, all seeds are extended in two directions until the similarity score falls below a cutoff threshold. The alignments with similarity scores larger than a certain threshold are called the high scoring segment pairs (HSPs). The HSPs with the highest score are treated as the most similar segments between the query and object sequences. An example of alarm sequences is presented here to illustrate the BLAST.

Example 1. With four unique alarms $\Sigma = \{1, 2, 3, 4\}$, a *lookup table* is built for all $k = 3$ short words so that the table has $4^3 = 64$ assemblies. Two alarm sequences <3, 2, 1, 4, 3, 2, 2> and <3, 4, 2, 1, 4, 2> are used for comparison. The first alarm sequence can be broken into <3, 2, 1>, <2, 1, 4>, <1, 4, 3>, <4, 3, 2>, and <3, 2, 2>. The second alarm sequence can also be broken into 4 short words in the same manner. By indexing the lookup table and determining which words from the two sequences have the same index, the matched pairs can be quickly found. As a result, <2, 1, 4> is a completely matched pair and is treated as a seed. In the extending step, the seed <2, 1, 4> is extended in two directions using a gapped extension scheme. By allowing one gap alignment, the best alignment for the two alarm sequences is achieved as

$$\begin{array}{cccccccc} 3 & - & 2 & 1 & 4 & 3 & 2 & 2 \\ | & & | & | & | & & | & \\ 3 & 4 & 2 & 1 & 4 & - & 2 & - \end{array} \tag{3.8}$$

where ‘$-$’ denotes a gap insertion, and ‘$|$’ indicates a matched pair. □

Comparing with the Smith-Waterman algorithm [84], the BLAST does not align all elements in sequences. Its main merit is to achieve fast sequence alignments by locating regions having high similarities and pruning most search space. Such a merit is important, especially for the pairwise alignment of a long sequence with the other sequences in a large query database.

To make the BLAST more suitable for alarm floods, some significant modifications are made as follows to achieve high computational efficiency and better alignment results. First, considering a large variety of alarm tags, a set based matching mechanism is designed as a pre-step to exclude irrelevant alarm floods and alarm tags. Second, the conventional BLAST requires a predefined look up table comprised of all possible short words of a fixed size. The construction and query of such a lookup table become infeasible if there are too many unique components. To solve this problem, the seeding stage is modified to search all matched pairs of different sizes. Third, specialized for alarm floods, a priority based scoring scheme is used to make the algorithm more sensitive to alarms of higher priorities. Finally, the conventional BLAST does not take the time information into account while the alarms are time stamped. Thus, in the extending stage, a time ambiguity tolerance strategy is proposed to make the algorithm less sensitive to orders of alarms occurring almost simultaneously. The details of these modifications are described in the following subsections.

3.3.2 Set-Based Pre-Matching Mechanism

A set-based pre-matching mechanism is proposed here, in order to exclude irrelevant alarm floods and alarm tags. First, to find common alarm tags between alarm flood sequences $X =< x_1, x_2, ..., x_M >$ and $Y =< y_1, y_2, ..., y_N >$, two binary-valued indexing vectors are formulated as

$$\left\{ \begin{array}{lcl} A_X & = & [a_1^x \quad a_2^x \quad \cdots \quad a_M^x] \\ A_Y & = & [a_1^y \quad a_2^y \quad \cdots \quad a_N^y] \end{array} \right. , \tag{3.9}$$

where

$$a_i^x = \left\{ \begin{array}{ll} 1 & \text{if } \exists j \in \{1, 2, ..., N\} \text{ s.t. } e_j^y = e_i^x \\ 0 & \text{otherwise} \end{array} \right. , \tag{3.10}$$

$$a_i^y = \left\{ \begin{array}{ll} 1 & \text{if } \exists j \in \{1, 2, ..., M\} \text{ s.t. } e_j^x = e_i^y \\ 0 & \text{otherwise} \end{array} \right. , \tag{3.11}$$

where e_i^x and e_j^y are the alarm tags of x_i in X and y_j in Y, respectively. Based on the indexing vectors A_X and A_Y, a set-based similarity index is formulated as

$$S_{\text{set}}(X, Y) = \frac{\sum_{i=1}^{M}(a_i^x \times \phi(p_i^x)) \sum_{i=1}^{N}(a_i^y \times \phi(p_i^y))}{\sum_{i=1}^{M}(\phi(p_i^x)) \sum_{i=1}^{N}(\phi(p_i^y))}. \tag{3.12}$$

The set-based similarity index $0 \leq S_{\text{set}}(X, Y) \leq 1$ is used to measure whether two alarm flood sequences are alike without considering the chronological orders. In other words, $S_{\text{set}}(X, Y) = 1$ indicates alarm floods X and Y have the same alarm set, while $S_{\text{set}}(X, Y) = 0$ means that X and Y share no common alarm. If more alarm tags are shared by two alarm floods, the set-based similarity index is closer to 1.

The calculation of $S_{\text{set}}(X, Y)$ is based on alarm sets rather than alarm sequences, and is a quick preliminary step to select suitable alarm floods for subsequent similarity analysis. That is, the set-based similarity index between X and Y is said to be too low, if

$$S_{\text{set}}(X, Y) \leq \gamma, \tag{3.13}$$

where γ is a user-selected threshold. In this case, there is no need to proceed to the next step of sequence alignment for alarm floods. The simplest choice is $\gamma = 0$, indicating that the computation will not proceed if two alarm floods share no common alarms. The value of γ can be set to be larger than 0 if there are a large number of alarm floods, so as to avoid time consumptions in comparing alarm floods with few common alarm tags.

Second, even if two alarm floods X and Y have a set-based similarity index $S_{\text{set}}(X, Y)$ larger than γ, they may still contain many uncommon alarm tags. It is meaningless to align these uncommon alarm tags, and is even distractive to have them in the subsequent sequence alignment. Hence, all irrelevant alarms are removed from the original alarm flood sequences X and Y, and only the common alarm tags are preserved. Based on the indexing vectors A_X and A_Y in eqn. (3.9), the alarms x_i with $a_i^x = 1$ and y_j with $a_j^y = 1$ are extracted from X and Y to construct two shorter sequences $\tilde{X} =< \tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_{\tilde{M}} >$ and $\tilde{Y} =< \tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_{\tilde{N}} >$. Here $\tilde{M} = \sum_{i=1}^{M} a_i^x$ and $\tilde{N} = \sum_{i=1}^{N} a_i^y$ are the numbers of common alarm tags in X and Y, respectively. Clearly, $S_{\text{set}}(\tilde{X}, \tilde{Y}) = 1$, i.e., all alarm tags in $\tilde{X}$ can also be found in $\tilde{Y}$, and vice versa.

Example 2. Two alarm sequences X and Y of different lengths ($M = 12$ and $N = 14$) are given in Table 3.4. By comparing the alarm symbols, two indexing vectors A_X and A_Y are calculated as

$$\left\{ \begin{array}{lcl} A_X & = & [1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1] \\ A_Y & = & [1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 1 \ \ 0 \ \ 1 \ \ 0 \ \ 1 \ \ 0] \end{array} \right. . \tag{3.14}$$

The indexing vector A_X is an all-one vector, indicating that all alarm tags in X can be found in Y. The 10th, 12th, and 14th elements in A_Y are zeros,

Table 3.4: Alarm floods X and Y with time and priority information.

Sequence X			Sequence Y		
Symbol	Time stamp	Priority	Symbol	Time stamp	Priority
9	00:00:01	Low	9	11:00:01	Low
5	00:00:55	Low	5	11:01:11	Low
6	00:02:02	Low	2	11:01:59	High
2	00:02:42	High	1	11:02:18	High
1	00:03:35	High	3	11:03:11	Low
3	00:04:22	Low	4	11:04:01	Low
4	00:05:11	Low	7	11:06:32	Emergency
3	00:07:02	Low	3	11:06:33	Low
4	00:07:03	Low	4	11:06:33	Low
7	00:07:03	Emergency	8	11:07:16	Low
4	00:09:05	Low	9	11:08:45	Low
5	00:10:05	Low	8	11:09:22	Low
			6	11:09:56	Low
			8	11:09:57	Low

owing to the alarm tag with symbol 8 that is not present in X. There are three ($L = 3$) alarm priorities in Table 3.4, namely, "Emergency" (p_1), "High" (p_2), and "Low" (p_3). Based on the priority-based similarity score in Table 3.3, the scores assigned to alarms of the three priorities are $\phi(p_1) = 3 + 1.5 \times (3-1) = 6$, $\phi(p_2) = 3 + 1.5 \times (3-2) = 4.5$, and $\phi(p_1) = 3 + 1.5 \times (3-3) = 3$, respectively. Using eqn. (3.12), the set-based similarity index is calculated as $S_{\text{set}}(X, Y) = \frac{42\times 39}{42\times 48} = 0.8125$. Thus, the set-based pre-matching mechanism says that X and Y are very similar alarm sequences. Next, two subsequences $\tilde{X}$ and $\tilde{Y}$ are obtained as $\tilde{X} =< 9, 5, 6, 2, 1, 3, 4, 3, 4, 7, 4, 5 >$ and $\tilde{Y} =< 9, 5, 2, 1, 3, 4, 7, 3, 4, 9, 6 >$ by removing the alarm tag 8 from Y. The two subsequences will be used in the similar alarm sequence alignment in the following subsections. □

3.3.3 The Seeding Step

The seeding step in the conventional BLAST establishes a lookup table for all possible assemblies of words having a fixed size. Such a lookup table is feasible for biological sequences and text strings, because the numbers of basic components are limited as shown in Table 3.5. However, for industrial alarm systems, the number of distinct alarm tags is quite large, usually larger than 1000. For instance, the number of alarm tags in one industrial plant was reported about 14000 [73]. Thus, it is infeasible to build a lookup table for alarm sequences. For example, if a 3-word table is built for 1000 alarm tags, then the table size would be $1000^3 = 10^9$ that is too large for computation. Hence, the seeding step of the conventional BLAST is not directly applicable to alarm sequences.

Table 3.5: Basic components of different sequences.

	Genome	Protein	Text string	Alarm flood
Component	Nucleotide	Amino acid	Alphabet	Alarm tag
Number of types	5	23	26	> 1000

To solve this problem, the seeding step needs to be modified. The main idea is to find all matched pairs rather than short words of a fixed size. A matched pair $Z(i), i = 1, 2, \cdots, K_b$, is indexed by its positions in two alarm sequences $\tilde{X}$ and $\tilde{Y}$ as

$$\mathbf{z}_i = (z_{i,1}, z_{i,2}, z_{i,3}), \tag{3.15}$$

where $z_{i,1}$ and $z_{i,2}$ are the start positions of the matched pair $Z(i)$ in $\tilde{X}$ and $\tilde{Y}$, respectively; $z_{i,3}$ represents the length of $Z(i)$; K_b indicates the total number of matched pairs. The matched pair $Z(i)$ satisfies two properties as follows. First, all alarm tags of $Z(i)$ in two alarm sequences $\tilde{X}$ and $\tilde{Y}$ should be identical, i.e. $e^x_{z_{i,1}+k} = e^y_{z_{i,2}+k}$ for $k = 0, 1, ..., z_{i,3} - 1$, where $e^x_{z_{i,1}+k}$ and $e^y_{z_{i,2}+k}$ are the alarm tags of the $(z_{i,1}+k)$th and $(z_{i,2}+k)$th alarms in $\tilde{X}$ and $\tilde{Y}$, respectively. Second, any matched pair should not be overlapping with other matched pairs. Given one matched pair $Z(i)$, two conditions $z_{i,1} \in [z_{j,1}, z_{j,1} + z_{j,3} - 1]$ and $z_{i,2} \in [z_{j,2}, z_{j,2} + z_{j,3} - 1]$ cannot be true simultaneously for any $Z(j), j \neq i$.

To select seeds from the matched pairs, the seeding score, namely, the similarity score of each matched pair $Z(i)$ is calculated as

$$h(Z(i)) = \sum_{k=0}^{z_{i,3}-1} s(\tilde{x}_{z_{i,1}+k}, \tilde{y}_{z_{i,2}+k}), \tag{3.16}$$

where $s(\tilde{x}_{z_{i,1}+k}, \tilde{y}_{z_{i,2}+k})$ is obtained from (3.4). The matched pairs having large seeding scores are preserved as seeds, while other pairs are discarded. The seeds and their seeding scores are denoted as $\tilde{Z}(k)$ and $h(\tilde{Z}(k))$ for $k = 1, ..., K$, respectively, where K is the number of seeds selected as the matched pairs of

Table 3.6: The matched alarm pairs.

Matched pair	$z_{i,1}$	$z_{i,2}$	$z_{i,3}$	$h(Z(i))$	Sequences
$Z(1)$	1	1	2	6	$< 9, 5 >$
$Z(2)$	1	10	1	3	$< 9 >$
$Z(3)$	3	11	1	3	$< 6 >$
$Z(4)$	4	3	4	15	$< 2, 1, 3, 4 >$
$Z(5)$	6	8	2	6	$< 3, 4 >$
$Z(6)$	8	5	3	12	$< 3, 4, 7 >$
$Z(7)$	8	8	2	6	$< 3, 4 >$
$Z(8)$	11	6	1	3	$< 4 >$
$Z(9)$	11	9	1	3	$< 4 >$
$Z(10)$	12	2	1	3	$< 5 >$

top K seeding scores. An example is presented here to illustrate the modified seeding step.

Example 3. Based on the two shorter sequences $\tilde{X} =< 9, 5, 6, 2, 1, 3, 4, 3, 4, 7, 4, 5 >$ and $\tilde{Y} =< 9, 5, 2, 1, 3, 4, 7, 3, 4, 9, 6 >$ obtained in Example 2, the matched pairs shown in Table 3.6 are found in the modified seeding step. Taking $\mathbf{z}_4 = (4, 3, 4)$ as an example, the matched pair is $Z(4) =< 2, 1, 3, 4 >$ with 4 alarms, and it starts at the 4th position of $\tilde{X}$ and the 3rd position of $\tilde{Y}$. Using the priority-based similarity score in Table 3.3, the similarity scores of all matched pairs are calculated as shown in the fifth column of Table 3.6. The seeding score of $Z(4)$ is $h(Z(4)) = 4.5 + 4.5 + 3 + 3 = 15$. By choosing $K = 1$, the matched pairs with the largest seeding scores is preserved as the seed, namely, $\tilde{Z}(1) =< 2, 1, 3, 4 >$ with $h(\tilde{Z}(1)) = 15$.

3.3.4 The Extending and Backtracking Steps

Once seeds are obtained, the conventional BLAST extends the seeds in two directions to find the HSPs, as stated in Section 3.3.1. In this subsection, the extending step is adapted to alarm sequences based on the priority-based similarity score in Section 3.2.2. There are two types of extensions, namely, the ungapped extension and the gapped one [4, 5, 56]. In order to have more flexibility in aligning alarm sequences, the gapped extension, which aligns irrelevant components with gaps, is used in this context.

Denote the subsequences of $\tilde{X}$ and $\tilde{Y}$ based on the k-th seed $\tilde{Z}(k)$ for extension in one direction as X_s of length m and Y_s of length n. An extending score matrix H is calculated in three ways as shown in Fig. 3.1. Initially, the first element of H is defined as

$$H_{1,1} = h(\tilde{Z}(k)) + U, \tag{3.17}$$

where $h(\tilde{Z}(k))$ is the seeding score of the seed $\tilde{Z}(k)$ and U is the cutoff threshold used to stop the extension. In principle, the extension grows until the extending score $H_{i,j}$ falls more than a threshold below the maximum extending s-core, namely, $H_{i,j} < H_{max} - U$, where $H_{\max}$ denotes the highest extending score achieved before the extension proceeds to $H_{i,j}$. Initially, $H_{\max} = H_{1,1}$. Thus, the incorporation of U in $H_{1,1}$ avoids the degenerated case that $H_{1,1} < U$.

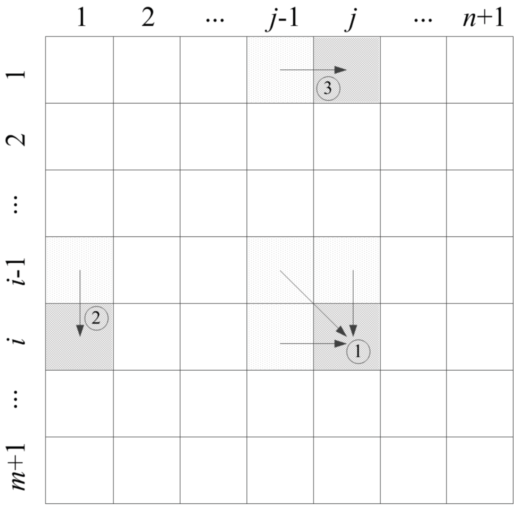


Figure 3.1: The calculation of the extending score matrix H in the gapped extension. The circled numbers 1, 2, and 3 indicate the first, second, and third ways to calculate the element $H_{i,j}$ associated with eqns. (3.18), (3.19), and (3.20).

For all $i = 2, 3, ..., m+1$ and $j = 2, 3, ..., n+1$, the element $H_{i,j}$ on the i-th row and j-th column of the matrix H is obtained as the first way in Fig. 3.1 and is calculated as

$$H_{i,j} = \max\{H_{i-1,j-1} + s(x_i, y_j), H_{i,j-1} + \delta, H_{i-1,j} + \delta, 0\}, \tag{3.18}$$

where $s(x_i, y_j)$ is the basic similarity score (defined in Section 3.2.2) between x_i and y_j with $x_i \in X_s$ and $y_j \in Y_s$. For all $i = 2, 3, ..., m+1$ and $j = 1$, $H_{i,1}$ is calculated as the second way in Fig. 3.1, i.e.,

$$H_{i,1} = \max\{H_{i-1,1} + \delta, 0\}. \tag{3.19}$$

For all $i = 1$ and $j = 2, 3, ..., n+1$, $H_{1,j}$ is calculated as the third way in Fig. 3.1, i.e.,

$$H_{1,j} = \max\{H_{1,j-1} + \delta, 0\}. \tag{3.20}$$

Algorithm 2 Gapped extension

```
$H = \mathbf{0}_{m+1,n+1}$; $Q = \mathbf{0}_{m+1,n+1}$; $H_{1,1} = h(\tilde{Z}(k)) + U$; $H_{\max} = H_{1,1}$;
for $i = 2$ to $m+1$ do
    if $H_{i-1,1} \geq H_{\max} - U$ then
        $H_{i,1} = \max\{H_{i-1,1} + \delta, 0\}$; $Q_{i,1} = 3$;
    end if
end for
for $j = 2$ to $n+1$ do
    if $H_{1,j-1} \geq H_{\max} - U$ then
        $H_{1,j} = \max\{H_{1,j-1} + \delta, 0\}$; $Q_{1,j} = 2$;
    end if
    for $i = 2$ to $m+1$ do
        if $\max\{H_{i-1,j-1}, H_{i,j-1}, H_{i-1,j}\} \geq H_{\max} - U$ then
            $H_{i,j} = \max\{H_{i-1,j-1} + s(x_i, y_j), H_{i,j-1} + \delta, H_{i-1,j} + \delta, 0\}$;
            if $H_{i,j} = H_{i-1,j-1} + s(x_i, y_j)$ then $Q_{i,j} = 1$;
            else if $H_{i,j} = H_{i,j-1} + \delta$ then $Q_{i,j} = 2$;
            else if $H_{i,j} = H_{i-1,j} + \delta$ then $Q_{i,j} = 3$;
            end if
            if $H_{i,j} > H_{\max}$ then $H_{\max} = H_{i,j}$;
            end if
        end if
    end for
    if $\max_{1 \leq i \leq m+1} H_{i,j} < H_{\max} - U$ then break;
    end if
end for
```

According to Fig. 3.1, there are three stop conditions respectively corresponding to the calculations of $H_{i,j}$ in (3.18), (3.19), and (3.20). First, for all $i = 2, 3, ..., m+1$ and $j = 2, 3, ..., n+1$, the extension to $H_{i,j}$ stops at $H_{i,j}$ if

$$\max\{H_{i-1,j-1}, H_{i,j-1}, H_{i-1,j}\} < H_{\max} - U, \tag{3.21}$$

Second, for all $i = 2, 3, ..., m+1$ and $j = 1$, the extension to $H_{i,1}$ stops at $H_{i,1}$ if

$$H_{i-1,1} < H_{\max} - U. \tag{3.22}$$

Third, for all $i = 1$ and $j = 2, 3, ..., n+1$, the extension to $H_{1,j}$ stops at $H_{1,j}$ if

$$H_{1,j-1} < H_{\max} - U. \tag{3.23}$$

The gapped extension in one direction can be described via the pseudo codes in Algorithm 2. The inputs of Algorithm 2 are the subsequences X_s, Y_s and the seeding score $h(\tilde{Z}(k))$, while the output is the extending score matrix H. Once the extension is stopped, the best alignment can be obtained by a backtracking step, which is based on the indicator matrix Q obtained in Algorithm 2. It starts at the element having the maximum score. If $Q_{i,j}$ is equal to 1, 2 or 3, the backtracking proceeds to $Q_{i-1,j-1}$, $Q_{i,j-1}$, or $Q_{i-1,j}$, respectively.

Denote the maximum extending scores in the backward and forward extensions as H_b and H_f, respectively. The final similarity score of the best alignment based on the seed $\tilde{Z}(k)$ is calculated as

$$S(\tilde{Z}(k)) = H_b + H_f - h(\tilde{Z}(k)) - 2U. \tag{3.24}$$

When several seeds are found, the best alignments for these seeds may be different. The subsequences with the highest score is treated as the best alignment. The corresponding score is regarded as the final best score between $\tilde{X}$ and $\tilde{Y}$, denoted by $S(\tilde{X}, \tilde{Y})$.

Example 4. To illustrate the gapped extension, the seed $\tilde{Z}(1) =< 2, 1, 3, 4 >$ with $h(\tilde{Z}(1)) = 15$ is investigated in details here. The basic similarity score in Table 3.3 is used. The extension threshold is set to $U = 2|\delta| = 2$, i.e., two gaps in the alignment are allowed. Thus, $H_{1,1} = h(\tilde{Z}(1)) + U = 15 + 2 = 17$. The seed $\tilde{Z}(1) =< 2, 1, 3, 4 >$ is firstly extended to the forward direction subsequences $X_f =< 3, 4, 7, 4, 5 >$ and $Y_f =< 7, 3, 4, 9, 6 >$. The extending score matrix H for X_f and Y_f is generated as Table 3.7. The backtracking in the forward extension yields the best alignment, as shown by the alarm tags with scores marked by underlines in Table 3.7. The seed $\tilde{Z}(1) =< 2, 1, 3, 4 >$ is secondly extended to the backward direction subsequences $X_b =< 6, 5, 9 >$ and $Y_b =< 5, 9 >$. The extending score matrix H for X_b and Y_b is obtained as Table 3.8. Fig. 3.2 presents the extensions in two directions based on the seed $\tilde{Z}(1)$. The green squares indicate the matched alarm pairs and the grey squares represent the entries where the extension proceeds. It is obvious that most search space marked by the white squares is pruned in the extensions, so that the computation cost is greatly reduced.

Combining the extension in the two directions, the best alignment is finally

Table 3.7: Extending score matrix H in the forward extension based on the seed $\tilde{Z}(1) =< 2, 1, 3, 4 >$.

		7	3	4	9	6
	$\underline{17}$	16	0	0	0	0
3	$\underline{16}$	15	0	0	0	0
4	$\underline{15}$	14	0	0	0	0
7	14	$\underline{21}$	$\underline{20}$	19	0	0
4	0	20	19	$\underline{23}$	22	21
5	0	19	18	22	21	20

Table 3.8: Extending score matrix H in the backward extension based on the seed $\tilde{Z}(1) =< 2, 1, 3, 4 >$.

		5	9
	$\underline{17}$	16	0
6	$\underline{16}$	15	0
5	15	$\underline{19}$	18
9	14	18	$\underline{22}$

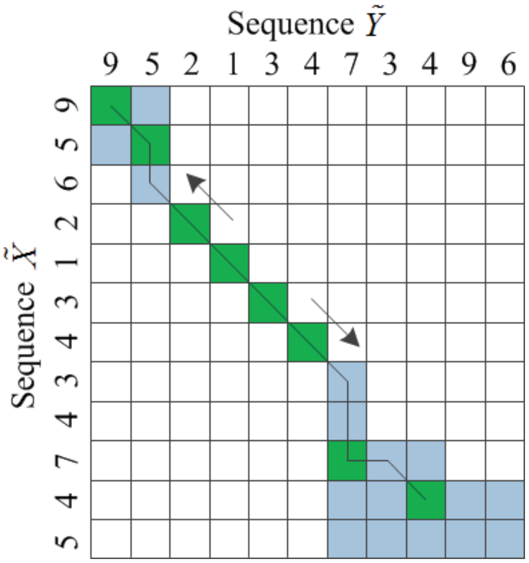


Figure 3.2: Graphic illustration of the gapped extension.

achieved as

$$\begin{array}{ccccccccccccc} \tilde{X}: & 9 & 5 & 6 & 2 & 1 & 3 & 4 & 3 & 4 & 7 & - & 4 \\ & | & | & & | & | & | & | & & & | & & | \\ \tilde{Y}: & 9 & 5 & - & 2 & 1 & 3 & 4 & - & - & 7 & 3 & 4 \end{array} \tag{3.25}$$

In particular, the alarm tag 7 having a higher alarm priority rather than the alarm tag 3 is found as the matched alarm tag. The priority-based basic similarity score in Section 3.2.2 makes the Algorithm 2 more sensitive to alarms with higher priorities. Based on $\tilde{Z}(1)$, the maximum scores of the backward and forward extensions are $H_b = 22$ and $H_f = 23$, respectively. Then, the final score is $S(\tilde{Z}(1)) = 22 + 23 - 15 - 2 \times 2 = 26$.

3.3.5 Time Ambiguity Tolerance

Physically connected alarm tags often arise almost simultaneously, but they may have different chronological orders in different alarm floods. Thus, a strategy of incorporating time information of alarms was proposed to tolerate certain time ambiguity of these alarms occurring almost simultaneously [17]. This strategy is adopted here in the proposed algorithm as follows. To redefine the basic similarity score $s(x_i, y_j)$ in (3.18), two vectors are defined, namely, the time distance vector $\mathbf{d}_i$ and the time weight vector $\mathbf{w}_i$. The distance vector $\mathbf{d}_i$ for the ith alarm x_i in $\tilde{X}$ is formulated as

$$\mathbf{d}_i = [d_{i1}\ d_{i2}\ \cdots\ d_{iV}]^T\,, \tag{3.26}$$

where

$$d_{iv} = \min_{1 \le k \le m} \left\{ |t_i - t_k| : e_k^x = v \right\}. \tag{3.27}$$

Here e_k^x is the alarm tag of the k-th alarm x_k in $\tilde{X}$, and v is the numerical symbol in the alarm tag alphabet Σ of size V defined in (3.3). Each entry d_{iv} represents the time gap between the ith alarm and the nearest alarm with alarm tag v. The time weighting vector $\mathbf{w}_i$ for x_i in $\tilde{X}$ is

$$\mathbf{w}_i = [w_{i1}\ w_{i2}\ \cdots\ w_{iV}]^T\,, \tag{3.28}$$

where $w_{iv} = f(d_{iv})$. The weighting function $f(\cdot)$ for $\tilde{X}$ can be chosen as the scaled Gaussian function [17],

$$f(d_{iv}) = e^{-\frac{d_{iv}^2}{2\sigma^2}}\,, \tag{3.29}$$

while the counterpart for $\tilde{Y}$ is determined as

$$f(d_{iv}) = \begin{cases} 1 & \text{if } d_{iv} = 0 \\ 0 & \text{if } d_{iv} \neq 0 \end{cases}. \tag{3.30}$$

Here the standard deviation σ in (3.29) is a user selected parameter. The value of σ decides how much the time difference can be tolerated. In general, the time difference between simultaneously occurred alarms is in the level of a few seconds. If a large value of σ is used, then the orders of two alarms with a large time difference are regarded as being not important. Eventually, the basic similarity score $s(x_i, y_j)$ in eqn. (3.4) is reformulated as

$$s(x_i, y_j) = \zeta + (\phi(p_j^y) - \zeta) \max_{1 \leq v \leq V} w_{iv}^x w_{jv}^y, \tag{3.31}$$

where w_{iv}^x and w_{jv}^y are the time weights for the ith alarm x_i in $\tilde{X}$ and and jth alarm y_j in $\tilde{Y}$, respectively. The gap penalty is preserved the same as eqn. (3.6).

Table 3.9: Extending score matrix H of $\tilde{Z}(1)$ in the forward extension with time stamp information.

		7	3	4	9	6
	$\underline{17}$	16	0	0	0	0
3	16	$\underline{19.3537}$	19	0	0	0
4	15	18.3537	$\underline{22.3537}$	22	0	0
7	14	21	23.3549	$\underline{27.3549}$	26.3549	25.3549
4	0	20	24	26.3549	25.3549	24.3549
5	0	19	23	25.3549	24.3549	23.3549

Example 5. From Table 3.4, it is obvious that the alarm tags at the 8th, 9th and 10th positions in X have very close time stamps. The same case can be found at the 7th, 8th, and 9th positions in Y; the chronological orders for these alarms should be not important in the sequence alignment. Thus, the time ambiguity tolerance strategy is adopted. The standard deviation of the Gaussian function is set as $\sigma = 2$. By using the reformulated similarity score $s(x_i, y_j)$ in (3.31), the score matrix H for the gapped extension of $\tilde{Z}(1) =< 2, 1, 3, 4 >$ in the forward direction sequences X_f and Y_f is obtained in Table 3.9. In the backward extension, the best alignment is the same as that in Table 3.8. Combining the extensions in two directions, the final best alignment is obtained as

$$\begin{array}{rcccccccccc} \tilde{X}: & 9 & 5 & 6 & 2 & 1 & 3 & 4 & \underline{3} & \underline{4} & \underline{7} \\ & | & | & & | & | & | & | & | & | & | \\ \tilde{Y}: & 9 & 5 & - & 2 & 1 & 3 & 4 & \underline{7} & \underline{3} & \underline{4} \end{array} \tag{3.32}$$

Here the alarm tags marked with underlines are not exactly matched, but are treated as matched pairs owing to their close time stamps. For the gapped extension based on $\tilde{Z}(1)$, the maximum scores for the backward and forward extensions are $H_b = 22$ and $H_f = 27.3549$, respectively, so that the best score is $S(\tilde{Z}(1)) = 22 + 27.3549 - 15 - 2 \times 2 = 30.3549$.

3.3.6 The Steps of the Local Alignment Algorithm

The proposed local alignment algorithm is composed of the following steps, which are also depicted as the flowchart in Fig. 3.3.

1. Two alarm floods X and Y are prepared with chattering alarms removed.

2. The set-based pre-matching mechanism is exploited for X and Y, where $S_{\text{set}}(X, Y)$ in (3.12) is calculated. If $S_{\text{set}}(X, Y) > \gamma$, then the subsequences $\tilde{X}$ and $\tilde{Y}$ are obtained from X and Y by preserving common alarm tags in both X and Y; otherwise, the computation is terminated.

3. In the seeding step, the matched segments between $\tilde{X}$ and $\tilde{Y}$ are indexed by $Z = (z_1, z_2, z_3)$. Their seeding scores $h(Z)$ in (3.16) are calculated. The segments having larger seeding scores are preserved as seeds $\tilde{Z}(k), k = 1, ...K$, while others are discarded. The seeding score of each seed is denoted as $h(\tilde{Z}(k))$.

4. The gapped extension is implemented to each seed $\tilde{Z}(k), k = 1, ...K$ using Algorithm 2. In each iteration, $s(x_i, y_j)$ is computed based on (3.26)-(3.31), $H_{i,j}$ is calculated based on (3.18)-(3.20), and $Q_{i,j}$ is recorded. The extension is terminated if the stop conditions (3.21)-(3.23) are satisfied. The extending score for the best alignment of $\tilde{Z}(k)$ is calculated using (3.24).

5. Find the aligned subsequence having the highest score as the final best sequence alignment of X and Y. To measure the computational accuracy and efficiency for comparing X and Y, two quantities are defined as: the number of matched pairs between X and Y, denoted as $C(X, Y)$, is counted to measure the alignment accuracy, and the computation time T is recorded for applying the proposed algorithm to X and Y.

The proposed algorithm has a much lower computational cost than the modified SW algorithm. Given two alarm floods X and Y containing M and N alarms respectively, the implementation of the modified SW algorithm consists

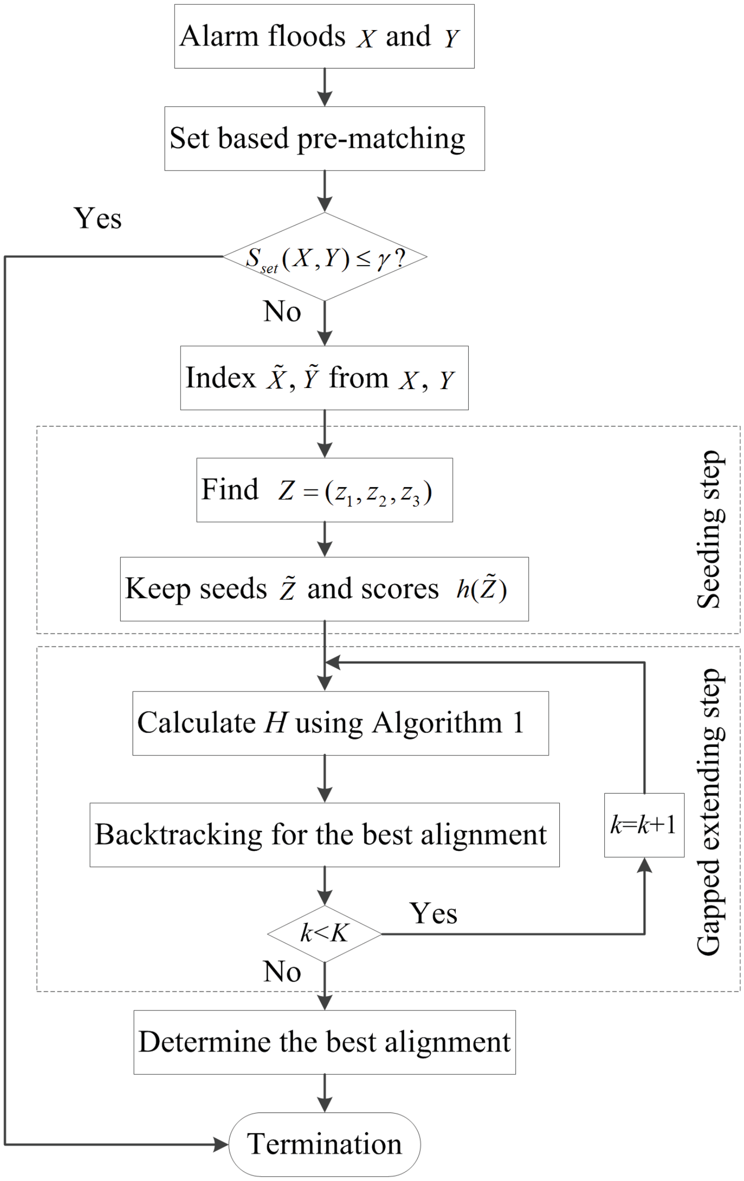

Alarm floods X and Y
Set based pre-matching
$S_{set}(X,Y) \le \gamma$?
Yes
No
Index $\tilde{X}, \tilde{Y}$ from X, Y
Seeding step
Find $Z=(z_1,z_2,z_3)$
Keep seeds $\tilde{Z}$ and scores $h(\tilde{Z})$
Gapped extending step
Calculate H using Algorithm 1
Backtracking for the best alignment
$k=k+1$
$k<K$
Yes
No
Determine the best alignment
Termination

Figure 3.3: Flowchart of the proposed local alignment algorithm.

of two steps: (a) the dynamic programming step calculates a score matrix H for all $M \times N$ elements; (b) the backtracking step finds the best alignment based on an indicator matrix associated with H. Thus, the computational complexity of the modified SW algorithm is $O(2MN)$.

In the proposed algorithm, the set-based pre-matching mechanism decreases the number of computations by excluding irrelevant alarm floods and alarm tags. The computation of this step is a constant time operation $O(1)$. If the set-based similarity index is lower than the threshold γ, the computation is terminated. If the set-based similarity index is larger than the threshold, two shorter alarm sequences $\tilde{X}$ and $\tilde{Y}$ are obtained (denote the length difference between X and $\tilde{X}$ as Δ_1, and the counterpart between Y and $\tilde{Y}$ as Δ_2.). The computation proceeds to the sequence alignment, which consists of a seeding step and an extending step: (a) The seeding step includes the indexing of all matched pairs $Z(i), i = 1, 2, \cdots, K_b$, and the calculation of each similarity score $h(Z(i))$. The indexing of all matched pairs is a constant time operation $O(1)$. Since the matched pairs can be found from either $\tilde{X}$ or $\tilde{Y}$, the maximum computational complexity should be $O(M - \Delta_1)$ or $O(N - \Delta_2)$. Thus the total computational complexity of the seeding step is $O(1 + M - \Delta_1)$ or $O(1 + N - \Delta_2)$. (b) In the extending step, the computational complexity for the dynamic programming and the backtracking with stopping strategy is $O(2(M - \Delta_1)(N - \Delta_2)\eta)$. Here $\eta \in (0, 1]$ indicates the ratio of cells to be calculated, and the value of η depends on the number of seeds K and the extension threshold U. Thus, the total computational complexity of the proposed method is $O(2 + M - \Delta_1 + 2(M - \Delta_1)(N - \Delta_2)\eta)$ or $O(2 + N - \Delta_2 + 2(M - \Delta_1)(N - \Delta_2)\eta)$.

3.3.7 Parameter Determination

The seeding step of the proposed local alignment algorithm finds all the matched alarm segments, among which the segments having the K largest seeding scores are preserved as seeds while others are discarded. In the extending step, the gapped extension stops till the extending score falls more than the threshold U below the maximum score. Thus, K and U are two important parameters influencing the computational cost and the alignment accuracy. To investigate the effects of K and U on the proposed algorithm, Monte Carlo simulations are implemented in the following example.

Example 6. 1000 Monte Carlo simulations are implemented. In each simulation, a pair of symbolic sequences X_l and $Y_l, l = 1, 2, ..., 1000$ are randomly generated. The symbolic tags come from a numeric alphabet $\Sigma = \{1, 2, ..., V\}$, where three priorities $\{p_1, p_2, p_3\}$ with a distribution 5%/15%/80% are assigned to V symbolic tags in Σ. The basic similarity scores in Table 3.3 are used. The sequence lengths L_x and L_y are uniform random integers in the range $[L_1, L_2]$. The time stamps $t_i^x, i = 1, 2, ..., L_x$ and $t_j^y, j = 1, 2, ..., L_y$ of symbolic tags in X and Y are Gaussian random variables in the range $[1, t_{max}]$, where t_{max} is a uniform variable in $[t_{max,1}, t_{max,2}]$. The parameters are set as $V = 100$, $L_1 = 50$, $L_2 = 500$, $t_{max,1} = 4320$ sec (0.05 day), $t_{max,2} = 86300$ sec (1 day). The algorithm is carried out at a personal computer with 3.3GHz CPU, 4G RAM and 64bit operating system. The number $C(X, Y)$ of matched pairs between X and Y is counted to measure the alignment accuracy, and the computation time T in applying the proposed algorithm to X and Y is recorded. Fig. 3.4 presents the average numbers of matched pairs for 1000 pairs of X_l and Y_l, i.e., $\bar{C}(X_l, Y_l) := \frac{1}{1000}\sum_{l=1}^{1000} C(X_l, Y_l)$, for different values of K and U. Fig. 3.5 is the counterpart of Fig. 3.4 for the average computation time $\bar{T} := \frac{1}{1000}\sum_{l=1}^{1000} T_l$.

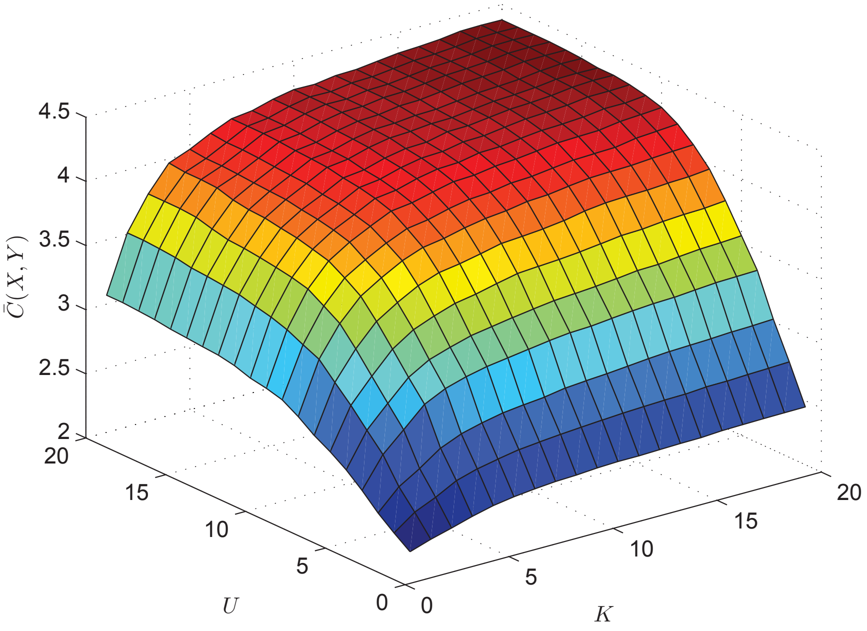


Figure 3.4: Average number of matched pairs $\bar{C}(X, Y)$ versus K and U.

The following observations can be obtained from Figs. 3.4 and 3.5. Given a fixed number $K$of seeds $\bar{C}(X, Y)$ drastically increases with the increment of U for $U \leq 10$, and does not significantly improve with the increment of U for $U > 10$. Given a fixed cutoff threshold U, similar observation can be drawn with $K \leq 7$ and $K > 7$. It is obvious that the increment of the two parameters increases the computation time. To achieve an accurate alignment result within a short time period, U and K are recommended as $U = 10$ and $K = 7$.

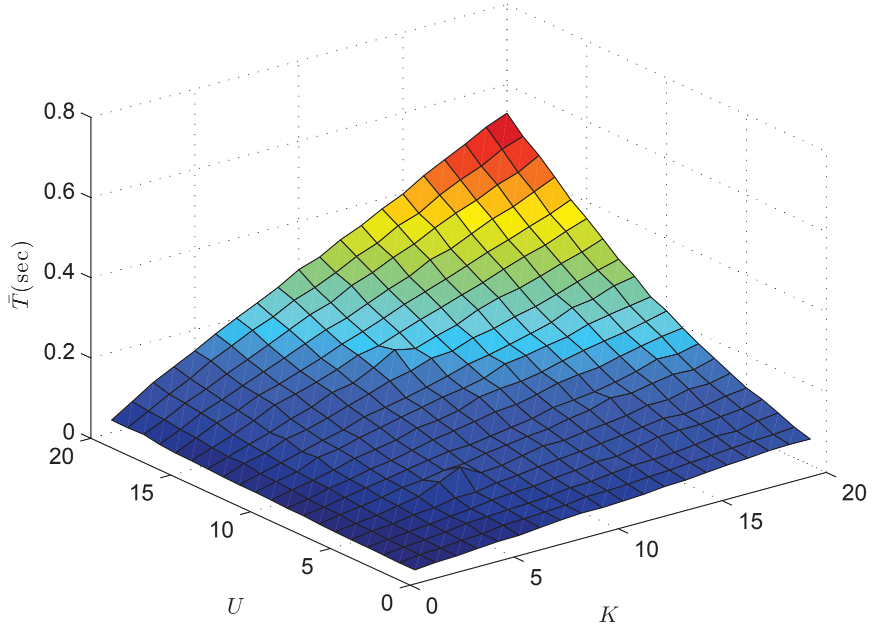


Figure 3.5: Average computation time $\bar{T}$ versus K and U.

3.4 Industrial Case Studies

This section provides industrial case studies to validate the performance of the proposed method. In Case I, multiple alarm floods are studied for multiple queries to a historical sequence database. In Case II, a detailed analysis for the query of one particular alarm flood is carried out.

3.4.1 Case I

The industrial process for case studies is an oil conversion plant, which is quite common in the petrochemical industry. It transforms the crude oil into valuable petroleum products, such as naphtha, gasoline, and diesel fuel [6]. The oil conversion plant is comprised of a number of interconnected pipelines, heaters, reactors, separators and vessels, as shown in Fig. 3.6 [23]. The crude oil and the hydrogen gas are heated to certain temperatures in the feed heater and hydrogen heater, and are contacted with a catalyst bed in the two reactors. The gas and liquid effluent overflows from the reactors to a common outlet line directed to the separators, where the vapor, liquid hydrocarbon, and water phases are separated. The liquid hydrocarbon streams are carried to the atmospheric tower and fractionated into desired products, which are stripped by gas oil strippers. The refining processes are operated by a distributed control system. To ensure the safe operation of the oil conversion plant, a large number of alarm variables are configured in different sections.

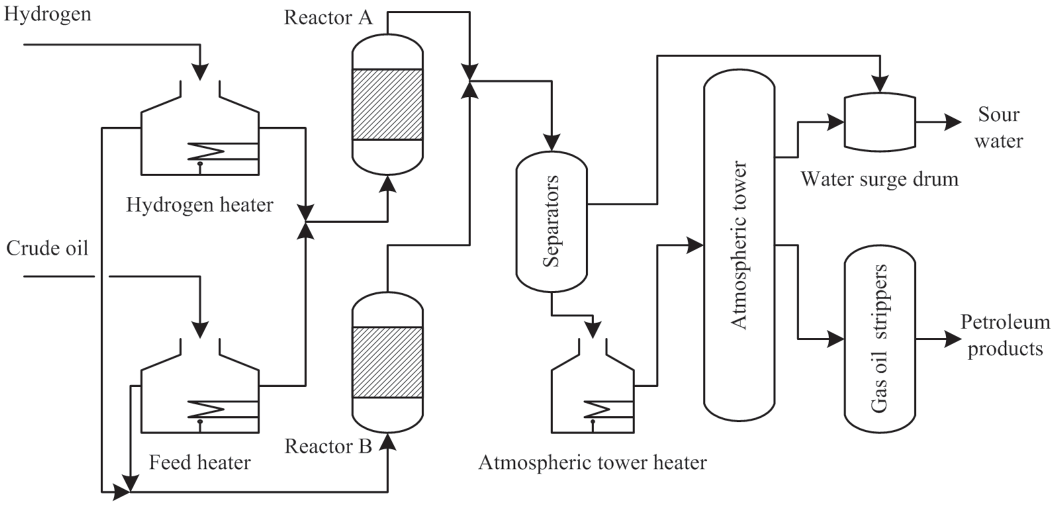


Figure 3.6: Simplified schematic diagram of the oil conversion plant.

Some historical alarm data with 1547 unique alarms and 3 alarm priority levels was collected from the oil conversion plant over the time period from May 12th, 2013 to November 12th, 2014. 389 alarm floods excluding chattering alarms were detected and saved as an database of alarm floods $\mathbb{B} = \{Y_l : l = 1, 2, ..., 389\}$. The longest alarm flood contains 1840 alarms while the shortest alarm flood includes 10 alarms. The average length of alarm floods is 69.2.

To illustrate the efficiency of the proposed local alignment algorithm, 10 alarm floods X_i for $i = 1, ..., 10$ occurred at different time periods are studied for the queries to the database $\mathbb{B}$. The number of alarms in X_i is given in the second column of Table 3.10. First, using the set-based pre-matching mechanisms with $\gamma = 0$, irrelevant object alarm floods are excluded from the database $\mathbb{B}$, while others are preserved for sequence alignment. The number of relevant object alarm floods for each query alarm flood X_i is listed in the third column of Table 3.10. Second, both the proposed algorithm and the modified SW algorithm [17] are applied to the query of X_i from the preserved object alarm floods. The same parameters $U = 10$ and $K = 7$ as those in Example 6 are used to implement the proposed algorithm. The alignment accuracy measured by the average number of matched pairs $\bar{C}(X,Y) := \frac{1}{389}\sum_{l=1}^{389} C(X_i, Y_l)$ and the total computation time T for querying X_i are presented in the last four columns of Table 3.10. Since alarm priorities are not considered in the modified SW algorithm, a unified match score rather than the priority based scoring strategy is utilized. That is, the match score 4.5, the average value of the match scores in Table 3.3, is used for all matched alarm tags.

Table 3.10: The alignment accuracy measured by the average number of the matched pairs $\bar{C}(X,Y)$ and the total computation time T in querying $\mathbb{B}$ for X_i ($i = 1, ..., 10$). $L(X_i)$ indicates the number of alarms contained in X_i. $N(Y)$ denotes the number of relevant alarm floods Y found in $\mathbb{B}$.

Index	$L(X_i)$	$N(Y)$	Proposed algorithm		Modified SW algorithm	
			$\bar{C}(X,Y)$	T(sec)	$\bar{C}(X,Y)$	T(sec)
1	1010	130	6.65	8.32	3.48	924.62
2	848	341	7.07	11.52	4.73	644.69
3	661	382	8.21	10.02	4.60	419.06
4	420	291	6.91	5.53	3.73	239.02
5	347	327	7.65	7.63	3.76	203.51
6	290	305	5.91	4.83	3.29	162.16
7	249	320	5.71	5.66	3.20	176.37
8	196	223	6.75	3.53	4.77	123.86
9	186	276	6.19	4.42	3.42	99.12
10	114	125	5.49	1.86	3.14	50.37

Comparing the average numbers of the matched pairs in the 4th and 6th columns of Table 3.10, it is obvious that the proposed algorithm achieves more accurate alarm sequence alignments than the modified SW algorithm. That is, more matched pairs are found using the proposed algorithm. This improvement is owing to the utilization of the set-based pre-matching mechanism that removes irrelevant alarm tags. Another comparison of the computation times in the 5th and 7th columns of Table 3.10 clearly says that the proposed algorithm is much more efficient in computation. In this case study, the proposed algorithm is about 20 $\sim$ 111 times faster than the modified SW algorithm.

3.4.2 Case II

A detailed analysis of the sequence alignment for the 10th query alarm flood in Case I is presented here. As shown in Table 3.10, this alarm flood X contains 114 alarms. 125 object alarm floods from $\mathbb{B}$ are found containing common alarms in X. For the proposed algorithm and the modified SW algorithm, the number of matched pairs $C(X, Y_i)$ and computation time T_i for X and $Y_i, i = 1, 2, .., 125$ are shown as the dotted lines in Fig. 3.7. In particular, for the queries to the 3rd, 4th, 36th, 38th, and 55th object alarm floods, the computation times of the proposed algorithm are much smaller than the modified SW algorithm; in addition, the alignment accuracies of the proposed algorithm are higher.

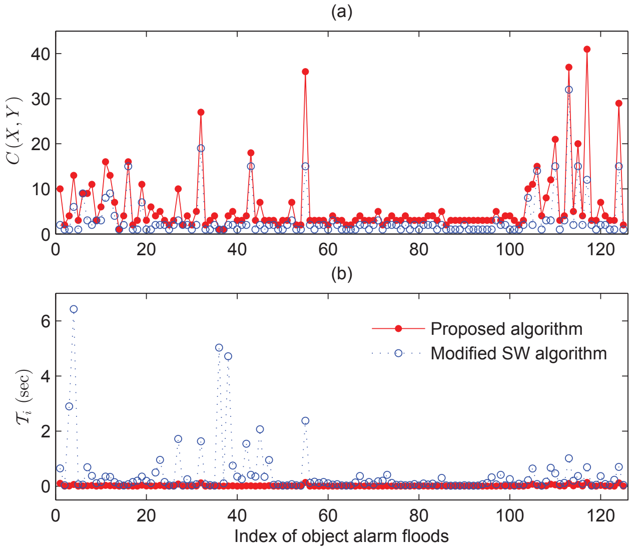


Figure 3.7: Performance statistics of the proposed local alignment algorithm (dot) and the modified SW algorithm (circle-dash) for the query of X to 125 object alarm floods: (a) the number of matched pairs; (b) the computation time.

As detected by the proposed algorithm, the 117th object alarm flood Y as shown in Fig. 3.7 was found to be the most similar to X. The best alignment between X and Y was determined by the backtracking step in Algorithm 1. Excluding irrelevant alarms, the alignment of matched alarms is shown in Fig 3.9. The alarms in the two alarm floods X and Y were chronologically sorted while the occurrence dates of X and Y were different. Among the 40 matches, 30 pairs of alarms were perfectly matched with the same chronological order while other 10 pairs were matched in disorder. Observing the time stamps of these disordered matches, it can be found that these alarms occurred almost simultaneously. Thus, their orders in the alignment should be not important.

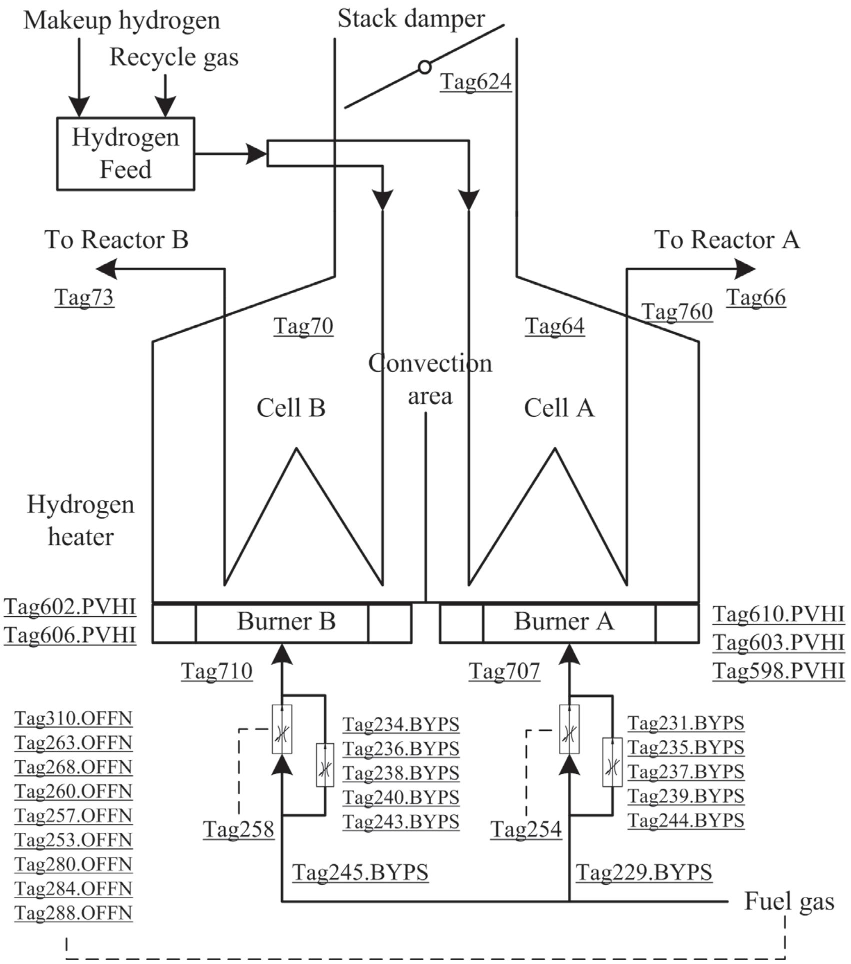
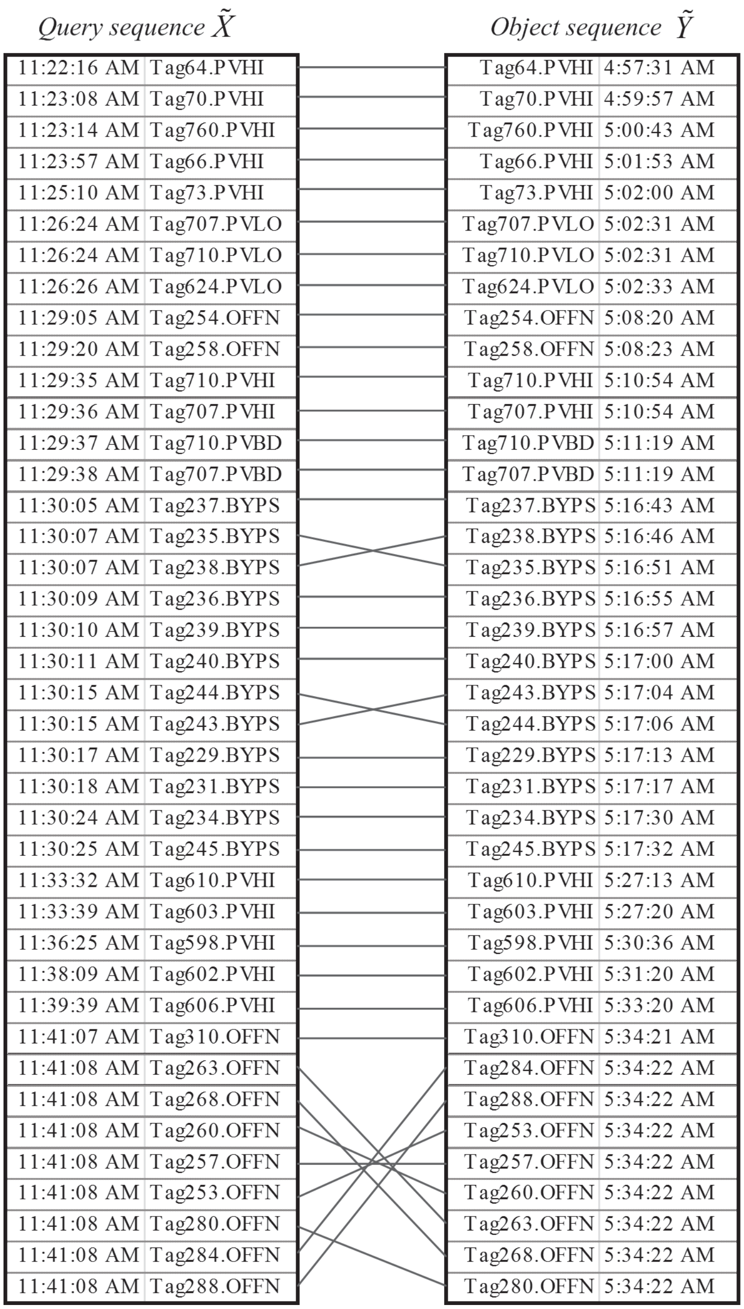


Figure 3.8: Simplified schematic diagram of the hydrogen heater (the dashed lines indicate data links).

The alarm flood Y appeared due to an inadequate feed of the purified recycle gas for one of the two hydrogen supplies in Fig. 3.6. For a clearer illustration, the simplified schematic diagram of the hydrogen heater is presented in Fig 3.8. The inadequate feed of supply led to the overheat of the convection skin of outlets. To maintain the outlet temperatures within their designed ranges, a series of automatical regulations and protective strategies were activated. Based on the knowledge about the cause of Y from the alarm flood database $\mathbb{B}$, one might predict that the alarm flood X was caused by the inadequate feed of hydrogen supply. By checking the historical records, it was found that the supply of the makeup hydrogen (one of the two hydrogen supplies) was decreased, which supported the similarity between X and Y.



Query sequence $\tilde{X}$

11:22:16 AM	Tag64.PVHI
11:23:08 AM	Tag70.PVHI
11:23:14 AM	Tag760.PVHI
11:23:57 AM	Tag66.PVHI
11:25:10 AM	Tag73.PVHI
11:26:24 AM	Tag707.PVLO
11:26:24 AM	Tag710.PVLO
11:26:26 AM	Tag624.PVLO
11:29:05 AM	Tag254.OFFN
11:29:20 AM	Tag258.OFFN
11:29:35 AM	Tag710.PVHI
11:29:36 AM	Tag707.PVHI
11:29:37 AM	Tag710.PVBD
11:29:38 AM	Tag707.PVBD
11:30:05 AM	Tag237.BYPS
11:30:07 AM	Tag235.BYPS
11:30:07 AM	Tag238.BYPS
11:30:09 AM	Tag236.BYPS
11:30:10 AM	Tag239.BYPS
11:30:11 AM	Tag240.BYPS
11:30:15 AM	Tag244.BYPS
11:30:15 AM	Tag243.BYPS
11:30:17 AM	Tag229.BYPS
11:30:18 AM	Tag231.BYPS
11:30:24 AM	Tag234.BYPS
11:30:25 AM	Tag245.BYPS
11:33:32 AM	Tag610.PVHI
11:33:39 AM	Tag603.PVHI
11:36:25 AM	Tag598.PVHI
11:38:09 AM	Tag602.PVHI
11:39:39 AM	Tag606.PVHI
11:41:07 AM	Tag310.OFFN
11:41:08 AM	Tag263.OFFN
11:41:08 AM	Tag268.OFFN
11:41:08 AM	Tag260.OFFN
11:41:08 AM	Tag257.OFFN
11:41:08 AM	Tag253.OFFN
11:41:08 AM	Tag280.OFFN
11:41:08 AM	Tag284.OFFN
11:41:08 AM	Tag288.OFFN

Object sequence $\tilde{Y}$

Tag64.PVHI	4:57:31 AM
Tag70.PVHI	4:59:57 AM
Tag760.PVHI	5:00:43 AM
Tag66.PVHI	5:01:53 AM
Tag73.PVHI	5:02:00 AM
Tag707.PVLO	5:02:31 AM
Tag710.PVLO	5:02:31 AM
Tag624.PVLO	5:02:33 AM
Tag254.OFFN	5:08:20 AM
Tag258.OFFN	5:08:23 AM
Tag710.PVHI	5:10:54 AM
Tag707.PVHI	5:10:54 AM
Tag710.PVBD	5:11:19 AM
Tag707.PVBD	5:11:19 AM
Tag237.BYPS	5:16:43 AM
Tag238.BYPS	5:16:46 AM
Tag235.BYPS	5:16:51 AM
Tag236.BYPS	5:16:55 AM
Tag239.BYPS	5:16:57 AM
Tag240.BYPS	5:17:00 AM
Tag243.BYPS	5:17:04 AM
Tag244.BYPS	5:17:06 AM
Tag229.BYPS	5:17:13 AM
Tag231.BYPS	5:17:17 AM
Tag234.BYPS	5:17:30 AM
Tag245.BYPS	5:17:32 AM
Tag610.PVHI	5:27:13 AM
Tag603.PVHI	5:27:20 AM
Tag598.PVHI	5:30:36 AM
Tag602.PVHI	5:31:20 AM
Tag606.PVHI	5:33:20 AM
Tag310.OFFN	5:34:21 AM
Tag284.OFFN	5:34:22 AM
Tag288.OFFN	5:34:22 AM
Tag253.OFFN	5:34:22 AM
Tag257.OFFN	5:34:22 AM
Tag260.OFFN	5:34:22 AM
Tag263.OFFN	5:34:22 AM
Tag268.OFFN	5:34:22 AM
Tag280.OFFN	5:34:22 AM

Figure 3.9: The best sequence alignment between the query alarm flood X and the 117th Y from $\mathbb{B}$.

3.5 Summary

Some alarm sequence alignment algorithms have been developed recently to discover useful patterns from alarm floods. However, the low computation efficiency prevents them from large-scale practical applications or online computation with demanding requirements on computation speeds. In view of this, a new local alignment algorithm was proposed. Compared with the Smith-Waterman algorithms, the proposed algorithm is much faster in computation and achieves higher alignment accuracy. These advantages are owing to three novelties in the proposed algorithm, namely, the priority-based similarity scoring strategy, the set-based pre-matching mechanism, and the modified seeding-extending steps. The performance of the proposed algorithm was validated by industrial case studies. With the proposed algorithm, it is ready for further pursuing the predication and prevention of alarm floods based on similar alarm sequences in historical alarm floods.

Chapter 4

Cause and Effect Analysis of Industrial Alarm Signals Using Modified Transfer Entropies*

4.1 Overview

In general, a large-scale industrial plant contains thousands of process variables, but not all of them are configured with analog sensors. Alarms are configured over the entire plant to ensure safe process operation and regulatory compliance. Some of them are related to analog measurements while many others are digital by nature, such as instrumental malfunctions, input-output failures, communication errors, and violation of safety conditions [28]. Thus, alarm data should be an important resource to track the propagation paths of abnormalities. Motivated by the above discussion, this chapter proposes a causality inference method based on binary-valued alarm data to identify abnormality propagation paths and detect root causes, in the absence of continuous-valued process data. The main novelties of the proposed methods are: 1) a modified transfer entropy (TE) and a direct transfer entropy (DTE) are defined and formulated as the basic statistical metrics for the proposed method, driven by two characteristics of alarm signals, namely, the random occurrence delays and the mutual independence of alarm occurrences; 2) a statistical test based on surrogate alarm signals is used to determine the significance thresholds for the normalized values of TE and DTE that represent

*A version of this chapter has been submitted for publication as: Hu, W., Wang, J., Chen, T., & Shah, S. L. (2016). Cause and effect analysis of industrial alarm signals using modified transfer entropies. *Control Engineering Practice*.

the causal strength between alarm signals. The proposed method provides an offline cause and effect analysis, which is valuable in assisting plant operators in the online diagnosis of root causes of abnormalities. A detected relation among cause and effect variables implies statistical regularities in historical data sets. These statistical regularities can be exploited to infer the current relationships between the alarms of interests. By doing so, industrial plant operators are able to make a more reliable judgement on how the current abnormality propagates from one variable to another and relate it to experiences from similar abnormal events in the past. The causality relations constructed may also reveal fault propagation pathways, and can be used to determine root causes in alarm floods, which is an important step for the prediction and prevention of alarm floods as stated in [42, 43].

4.2 Transfer Entropy of Alarm Data

This section defines the modified transfer entropy (TE) and direct transfer entropy (DTE), as well as their normalized counterparts NTE and NDTE.

4.2.1 Transfer Entropy of Alarm Signals

The transfer entropy (TE) is a nonparametric method proposed in [80] to measure temporal asymmetric information transfer between variables and thus identify cause and effect signals. In recent years, it has become a popular method for measuring causal influences in multivariate time series [37, 55]. Compared with the structural methods, such as the Ganger causality, the TE is capable of detecting causality for either linear or nonlinear processes, whereas the Ganger causality may only work for linear processes. If the Ganger causality is considered, we would need to formulate a linear model for binary-valued alarm data, which is an unsolved open problem. In addition, the process related to alarms may be nonlinear. Thus, the TE approach is adopted.

Given two continuous-valued signals $\tilde{X}$ and $\tilde{Y}$, let them be sampled at time instants t and denoted by $\tilde{x}_t \in \tilde{X}$ and $\tilde{y}_t \in \tilde{Y}$ with $t = 1, 2, \cdots, N$, where

N denotes the number of samples. The TE from $\tilde{X}$ to $\tilde{Y}$ is calculated as

$$\begin{aligned} T_{\tilde{X}\rightarrow\tilde{Y}} &= \int f(\tilde{y}_{t+1}, \tilde{\mathbf{y}}_t^{(k)}, \tilde{\mathbf{x}}_t^{(l)}) \\ &\quad \cdot \log_2 \frac{f(\tilde{y}_{t+1}|\tilde{\mathbf{y}}_t^{(k)}, \tilde{\mathbf{x}}_t^{(l)})}{f(\tilde{y}_{t+h}|\tilde{\mathbf{y}}_t^{(k)})} \mathrm{d}\mathbf{w}, \end{aligned} \tag{4.1}$$

where $f(\cdot)$ indicates the probability density function. Integers k and l indicate the orders of $\tilde{y}_t$ and $\tilde{x}_t$, respectively. Symbols $\tilde{\mathbf{y}}_t^{(k)} := [\tilde{y}_t, \tilde{y}_{t-1}, \cdots, \tilde{y}_{t-k+1)}]$ and $\tilde{\mathbf{x}}_t^{(l)} := [\tilde{x}_t, \tilde{x}_{t-1}, \cdots, \tilde{x}_{t-l+1}]$ denote the embedding vectors of past states of $\tilde{Y}$ and $\tilde{X}$, respectively. The integral variable is $\mathbf{w} = [\tilde{y}_{t+1}, \tilde{\mathbf{y}}_t^{(k)}, \tilde{\mathbf{x}}_t^{(l)}]$. Integers k and l indicate the orders of $\tilde{y}_t$ and $\tilde{x}_t$, respectively. To calculate the TE between process variables, a critical step is to estimate the probability density functions [8, 25], which are computationally burdensome. The computational complexity of process data based TE is $O((n + 2\sigma)^{k+l+1})$, where n is the number of amplitude bins for estimating the probability density function, and σ is an estimated integer of the discrete kernel width.

In contrast to this, alarm signals are binary-valued variables, taking the values of 0 or 1. Entropy approaches to binary data have been exploited in the communication and neuroscience areas. To solve multi-user communication problems, Wyner & Ziv [94] proposed theorems on the convexity and boundary of entropies of binary random variables. In the neuroscience areas, Gourévitch & Eggermont [33] and Ito et al. [48] exploited TE to evaluate information transfer between auditory cortex neurons, which involve signals with a spiking trains or a consecutive bursts of binary data. However, the application of TE to alarm data has rarely been studied. Owing to physical connectivities, abnormalities often propagate from one alarm signal to another. If the causal relations of these alarm signals can be quantified, they would be helpful in revealing the propagation of abnormalities.

As discussed in Section 2.2.1, alarm signals usually have two forms: The first form only takes the alarm occurrences into account, i.e., an alarm signal takes the value 1 only at the time instants when the normal specification of its monitored variable is violated; The second form is that an alarm signal takes the value 1 throughout the time period when the monitored process signal stays in the abnormal state. For the causality inference based on alarm data, the overlapping of '1's between two alarm signals in the second form (2.2) is not related to the mutual interactions between alarm signals, and thus may

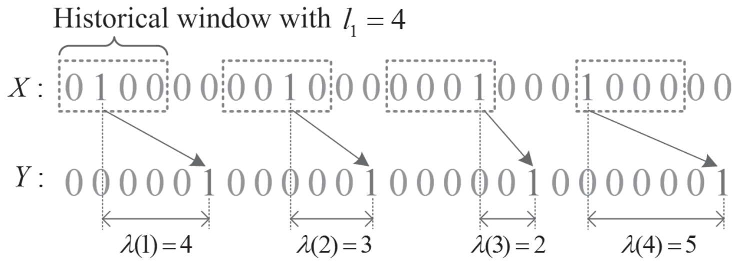


Figure 4.1: A numerical example for predicting Y based on the history of X under the influence of random occurrence delays.

lead to erroneous conclusions. Thus, the alarm occurrence signal in (2.1) is used for the purpose of causality inference. In the sequel, the alarm signal is referred to as the alarm occurrence signal if not otherwise stated.

There are two characteristics of alarm signals to be considered in the usage of discrete transfer entropies to detect causality between alarm signals. First, the random occurrence delays between alarm signals make the calculation of TE complicated. Given two correlated alarm signals X and Y, the time interval that an alarm occurrence in Y follows its counterpart in X may be changing due to disturbances. Such a time interval is referred to as the random occurrence delay [41]. A numerical example of random occurrence delays between X and Y is shown in Fig. 4.1. There are four alarm occurrences in X. For each occurrence in X, there is a corresponding alarm occurrence in Y with some occurrence delay λ. Four random occurrence delays are $\lambda(1) = 4$, $\lambda(2) = 3$, $\lambda(3) = 2$, and $\lambda(4) = 5$. If the alarm occurrences in X are defined as 1's in the dashed windows in Fig. 4.1, they are helpful in the prediction of alarm occurrences in Y, i.e. $[0, 1, 0, 0] \to 1$, $[0, 0, 1, 0] \to 1$, $[0, 0, 0, 1] \to 1$, $[1, 0, 0, 0] \to 1$. The causality from X to Y is obvious because whenever X occurs, Y is found to occur after X. Thus, the random occurrence delays lead us to only care about the alarm occurrence in X *within a historical window*, instead of at each time instant. The second characteristic is that alarm occurrences in Y are mutually independent, so that there is no need to consider the effect of the past states of Y. In view of these two characteristics, a modified TE is proposed as

$$\begin{aligned} T_{X\to Y} &= \sum p(y_{t+1}, \overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)}) \\ &\quad \cdot \log_2 \frac{p(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)})}{p(y_{t+1}|\overset{\circ}{y}_t^{(k_1)})}. \end{aligned} \tag{4.2}$$

where k_1 and l_1 indicate the orders of the effect variable and cause variable respectively, and d_1 denotes the time lag between X and Y. The summation symbol $\sum$ denotes all possible combinations of 3 dimensional states of y_{t+1}, $\overset{\circ}{y}_t^{(k_1)}$, and $\overset{\circ}{x}_{t-d_1}^{(l_1)}$, among which $\overset{\circ}{x}_{t-d_1}^{(l_1)}$ and $\overset{\circ}{y}_t^{(k_1)}$ are calculated as

$$\overset{\circ}{x}_{t-d_1}^{(l_1)} = \begin{cases} \bigcup x_{(t-d_1):(t-d_1-l_1+1)}, & \text{if } y_{t+1} = 1, \\ x_{t-d_1}, & \text{otherwise}, \end{cases} \tag{4.3}$$

$$\overset{\circ}{y}_t^{(k_1)} = \begin{cases} \bigcup y_{t:(t-k_1+1)}, & \text{if } y_{t+1} = 1, \\ y_t, & \text{otherwise}. \end{cases} \tag{4.4}$$

Here x_t and y_t take values from $\{0, 1\}$, and $\bigcup x_{t_1:t_2} = x_{t_1} \vee x_{t_1-1} \vee \cdots \vee x_{t_2}, t_1 > t_2$, and the symbol $\vee$ stands for the "or" operation of Boolean algebra. Accordingly, the higher-order TE in (4.1) is turned to be a first-order TE in (4.2). The computational complexity is decreased to $O(2^3)$, which is much lower than that of the higher-order TE in (4.1).

Remark 1: The definition of TE in (4.2) is different from the existing methods in [33], [48] and [101] in two aspects. i) The modified TE is specific for alarm occurrence signals in (2.1) so as to tolerate random occurrence delays, whereas the influence of random occurrence delays was not considered in [101] that exploited the alarm signals in (2.2). ii) The historical states within a time window are used, while the conventional discrete transfer entropy in [33], [48] and [101] requires the calculation based on the historical state at every time instant. Compared with the conventional TE, the modified TE in (4.2) is more suitable for alarm occurrence signals with the consideration of the above-mentioned two characteristics of alarm signals.

Essentially, the TE from X to Y is the amount of uncertainty in predicting the future of Y reduced by given the history of X. Thus, analogously to the TE in [33] and [55], the TE from X to Y in (4.2) can be expressed in another form as

$$T_{X\to Y} = H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}) - H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)}), \tag{4.5}$$

where $H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)})$ and $H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)})$ are conditional entropies formulated as

$$H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}) = -\sum p(y_{t+1}, \overset{\circ}{y}_t^{(k_1)}) \cdot \log_2 p(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}),$$

$$H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)}) = -\sum p(y_{t+1}, \overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)}) \cdot \log_2 p(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)}).$$

The above conditional entropies are positive and hold the relationship as $H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}) \leq H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)})$. Accordingly, a normalized transfer entropy (NTE) is formulated as

$$\begin{aligned} NT_{X\to Y} &= \frac{T_{X\to Y}}{H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)})} \\ &= \frac{H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}) - H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)})}{H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)})} \\ &= 1 - \frac{H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)}, \overset{\circ}{x}_{t-d_1}^{(l_1)})}{H(y_{t+1}|\overset{\circ}{y}_t^{(k_1)})}. \end{aligned} \tag{4.6}$$

The NTE $NT_{X\to Y}$ is in the range of $[0, 1]$. The NTE closer to 1 indicates stronger causality, while a NTE closer to 0 denotes weaker causality.

Remark 2: The definition of the NTE in (4.6) is different from the existing methods in [25, 33] in two aspects. i) The NTE in (4.6) works for discrete transfer entropies, whereas the counterpart in [25] only works for continuous transfer entropies. ii) The calculation of NTE in (4.6) is based on conditional entropies while the counterpart in [33] gave a biased estimate of NTE by incorporating the mean value of TEs from surrogates.

In (4.6), there are three parameters, namely, d_1, k_1 and l_1, which are nonnegative integers. The time lag d_1 can be determined as the one associated with the maximum value of $NT_{X\to Y}(d_1)$'s for all nonnegative values of $d_1 \leq D$, where D is a user-defined maximum value of time lags; see the step 3 in Section 4.3.3 and Figs. 4.10 and 4.13 in Section 4.4 for the trends of NTEs versus time lags as illustration. Since the alarm occurrences, namely, '1's of x_t in (2.1), are almost mutually independent, the order of the effect variable can be set as $k_1 = 0$. The parameter l_1 indicates the number of historical states of X to be included for the prediction of Y. To determine l_1, the principle is to make l_1 tolerate the majority of random occurrence delays λ. Assume the time stamps of all alarm occurrences in X (or Y) as $S^x = [s_1^x, \cdots, s_{M_x}^x]^T$ (or $S^y = [s_1^y, \cdots, s_{M_y}^y]^T$), where M_x (or M_y) indicates the total number of alarm occurrences in X (Y). Then, the time intervals between the alarm occurrences in Y and the counterparts in X are

$$\Gamma = [\gamma]_{M_y\times M_x} = S^y \cdot \mathbf{I}_{1\times M_x} - \mathbf{I}_{M_y\times 1} \cdot (S^x)^T, \tag{4.7}$$

where **I** is an all-one vector. The minimum positive element in each row is found as

$$\lambda_i = \min\left\{\gamma_{i,j} : \gamma_{i,j} > 0, j = 1, 2, \cdots, M_x\right\}. \tag{4.8}$$

where $i = 1, 2, \cdots, M_y$. The majority of random occurrence delays can be found from 1.5×IQR (interquartile range) while the values beyond the range are outliers [66]. So as to tolerate random occurrence delays, the order of the cause variable can be set as $l_1 = 1.5 \times \mathrm{IQR}_\lambda$, where IQR_λ indicates the difference between the third quartile and the first quartile of $\lambda_i, i = 1, 2, \cdots, M_y$. Further, l_1 should not be set too large so as to avoid the all-one vector of $\overset{\circ}{x}_t^{(l_1)}$ in (4.3). An upper bound of l_1 is given as 20, which is the maximum time interval of non-chattering alarms according to ISA standard [47]. As a result, the parameter l_1 is finally determined as

$$l_1 = \min\{1.5 \times \mathrm{IQR}_\lambda, 20\}. \tag{4.9}$$

4.2.2 Detection of Direct Causality

To detect whether the causality between two continuous-valued signals is along a direct pathway or through some intermediate variables, the concept of direct transfer entropy (DTE) was proposed in [25]. More specifically, given three continuous-valued process signals X, Y, Z with the transfer entropies $T_{X\to Y}$, $T_{X\to Z}$, and $T_{Z\to Y}$ significantly large, we need to know whether the causality from X to Y is direct or through Z. By calculating the DTE from X to Y based on Z, the direct information flow paths can be inferred.

Analogous to the DTE of process signals in [25], the DTE of alarm signals is defined as follows. Based on (4.6) and given the third alarm signal Z, the NTE from X to Z is calculated as

$$NT_{X\to Z} = 1 - \frac{H(z_{t+1}|\overset{\circ}{z}_t^{(m_1)}, \overset{\circ}{x}_{t-d_2}^{(l_2)})}{H(z_{t+1}|\overset{\circ}{z}_t^{(m_1)})} \tag{4.10}$$

where $\overset{\circ}{z}_t^{(m_1)} = z_t \vee z_{t-1} \vee ... \vee z_{t-m_1+1}$, and d_2 indicates the time lag between X and Z. The NTE from Z to Y is

$$NT_{Z\to Y} = 1 - \frac{H(y_{t+1}|\overset{\circ}{y}_t^{(k_2)}, \overset{\circ}{z}_{t-d_3}^{(m_2)})}{H(y_{t+1}|\overset{\circ}{y}_t^{(k_2)})} \tag{4.11}$$

where d_3 indicates the time lag between Z and Y. As a result, the DTE from X to Y based on Z is formulated as

$$D_{X\rightarrow Y|Z} \quad = \quad \sum p(y_{t+1}, \overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)}, \overset{\circ}{x}_{t-d_1}^{(l_1)}) \cdot \log_2 \frac{p(y_{t+1}|\overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)}, \overset{\circ}{x}_{t-d_1}^{(l_1)})}{p(y_{t+1}|\overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)})},$$

where $k' = k_1 = k_2$. Analogous to (4.6), the normalized direct transfer entropy (NDTE) from X to Z based on Y in (4.12) is formulated as

$$\begin{aligned} ND_{X\rightarrow Y|Z} \quad &= \quad \frac{D_{X\rightarrow Y|Z}}{H(y_{t+1}|\overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)})} \\ &= \quad 1 - \frac{H(y_{t+1}|\overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)}, \overset{\circ}{x}_{t-d_1}^{(l_1)})}{H(y_{t+1}|\overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)})}, \end{aligned} \tag{4.12}$$

where

$$H(y_{t+1}|\overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)}) \quad = \quad -\sum p(y_{t+1}, \overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)}) \cdot \log_2 p(y_{t+1}|\overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)}),$$

$$\begin{aligned} H(y_{t+1}|\overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)}, \overset{\circ}{x}_{t-d_1}^{(l_1)}) = -\sum p(y_{t+1}, \overset{\circ}{y}_t^{(k')}, \\ \overset{\circ}{z}_{t-d_3}^{(m_2)}, \overset{\circ}{x}_{t-d_1}^{(l_1)}) \cdot \log_2 p(y_{t+1}|\overset{\circ}{y}_t^{(k')}, \overset{\circ}{z}_{t-d_3}^{(m_2)}, \overset{\circ}{x}_{t-d_1}^{(l_1)}), \end{aligned} \tag{4.13}$$

The range of $ND_{X\rightarrow Y|Z}$ is $[0, 1]$. An NDTE closer to 1 indicates a stronger direct causality, while an NDTE closer to 0 denotes a weaker direct causality.

4.3 Causality Inference

This section proposes a statistical test to determine the significance levels for NTE and NDTE, investigates the minimum requirement of alarm occurrences, and presents the steps of the proposed causality inference method.

4.3.1 Significance Test

To determine whether the causality and direct causality between alarm signals are significant, a significance test is a necessary step. In [8], a significance threshold for the transfer entropy of continuous-valued variables was established using Monte Carlo simulations with surrogate data. The main idea is as follows. To detect the causality $T_{\tilde{X}\rightarrow\tilde{Y}}$ from process variable $\tilde{X}$ and $\tilde{Y}$, the null hypothesis is that $T_{\tilde{X}\to\tilde{Y}}$ is not significant, implying there is no causality from $\tilde{X}$ to $\tilde{Y}$. By implementing N-trial Monte Carlo simulations, the transfer entropies of surrogate data are calculated as $T_{\tilde{X}^s\to\tilde{Y}^s}(i), i = 1, 2, \cdots, N$. The mean value and standard deviation of $T_{\tilde{X}^s\to\tilde{Y}^s}(i)$ are μ_T^s and σ_T^s, respectively. A six-sigma threshold is determined as

$$\gamma_T = \mu_T^s + 6\sigma_T^s. \tag{4.14}$$

If the observed TE $T_{\tilde{X}\to\tilde{Y}} > \gamma_T$, the null hypothesis is rejected and $\tilde{X}$ is determined to be the cause variable of $\tilde{Y}$; otherwise, there is no causality from $\tilde{X}$ to $\tilde{Y}$.

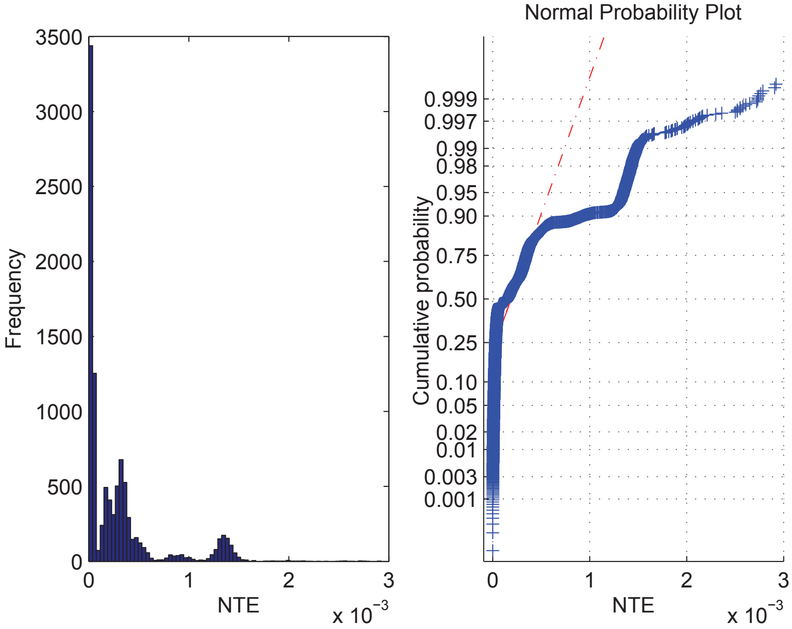


Figure 4.2: Distribution of NTEs calculated from 10,000 pairs of surrogate alarm signals X^s and Y^s: (a) the histogram is skewed right and multimodal; (b) the normal probability plot deviates from the red straight line.

As indicated in [8], this significance test method requires that the TEs calculated from the surrogate data follow the Gaussian distribution. However, the TEs or DTEs of surrogate alarm data are not necessarily Gaussian distributed. Thus, the significance method in [8] is not appropriate to determine significance thresholds for alarm signals, which is illustrated numerically as follows. Given original alarm signals X, Y, and Z, the surrogate alarm signals X^s, Y^s, and Z^s should have the same length and contain the same number of 1's as the original data. For instance, assuming the sequence length of X as L and the number of 1's as M, X_s is generated with Bernoulli distribution $\mathcal{B}(1, \frac{M}{L})$.

By implementing N-trial Monte Carlo simulations, the NTEs and NDTEs for all surrogate alarm data are calculated as $NT_{X^s \to Y^s}(i)$ and $ND_{X^s \to Y^s|Z^s}(i)$, where $i = 1, 2, \cdots, N$. Fig. 4.2 shows an example of 10,000 pairs of uncorrelated random signals X^s and Y^s with sequence length $43,200$ and 345 alarm occurrences. It is obvious that NTEs calculated from the 10,000 pairs of alarm signals do not follow the Gaussian distribution. Thus, the method in [8] cannot be applied to determine a reliable significance threshold.

With the unknown distribution of NTEs from surrogate alarm data, the significance test based on a Monte Carlo p-value presented in [81], [32], [39] and [76] is more appropriate. The Monte Carlo p-value is formulated as [39, 76]

$$p = \frac{b+1}{N+1}, \tag{4.15}$$

where b indicates the number of NTEs (or NDTEs) from the surrogates larger than the NTE (or NDTE) from the observed data. If $p < \alpha$, the null hypothesis is rejected at the α significance level. Assuming $p = \alpha$, the number of trials of Monte Carlo simulations needed for the significance test is [32, 81]

$$N = \frac{b+1}{\alpha} - 1. \tag{4.16}$$

Here we choose $b = 1$, i.e., only one sample with $NT^s_{X^s \to Y^s}(i) \geq NT_{X \to Y}$ (or $ND^s_{X^s \to Y^s|Z^s}(i) \geq ND_{X \to Y|Z}$) is allowed for the rejection of null hypothesis. Given $\alpha = 0.01$, the number of Monte Carlo simulations is $N = 199$. Thus, the significance threshold γ_T (or γ_D) for TE (or DTE) is found as the 2nd maximum value $NT^s_{X^s \to Y^s}(i)$ (or $ND^s_{X^s \to Y^s|Z^s}(i)$). To make the significance test more robust, the above procedures are repeated for K times. The significance thresholds γ_T and γ_D for NTE and NDTE are calculated as

$$\gamma_T = \frac{1}{K} \Sigma_{k=1}^{K} \gamma_T(k), \tag{4.17}$$

$$\gamma_D = \frac{1}{K} \Sigma_{k=1}^{K} \gamma_D(k), \tag{4.18}$$

where $\gamma_T(k)$ (or $\gamma_D(k)$) is the 2nd maximum value in the k-th Monte Carlo test. Here, $K = 15$ is chosen as a rule of thumb, determined via Monte Carlo simulations with different values of L (the sequence length) and M (the number of alarm occurrences). An example of the significance thresholds of NTEs versus K is given as Fig. 4.3. Uncorrelated alarm signals X_s and Y_s are generated with $L = 10,000$ and the alarm occurrence following $\mathcal{B}(1, 0.015)$. Fig. 4.3-(a) presents the significance thresholds of 1000 trials of simulations while Fig. 4.3-(b) gives the trend of variance of γ_T versus K. It can be found that γ_T changes drastically around the mean value for $K < 15$ while the variation is not significant for $K \geq 15$.

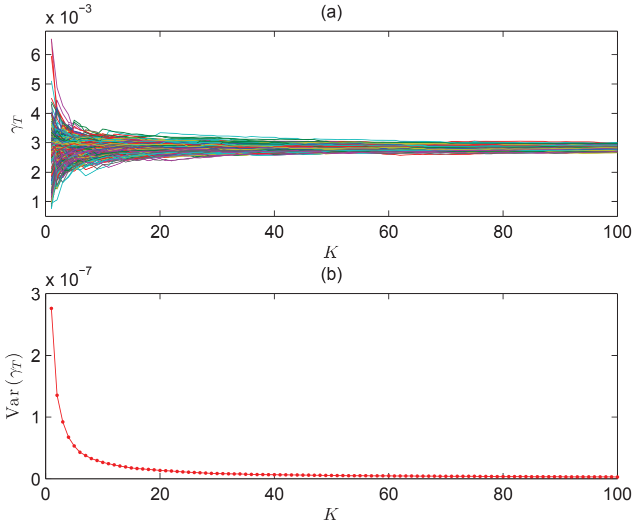


Figure 4.3: Significance thresholds γ_T of NTEs versus K in 1000 Monte Carlo simulations.

Based on (4.17), the null hypothesis is rejected if $NT_{X \to Y} > \gamma_T$. Otherwise, there is no causality from X to Y. Similarly, based on (4.18), the null hypothesis is rejected if $ND_{X \to Y|Z} > \gamma_D$, so that the direct causality from X to Y based on Z can be determined. To test the effectiveness of the proposed method, a numerical example is given next.

Example 1. Two alarm signals X and Y are created as two independent random binary sequences that follow the Bernoulli distribution $\mathcal{B}(1, p)$, where p is the probability of alarm occurrence. As a result, X and Y have no causal relationship. Then, Monte Carlo simulations are implemented. The sequence length is set as $L = 10000$ while p differs in every simulation and $p \sim \mathcal{U}(0.005, 0.05)$, where $\mathcal{U}$ is a uniform distribution. The NTE $NT_{X \to Y}$ is calculated based on (4.6). The corresponding significance threshold γ_T is obtained using the proposed significance test method in (4.17). 1000 pairs of $NT_{X\to Y}$ and γ_T are obtained as shown in Fig. 4.4-(a). Comparing 1000 pairs of $NT_{X\to Y}$ and γ_T, only four pairs are found with $NT_{X\to Y} > \gamma_T$ as indicated in Fig. 4.4-(b). The detection accuracy is 99.6%. Similar results are also observed for NDTEs (omitted here due to space limitation). Thus, it can be concluded that the proposed method is quite effective in determining the significance threshold.

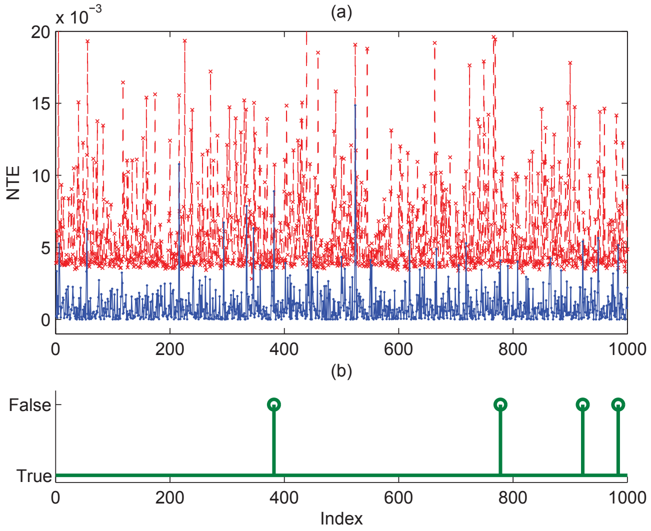


Figure 4.4: Results of 1000 Monte Carlo simulations for the case without causal relation: (a) NTEs (blue line) and corresponding significance thresholds (red line); (b) correctness of results ("True" for correct estimate while "False" for incorrect estimate).

4.3.2 Number of Alarm Occurrences

In the estimation of transfer entropies, a natural question is: What is the minimum number of alarm occurrences required to achieve reliable detection? Hu et al. [41] investigated a similar question for the detection of correlated alarms, where a rule of thumb recommended is that there should be as least 27 alarm occurrences. Analogous to [41], this study investigates the minimum requirement of alarm occurrences for the detection of causality between alarm signals, based on Monte Carlo simulations.

Example 2. A pair of alarm signals X and Y with causal relation from X to Y are created as follows: Given the sequence length L and the probability of alarm occurrences p, a binary sequence that follows the Bernoulli distribution

$\mathcal{B}(1, p)$ is generated and denoted as X. The alarm occurrences in Y are assumed to follow their counterparts in X with some occurrence delays. Assume the time stamp of the ith alarm occurrence in X as s_i^x, then the time stamp of the consequential alarm occurrence in Y is

$$s_i^y = s_i^x + \lambda_i \tag{4.19}$$

where the integer λ_i is the ith occurrence delay and $\lambda_i \sim \mathcal{N}(v_m, v_s), v_m \in \mathbb{N}^+, v_s \in \mathbb{R}_{\geq 0}$. Fig. 4.5 shows an example of correlated alarm signals with $L = 20,000$, $p = 0.005$, $v_m = 10$, and $v_s = 1$. The binary sequences of X and Y are shown in Fig. 4.5-(a) and (c), respectively. It can be found from Fig. 4.5-(b) and (d) that the occurrence delay between the alarm occurrences in two sequences is changing around the mean value 10.

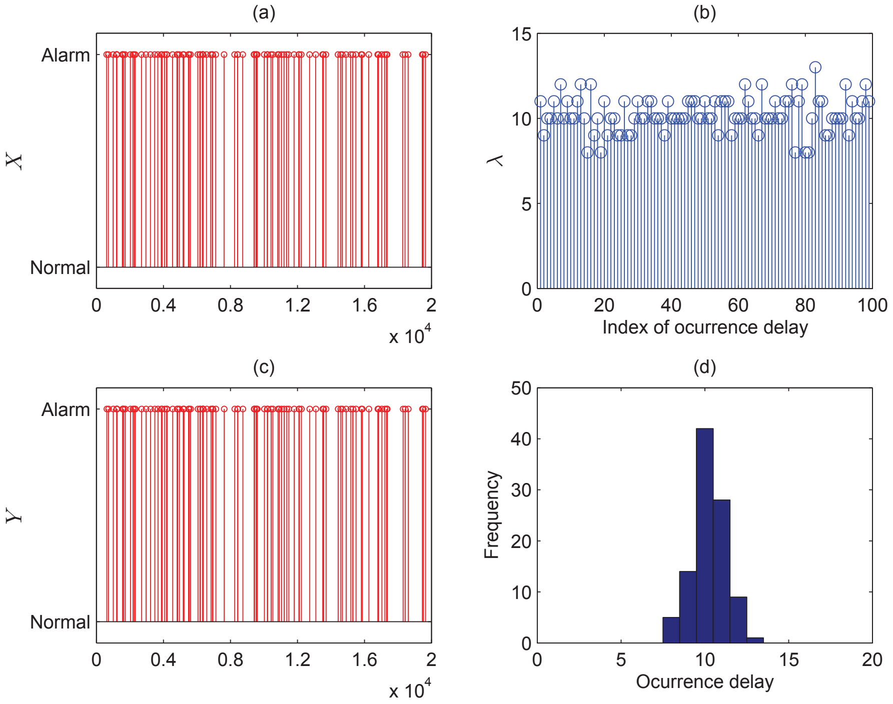


Figure 4.5: An example of correlated alarms and the distribution of random occurrence delays: a) the alarm signal X, c) the alarm signal Y, b) the occurrence delays, and d) the histogram of occurrence delays.

Setting the parameters as $L = 2,000,000$, $v_m = 10$, $v_s = 0.5$, and $p \in \{p : p = 0.005j, j = 1, 2, ..., 10\}$, alarm signals X and Y containing M_w alarm occurrences are created. Then, subsequences X_s and Y_s containing N alarm occurrences are constructed based on X and Y. The segmentation is done as follows: Y_s starts from the ith alarm occurrence and ends at the $(i+N-1)$th alarm occurrence of Y, $i = 1, 2, ..., M_w - N + 1, M_w \gg N$. X_s is extracted from X over the same time duration as Y_s in Y. As a result, $M_w - N + 1$ pairs of subsequences X_s and Y_s are constructed. The number of alarm occurrences N is set as $N \in \{N : N = 5 + 5j, j = 1, 2, ..., 39\}$. $M_w - N + 1$ trials of simulations are performed to calculate the NTEs $NT^{(i)}_{X_s \to Y_s}$. The mean value of NTEs is computed as $\overline{NT}_{X_s \to Y_s} = \Sigma_{i=1}^{M_w - N+1} NT^{(i)}_{X_s \to Y_s}$. For $p \in \{p : p = 0.005j, j = 1, 2, ..., 10\}$, the trends of $\overline{NT}_{X_s \to Y_s}$ versus N are shown in Fig. 4.6-(a). The differences $\Delta\overline{NT}(N) = \overline{NT}_{X_s \to Y_s}(N) - \overline{NT}_{X_s \to Y_s}(N-5)$ versus $N = 15, 20, ..., 200$ are shown in Fig. 4.6-(b). It can be found that $\overline{NT}_{X_s \to Y_s}$ rises strikingly with the increase of N for $N \leq 50$. For $N > 50$, the change of $\overline{NT}_{X_s \to Y_s}$ is not significant. Accordingly, the minimum number of alarm occurrences $N = 50$ is chosen as a rule of thumb.

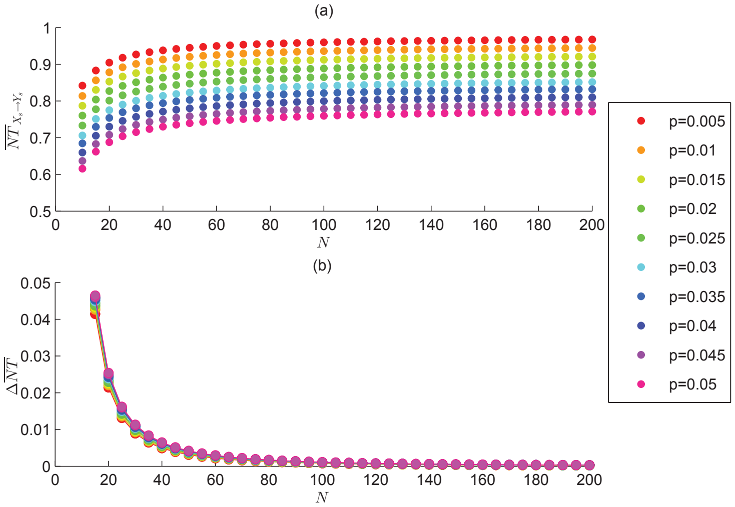


Figure 4.6: Average NTEs v.s. numbers of alarm occurrences for different p.

4.3.3 Causality Inference Based on NTEs

To implement the proposed method, the following conditions are assumed: 1) alarm occurrences of each alarm tag are mutually independent; 2) at least 50 alarm occurrences should be present in each alarm sequence; 3) the orders k_1 and l_1 in (4.2) and (4.3) should be predetermined. The first condition is generally satisfied for the following reason. Each occurred alarm should be associated with an abnormality, so that industrial plant operators have to pay attention or to take action in a prompt manner; otherwise, abnormal situations associated with occurred alarms would have negative effects on operation safety and/or efficiency. Since the occurrence of abnormality is generally independent from its previous or next occurrence, we can reasonably assume the alarm occurrences to be mutually independent. The second condition is for a reliable detection of causality, as observed in Section 3.2. For the third condition, the recommended values of k_1 and l_1 have been given at the end of Section 2.2.

The sequence of steps for the causality inference based on NTEs consists of the following steps, also depicted as a flowchart in Fig. 4.7:

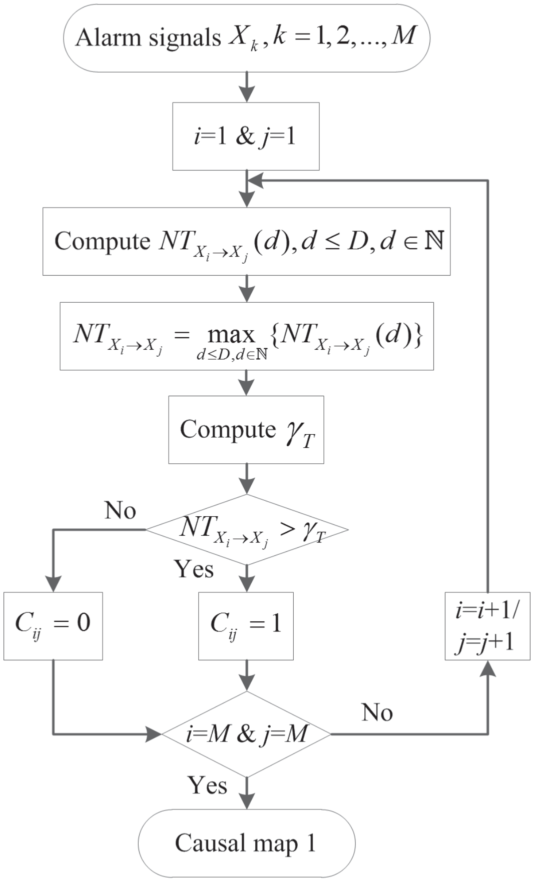

Alarm signals $X_k, k = 1, 2, ..., M$
i=1 & j=1
Compute $NT_{X_i \to X_j}(d), d \leq D, d \in \mathbb{N}$
$NT_{X_i \to X_j} = \max_{d \leq D, d \in \mathbb{N}} \{NT_{X_i \to X_j}(d)\}$
Compute γ_T
No
$NT_{X_i \to X_j} > \gamma_T$
Yes
$C_{ij} = 0$
$C_{ij} = 1$
i=i+1/
j=j+1
No
i=M & j=M
Yes
Causal map 1

Figure 4.7: Flowchart for the causality inference based on NTEs.

1. A group of alarm signals $X_k, k = 1, 2, .., M$, are prepared for the causality inference.

2. The iterative computation of TEs for pairwise alarm signals X_i and X_j starts from $i = 1$ and $j = 1$.

3. $NT_{X_i \to X_j}(d), i \neq j$ is calculated based on (4.6) for $\forall d \leq D, d \in \mathbb{N}$. The NTE from X_i to X_j is calculated as $NT_{X_i \to X_j} = \max_d NT_{X_i \to X_j}(d)$ and the corresponding time lag is found as $d_{ij} = \arg\max_d NT_{X_i \to X_j}(d)$.

4. The significance threshold γ_T in (4.17) is calculated using the modified Monte Carlo test in Section III-A.

5. Compare $NT_{X_i \to X_j}$ with γ_T. If $NT_{X_i \to X_j} > \gamma_T$, $C_{ij} = 1$; otherwise, $C_{ij} = 0$.

6. Check if NTEs for all pairwise alarm signals have been calculated or not. If it is, the causality inference proceeds to the next step; otherwise, it goes back to Step 3 for another pair of alarm signals by increasing i or j.

7. Plot a causal map indicating information flow paths based on the matrix C. If $C_{ij} = 1$, an arrowed line is drawn to denote causality from X_i to X_j; otherwise, there is no connection between X_i and X_j.

4.3.4 Detection of Direct Causality Based on NDTEs

Based on the causal map in Section 4.3.3, the computation proceeds to detect the direct causality. The complete detection procedure based on NDTEs is composed of the following steps, also depicted as a flowchart in Fig. 4.8:

1. A group of alarm signals $X_k, k = 1, 2, .., M$, are prepared as described in Section 4.3.3.

2. The iterative computation of NDTEs from X_i to X_j based on X_l starts from $i = 1$ and $j = 1$.

3. If $C_{ij} = 1$, the computation proceeds to the next step; otherwise, it proceeds to Step 9.

4. The computation starts from the intermediate variable X_l with $l = 1$.

5. If $C_{il} = 1$ & $C_{lj} = 1$, the computation proceeds to the next step; otherwise, it proceeds to Step 8.

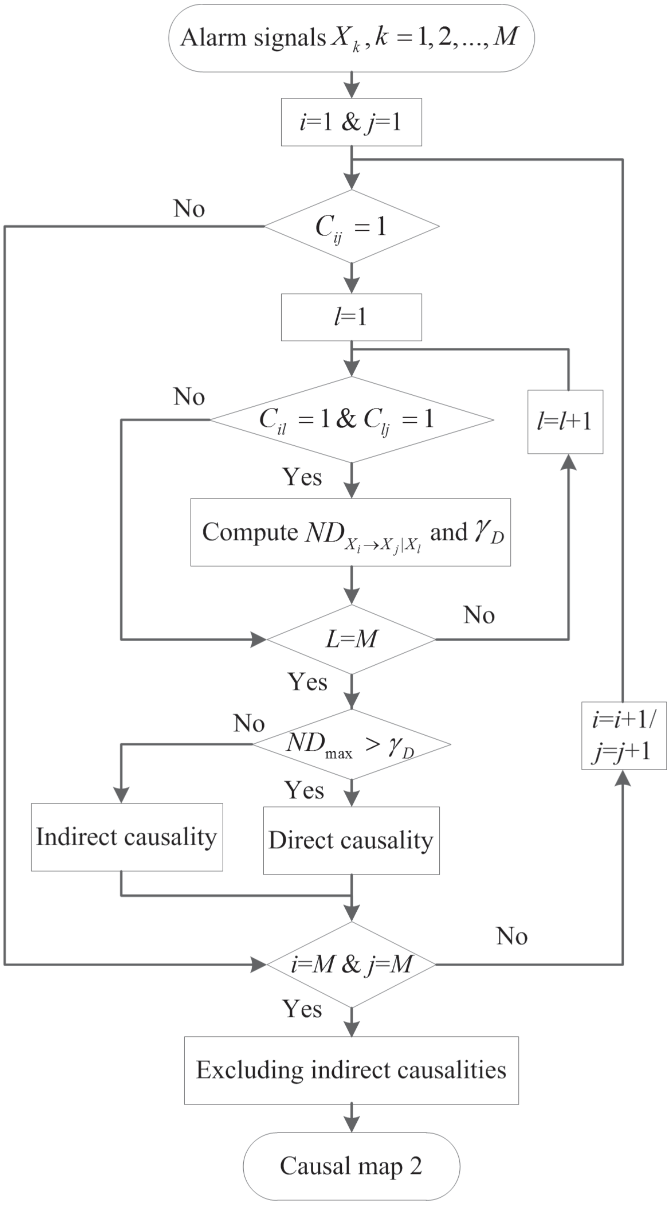


Figure 4.8: Flowchart for the detection of direct causalities based on the NDTEs.

6. $ND_{X_i \to X_j | X_l}$ is calculated based on (4.12). The time delays d_{ij} and d_{lj} are the ones corresponding to the maximum value of $NT_{X_i \to X_j}(d)$ and $NT_{X_l \to X_j}(d)$ in Section 4.3.3. The significance threshold γ_D in (4.18) is calculated using the method in Section III-A.

7. Check if NDTEs from X_i to X_j through all intermediate variables are calculated or not. If it is, the computation proceeds to the next step;

otherwise, it goes back to Step 5 for another intermediate alarm signal by increasing l.

8. Determine the maximum NDTE $ND_{\max}$ from X_i to X_j based on all intermediate alarm signals. Compare $ND_{\max}$ with its corresponding γ_D. If $ND_{\max} > \gamma_D$, we say that the causality from X_i to X_j is direct; otherwise, it is indirect.

9. Check if NDTEs for all pairwise alarm signals have been calculated or not. If yes, the computation proceeds to the next step; otherwise, go back to Step 3 for another pair of alarm signals by increasing i or j.

10. The information flow path from X_i to X_j that are determined to be indirect should be excluded from the causal map in Section 4.3.3.

11. Plot a causal map indicating the direct information flow paths.

The proposed method is unsupervised; no labelled training data sets are required in the above steps. This is also shown by the flowcharts in Figs. 4.7 and 4.8.

4.4 Case Studies

This section demonstrates the effectiveness of the proposed causality inference methods for alarm signals using both the simulated and industrial case studies.

4.4.1 Case I: Numerical Case Study

This subsection illustrates the effectiveness of the proposed method in causality inference by simulating a group of correlated alarm signals. The method in Example 2 is used to create three alarm signals with causal relations as $X_1 \to X_2 \to X_3$. First, X_1 is produced as a binary sequence that follows $\mathcal{B}(1, p)$. Assume the time stamp of the ith alarm occurrence in X_1 as s_i^1, then the time stamps of the consequential alarm occurrences in X_2 and X_3 are

$$\begin{cases} s_i^2 = s_i^1 + \lambda_i^{(1)} \\ s_i^3 = s_i^2 + \lambda_i^{(2)} \end{cases} . \tag{4.20}$$

Here $\lambda_i^{(1)}$ and $\lambda_i^{(2)}$ are independent Gaussian random variables, each of which takes the mean value 20 and standard deviation 1. Setting the parameters as $L = 20,000$ and $p = 0.01$, the simulated alarm occurrence signals X_1, X_2, and X_3 are generated as shown in Fig. 4.9. The number of alarm occurrences in each sequence is 102. According to Section 4.3.2, the alarm occurrences are sufficient for the causality inference.

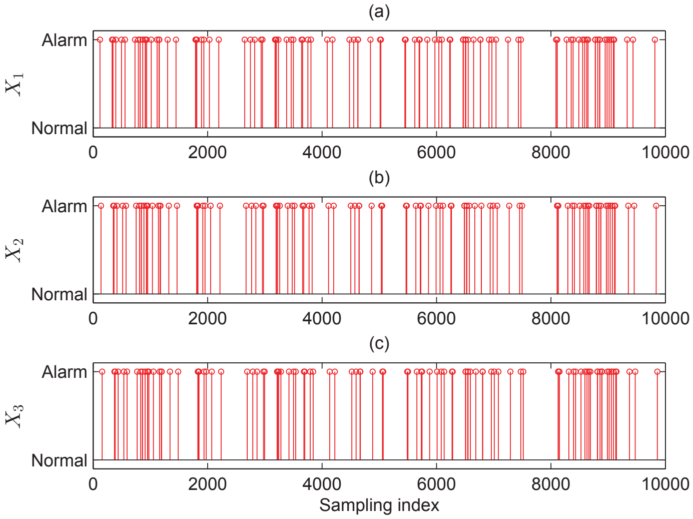


Figure 4.9: Alarm occurrence signals: (a) the alarm signal X_1; (b) the alarm signal X_2; (c) the alarm signal X_3.

The proposed method is applied to alarm occurrence signals in Fig. 4.9. First, the causal relations are detected by following the proposed procedures in Section 4.3.3. The calculated NTEs versus time lags for each pair of alarm signals are shown in Fig. 4.10. The normalized transfer entropies and their corresponding significance thresholds (in the brackets) are calculated as shown in Table 4.1. As a result, the causalities among the three variables are concluded via a causal map in Fig. 4.11-(a).

Table 4.1: Normalized transfer entropies (From row to column).

$T_{X_i \to X_j}$	X_1	X_2	X_3
X_1		0.73(0.022)	0.62(0.023)
X_2	0.109(0.112)		0.78(0.025)
X_3	0.110(0.111)	0.110(0.111)	

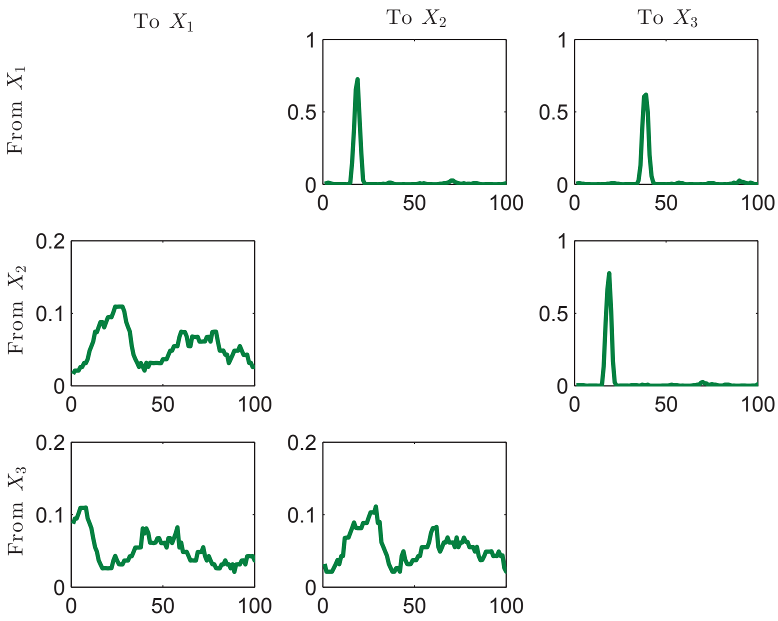


Figure 4.10: NTEs versus time lags.

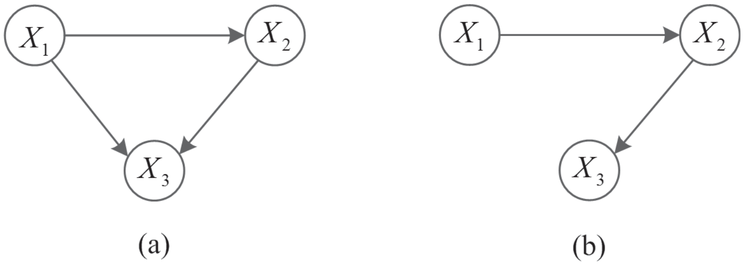


Figure 4.11: Causal maps: (a) the causal map of all information flow paths; (b) the causal map of direct information flow paths

Furthermore, to detect whether the causality from X_1 to X_3 is direct or through the intermediate variable X_2, the NDTE is calculated to be $ND_{X_1 \to X_3|X_2} = 0.0037$. The threshold from Monte Carlo test is obtained to be $\gamma_D = 0.025$. Since $ND_{X_1 \to X_3|X_2} < \gamma_D$, the causality from X_1 to X_3 is indirect. Then, the information flow path from X_1 to X_3 should be excluded and a new causal map of direct causality is drawn as Fig. 4.11-(b). According to the detection results, X_1 is the cause of X_2, X_2 is the cause of X_3, and X_1 is the cause of X_3, among which the causality from X_1 to X_3 is indirect. The conclusion is consistent with the given relationship.

4.4.2 Case II: Industrial Case Study

To validate the effectiveness of the proposed method, an industrial example is provided. The facility is an oil plant. Totally, more than five thousand alarms are configured to monitor process variable variations, operation and safety conditions. In a time period of 3.5 days, there were 288 alarm tags being annunciated. The proposed causality inference method was applied to all of these alarm tags. Owing to space limitation, we select six alarm tags as representatives to demonstrate the effectiveness of the proposed method. The first five alarm tags from an oil plant are located in the same interconnected unit, while the 6th alarm tag is located in a disconnected unit. The first five alarm tags have the following physical connections: Tag 2 represents the total gas flow in a burner which has two supplies from cell A and cell B. Tag 1 indicates the gas flow from the feed cell B. Tags 3, 4, and 5 are associated with 3 products as the outputs from the burner. Once Tag 1 turns abnormal, Tag 2 will become abnormal, too, followed by subsequent abnormalities in Tags 3, 4 and 5. Based on this, the information flow paths follow the order as: Tag 1 → Tag 2 → Tag 3 → Tag 4 → Tag 5. The 6th alarm tag is located in a disconnected unit.

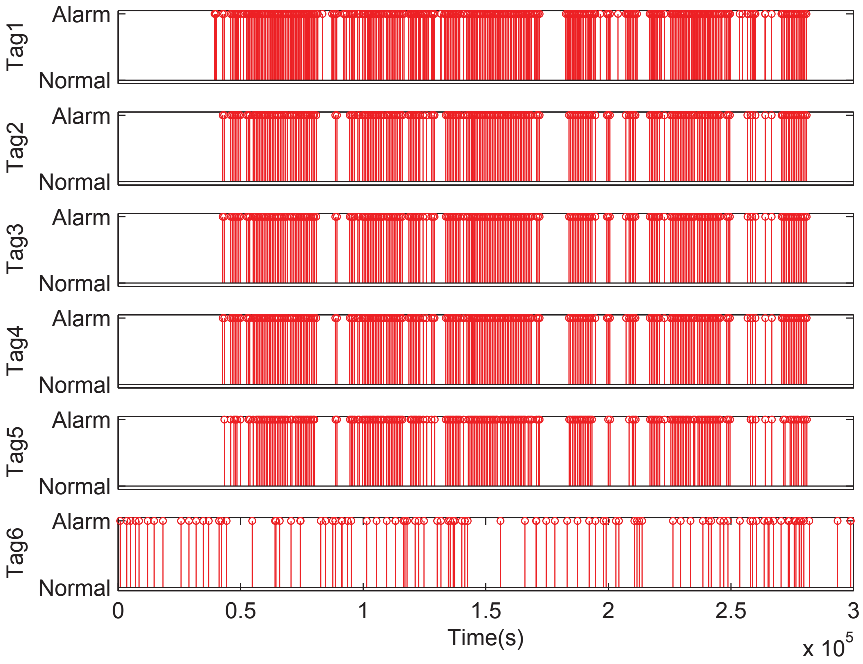


Figure 4.12: Alarm occurrence signals over 3.5 days time period.

The historical data samples of the six alarm tags are shown in Fig. 4.12. Over the 3.5-day time period, the number of alarm occurrences of the six alarm signals were 653, 242, 242, 242, 219, and 106, respectively. Calculating the chattering indices as proposed in [90] and [58], Tag 1 was found to be chattering. Using a 30-second off-delay timer, the number of alarm occurrences is reduced to 282. According to Section 4.3.2, the number of alarm occurrences in this case is sufficient to reach a reliable detection of causality.

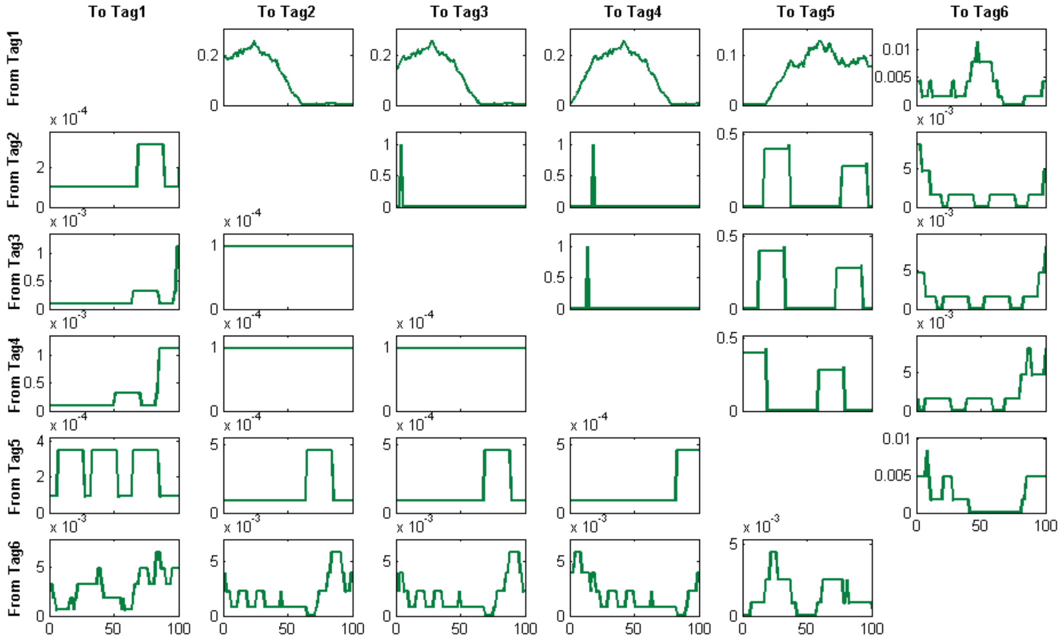


Figure 4.13: Trends of transfer entropies versus time lags.

Then, the proposed method is applied to calculate the NTEs among alarm signals in Fig. 4.12. The NTEs under different time delays are calculated as shown in Fig. 4.13. The maximum values of NTEs are selected. Meanwhile, significance thresholds are calculated from the Monte Carlo test in Section 4.3.1. Taking the causality inference from Tag 1 to Tag 2 as an example, the maximum value of NTE is $NT(\text{Tag1} \rightarrow \text{Tag2}) = 0.2577$ and the significance threshold is $\gamma_T = 0.0155$. Since $NT(\text{Tag1} \rightarrow \text{Tag2}) > \gamma_T$, we say that there is causality from Tag1 to Tag2. As a result, a causal map describing the information flow paths is drawn in Fig. 4.14.

The NTEs are presented in Table 4.2. It is found that the causalities from Tag 2 to Tag 3, from Tag 2 to Tag 4, and from Tag 3 to Tag 4 are very strong while the causalities from Tag 1 to other alarm tags are relatively small. The

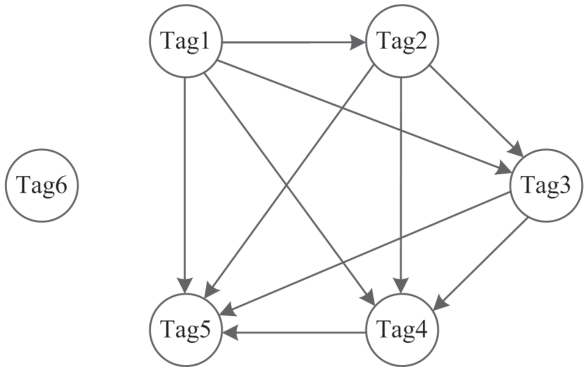


Figure 4.14: Causal map of information flow paths.

result is reasonable because Tag 1 is associated with one of the two supplies for the process related to Tag 2, making the causality is not as strong as those among Tags 3, 4, and 5.

Table 4.2: Normalized transfer entropies (From row to column).

	Tag1	Tag2	Tag3	Tag4	Tag5	Tag6
Tag1		0.2577 (0.0155)	0.2577 (0.0153)	0.2577 (0.0155)	0.1293 (0.0151)	0.0115 (0.0215)
Tag2	0.0003 (0.0127)		1 (0.0011)	1 (0.0013)	0.4336 (0.0142)	0.0082 (0.0188)
Tag3	0.0011 (0.0134)	0.0001 (0.0135)		1 (0.0011)	0.4336 (0.0144)	0.0082 (0.0183)
Tag4	0.0011 (0.0130)	0.0001 (0.0139)	0.0001 (0.0140)		0.4336 (0.0135)	0.0082 (0.0204)
Tag5	0.0004 (0.0122)	0.0004 (0.0131)	0.0005 (0.0131)	0.0005 (0.0133)		0.0085 (0.0167)
Tag6	0.0065 (0.0090)	0.0058 (0.0089)	0.0058 (0.0087)	0.0058 (0.0089)	0.0045 (0.0102)	

Furthermore, the NDTE is calculated for each pair of alarm tags with NTEs larger than their corresponding thresholds. Taking Tag 1 and Tag 5 as an example, the possible intermediate variables include Tags 2, 3, and 4. Thereby, the NDTEs from Tag 1 to Tag 5 based on Tags 2, 3, and 4 are calculated in Table 4.3. It can be seen that no direct causality is found between the cause variables and effect variables in Table 4.3. By excluding the indirect causalities from the causal map in Fig. 4.14, a causal map describing all direct information flow paths is drawn in Fig. 4.15. Tag 1 is identified as the root cause of this group of alarm signals. The conclusion is consistent with the physical connections given at the beginning of this subsection and was validated by the operating engineers.

Table 4.3: Detection of direct causalities based on NDTEs.

Cause variable	Effect variable	Intermediate variables	NDTE	Threshold
Tag1	Tag3	Tag2	0	0.0129
Tag1	Tag4	Tag2, Tag3	0	0.0122
Tag1	Tag5	Tag2, Tag3, Tag4	0.00014	0.0118
Tag2	Tag4	Tag3	0	0.0005
Tag2	Tag5	Tag3, Tag4	0	0.0149
Tag3	Tag5	Tag4	0	0.0152

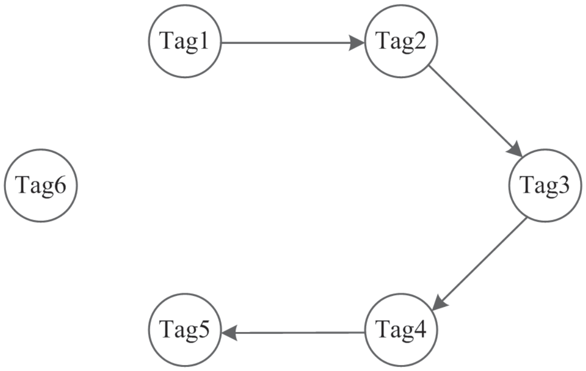


Figure 4.15: Causal map of direct information flow paths.

4.5 Summary

To detect causes and effects of a group of alarm signals, find a possible root cause, and track the propagation of an abnormality, a causality inference method based on the modified transfer entropies is proposed. The modified TE and the modified DTE, as well as their normalized counterparts NTE and NDTE, are defined for binary-valued alarm signals, considering their characteristics in terms of the random occurrence delays and the mutual independence of alarm occurrences. Considering that NTEs or NDTEs calculated from surrogate alarm signals do not follow Gaussian distributions, a statistical test is proposed to determine the significant thresholds for NTEs and NDTEs. By following the sequence of steps of the proposed causality inference method outlined in Sections III-C and III-D, information flow paths can be found. The effectiveness of the proposed method is illustrated via numerical and industrial case studies to detect causal interactions from alarm signals.

Chapter 5

Discovering Association Rules of Mode-Dependent Alarms from Alarm and Event Logs*

5.1 Overview

State-based or condition-based alarming has emerged as a prevalent method to reduce nuisance alarms and inhibit alarm floods in the alarm management of process industries. Such a strategy minimizes the number of active alarms by modifying the alarm attributes or suppression status based on certain conditions. However, the configuration of state-based alarms in practice relies on process knowledge, making it time and resource intensive. In order to identify associations between alarms and states in an automated way, this paper proposes a data-driven method to detect mode-dependent alarms from Alarm & Event (A&E) logs, where the messages of alarms and operating modes are stored. To the best knowledge of the authors, this is the first reported exploratory study to detect mode-dependent alarms using a data driven method. The major contributions of the proposed methodologies are as follows: 1) the determination of association rules of mode-dependent alarms is formulated as a hypothesis testing problem; 2) a modified *a-priori algorithm* is developed to find frequent patterns of operating modes; and 3) the proposed procedure is evaluated on an industrial A&E data set. Using the proposed method, mode-dependent alarms for either single operating mode or multiple operat-

*A version of this chapter has been submitted for publication as: Hu, W., Chen, T., & Shah, S. L. (2016). Discovering association rules of mode-dependent alarms from alarm and event logs. *IEEE Trans. Control Systems Technology.*

ing modes are discovered. The results can be used to help users in identifying the root causes of some nuisance alarms or configuring state-based alarming modules.

5.2 Preliminaries on Mode Dependent Alarms

This section introduces state-based alarming in practice, and then illustrates the switching of operating modes and the triggering of consequential alarms.

5.2.1 State-Based Alarming in Practice

According to ISA-18.2 [47], state-based alarming is an important logic-based technique that modifies alarm attributes or suppresses alarms based on the state changes of equipment or processes. In process industries, most alarms work normally only for one single-state [38]. Once the operating state changes, some alarms may no longer indicate true abnormalities and often are triggered and therefore become nuisance alarms. By using a state-based alarming strategy, these alarms will not be presented to operators under certain conditions. As a result, the operators are less overloaded and have more capability to process critical alarms. An industrial case of a simple lube oil pump, where the state-based alarm should be configured, is presented in Example 1.

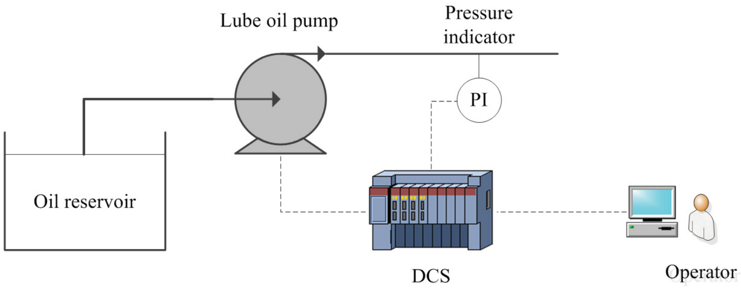


Figure 5.1: Sketch of a lube oil pump (PI is an indicator and the pump is controlled by an operator via a DCS (Distributed Control System)).

Example 1. A lube oil pump is used to supply lube oil for large rotary and alternating machinery. It is opened or stopped by operators with commands from the control room. An indicator is configured to monitor the discharge pressure as shown in Figure 5.1. Whenever the discharge pressure drops below the low alarm limit, a low pressure alarm $x^{(e1)}$ will be annunciated. When the pump was manually stopped by an operator, the discharge pressure decreased and triggered an alarm. But this alarm was a consequence of stopping the pump and it did not indicate any abnormality. Hence, no action was required and the alarm became useless. Figure 5.3 shows the trends of state transitions of $x^{(e1)}$ and mode changes of the pump. It can be seen that the occurrence (ALM) of $x^{(e1)}$ always followed after the pump was stopped and the return-to-normal instant (RTN) of $x^{(e1)}$ always appeared after the start of the pump. In this case, the state-based alarming solution is effective in removing the consequential nuisance alarm, i.e., $x^{(e1)}$ should be suppressed immediately when the operator stops the pump and enabled when the pump is started.

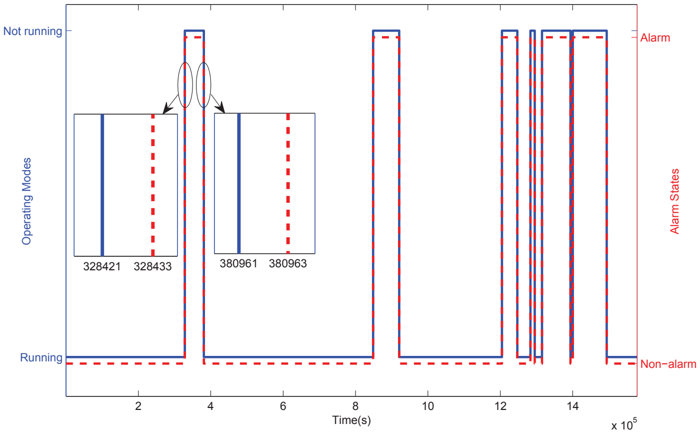


Figure 5.2: Switching between “Running” and “Closure” modes of a lube oil pump (blue line) and the transition of alarm states (dashed red line). Two boxed plots show zoomed details when the pump started or stopped, and the corresponding alarm states.

5.2.2 Operating Modes and Consequential Alarms

The stated-based alarming strategies are usually configured based on predefined states, e.g., the running or closure of equipment, the startup or shutdown of a plant, the feed of different fuel or raw material, the variation of operating rates, and the change for making different products [38, 51]. However, not all such state changes are explicit. Some states, such as the last three, are often difficult to identify and require algorithms to detect them from process data [63]. As for the startup or shutdown of a plant, the annunciator is often full of nuisance alarms. Thus, good process knowledge is required for the configuration of state-based alarms for such cases. This study only focuses on the states of equipment rather than a large section of processes or the whole plant. These states are also known as operating modes of equipment [10] and usually determined by operators. The changes of such operating modes are explicitly known and easily recorded by data acquisition systems. So no additional algorithms or process knowledge is required to identify the modes.

Table 5.1: Operating modes of some commonly used equipment.

Equipment	Operating modes
Pump	Start / Stop / Standby
Valve	Open / Close
Motor	Start / Stop / Standby / Forward / Reverse
Controller	Auto / Manual / Cascade

Some commonly used equipment or modules, such as pumps, valves, motors, controllers, and their operating modes, are listed in Table 5.1. Depending on processes and manipulation guidelines, the equipment could be operated either at on-off modes, such as the start and stop of a pump, or at multiple modes, such as the use of an additional "standby" mode for a pump. Generally, the alarm may work normally at one specific mode and become a consequential nuisance alarm when the operating mode switches. Thus, the state-based alarming should be used to suppress nuisance alarms in such cases. To configure state-based alarms for a specific operating mode, it should be known which alarms are the consequential alarms. In practice, this mainly relies on process knowledge of engineers or operating experience of field operators. However, such procedure analysis work is usually time and resource intensive, and the configurations could be incomplete. Alternatively, a data driven method suggested in this study can automate the detection of mode-dependent alarms, and thus significantly reduce the work in associating the alarms and operating modes. Generally, the consequential alarms are found to be annunciated by following the switching to a specific mode and not necessarily returned to normal after another mode, due to the influence of multiple operating states or the change of normal specifications. To detect association rules of mode-based alarms, only the alarm occurrences are included in the computation while the return-to-normal instants may lead to erroneous detection results.

5.3 Problem Formulation

This section formulates the problem of the detection of mode dependent alarms. Some preliminary definitions used throughout this chapter are presented first.

5.3.1 Mathematical Definitions

Given an A&E log containing the events of alarms and operating modes, some preliminary concepts are defined as follows.

Some basic mathematical notations are: Given a set $V = \{v_i : i = 1, 2, \cdots, |V|\}$, $|V|$ denotes the cardinality of V; $V \cup v'$ indicates that an additional element v' is added to V, and $V \backslash v'$ indicates that an element v' is excluded from V; given a column vector $\mathbf{v} = [v_1, v_2, \cdots, v_{|\mathbf{v}|}]^T$, $|\mathbf{v}|$ denotes the length of $\mathbf{v}$.

Definition 2. An operating mode variable, denoted as x_i, is a binary-valued variable indicating the change of operating modes. $\mathcal{X} = \{x_i : i = 1, 2, \cdots, |\mathcal{X}|\}$ is denoted as a set of distinct mode variables.

Definition 3. An alarm occurrence variable, denoted as y_j, is a binary-valued variable indicating the occurrence of an alarm. $\mathcal{Y} = \{y_j : j = 1, 2, \cdots, |\mathcal{Y}|\}$ is denoted as a set of distinct alarm variables.

Fig. 5.3 gives an example of an operating mode variable (a) and an alarm occurrence variable (b). By taking x_i as the open of a valve (and y_j as the occurrence of a low flow alarm), $x_i(t) = 1$ ($y_j(t) = 1$) indicates that the valve was opened (the low alarm was annunciated) at time instant t. It can seen from Fig. 5.3 that the valve was opened five times and the low alarm was annunciated five times too. For simplicity, in the rest of the paper, the "operating mode variable" and "alarm occurrence variable" are notated as "mode" and "alarm", respectively.

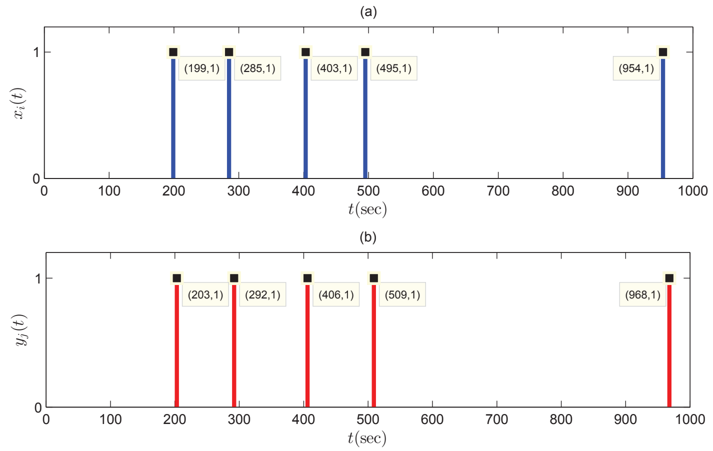


Figure 5.3: An example of an operating mode variable (a) and an alarm occurrence variable (b).

Definition 4. An event (v, t_k) indicates the occurrence of v at a specific time instant t_k. A log is a collection of events for all $v \in \mathcal{V}$, denoted as $\{(v, t_k) : v \in \mathcal{V}, k \in \mathbb{N}^+\}$, where $\mathbb{N}^+$ indicates natural numbers. Thus, the A&E log is denoted as $\mathbb{D} = \{(v, t_k) : v \in \mathcal{X} \cup \mathcal{Y}, k \in \mathbb{N}^+\}$. The sub-databases $\mathbb{D}(\mathcal{X}) = \{(x_i, t_k) : x_i \in \mathcal{X}, k \in \mathbb{N}^+\}$ and $\mathbb{D}(\mathcal{Y}) = \{(y_j, t_k) : y_j \in \mathcal{Y}, k \in \mathbb{N}^+\}$ denote databases of all mode events and all alarm events, respectively. It is obvious that $\mathbb{D} = \mathbb{D}(\mathcal{X}) \cup \mathbb{D}(\mathcal{Y})$.

Definition 5. Given all events $\{(v, t_k) : k \in \mathbb{N}^+\}$ for a specific v, a time vector of v is formulated as $\mathbf{t}^{(v)} = [t_1^{(v)}, t_2^{(v)}, \cdots, t_{|\mathbf{t}^{(v)}|}^{(v)}]^T$, where $t_{k_1}^{(v)} < t_{k_2}^{(v)}$ if $k_1 < k_2$. Then, $\mathbf{t}^{(x_i)}$ and $\mathbf{t}^{(y_j)}$ denote the time vectors for $x_i \in \mathcal{X}$ and $y_j \in \mathcal{Y}$, respectively.

Definition 6. The k_1-th occurrence of $x_i \in \mathcal{X}$ is said to be followed by the k_2-th occurrence of $y_j \in \mathcal{Y}$, if $\Delta t = t_{k_2}^{(y_j)} - t_{k_1}^{(x_i)} \in [0, W_{th}]$, where W_{th} is a user-defined time window.

Definition 7. The frequency $\xi(v)$ indicates how many times an alarm or a mode $v \in \mathcal{X} \cup \mathcal{Y}$ occurs in a given database $\mathbb{D}$. It is obvious that $\xi(v) = |\mathbf{t}^{(v)}|$. The frequency $\xi(x_i \rightarrow y_j)$ indicates how many times $x_i \in \mathcal{X}$ is followed by $y_j \in \mathcal{Y}$ in $\mathbb{D}$.

5.3.2 Problem Formulation

The objective of the study is to discover all association rules $\psi : \mathcal{X} \to \mathcal{Y}$ of mode-dependent alarms from a given A&E log $\mathbb{D}$ on $\mathcal{X}$ and $\mathcal{Y}$. The feasibility is based on the principle that the switching to a specific operating mode is always followed by the occurrence of an alarm. Accordingly, the mode-dependent alarm is defined as follows:

Definition 8. The alarm $y_j \in \mathcal{Y}$ is dependent on the operating mode $x_i \in \mathcal{X}$, if
(i) $\xi(x_i \to y_j) \geq F_{th}$, and
(ii) $p(y_j|x_i) = 1$.
Then, y_j is said to be a mode-dependent alarm of x_i, denoted using an association rule as $x_i \Rightarrow y_j$. Otherwise, y_j is not a mode-dependent alarm of x_i and this is denoted as $x_i \not\Rightarrow y_j$.

The first condition makes sure that the association rule of the mode-dependent alarm is determined from a non-sparse data set. The second condition guarantees the cause-effect relation between x_i and y_j, which relies on the conditional probability $p(y_j|x_i)$, an estimation of which is given by

$$\hat{p}(y_j|x_i) = \frac{\xi(x_i \to y_j)}{\xi(x_i)}. \tag{5.1}$$

Note that the minimum frequency F_{th} in condition (i) is a user-defined threshold to ensure that the association rules are determined from a commensurate data set. In general, we may have $x_i \Rightarrow y_j$ even when there is a relatively small specification on F_{th}. However, this will lead to a likely large and many false association rules. Thus, the setting of F_{th} helps to prune a large amount of infrequent association rules.

A graphical example is shown in Fig. 5.4. The vertical color bars denote the occurrences of events for the operating mode “PUMP01:STOP” (x_1), and three alarms “PI01.LOW:ALM” (y_1), “FI01.LOW:ALM” (y_2), and “FI05.FAIL:ALM” (y_3). It can be found that $\hat{p}(y_1|x_1) = 1$, $\hat{p}(y_2|x_1) = 0.8$, $\hat{p}(y_3|x_1) = 0$, which are calculated based on a time window W_{th} in Definition 6. However, these estimated probabilities do not necessary imply that $p(y_j|x_i) = 1$ in Definition 8. To conclude which alarm is a mode-dependent alarm of x_1, we first need to compare their frequencies with the user-defined

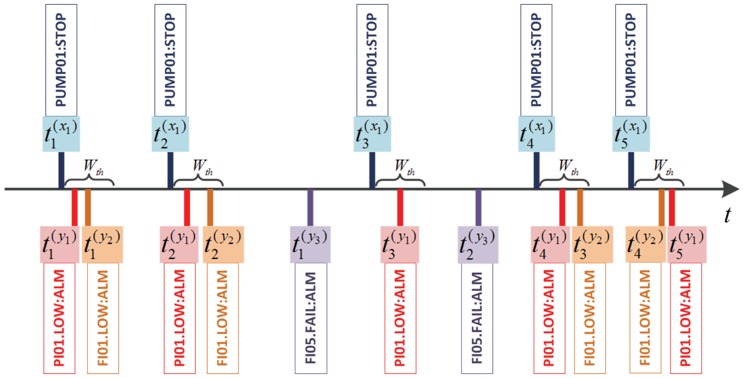


Figure 5.4: Example of the mode-dependent alarm on a time axis.

threshold F_{th}. Furthermore, a hypothesis is required to check the population parameter, namely, $p(y_j|x_i)$. For each pair of $x_i \in \mathcal{X}$ and $y_j \in \mathcal{Y}$, the association is determined via a hypothesis test as

$$\begin{cases} H_0: & p(y_j|x_i) = 1, \\ H_A: & p(y_j|x_i) \neq 1. \end{cases} \tag{5.2}$$

If the null hypothesis H_0 is rejected, we accept the alternative hypothesis and conclude that $x_i \nRightarrow y_j$; otherwise, we have $x_i \Rightarrow y_j$. Eventually, the problem of detecting association rules $\psi : \mathcal{X} \to \mathcal{Y}$ from an A&E log $\mathbb{D}$ can be resolved by implementing the hypothesis test for each pair of $x_i \in \mathcal{X}$ and $y_j \in \mathcal{Y}$.

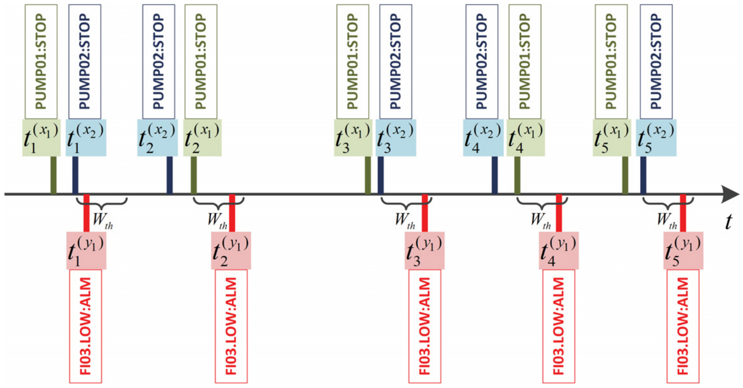


Figure 5.5: Example of mode-dependent alarms of multiple modes on a time axis.

In addition to the mode-dependent alarm for a single operating mode in Fig. 5.4, it is not uncommon to see that one alarm may occur after the co-occurrence of multiple operating modes. A graphical example with two modes “PUMP01:STOP” (x_1) and “PUMP02:STOP” (x_2), and one alarm

"FI01.LOW:ALM" (y_1) is shown in Fig. 5.5. It can be seen that whenever both pumps stopped running, the low flow rate alarm y_1 was annunciated after a short time period. In the case of multiple operating modes, a critical step is the detection of co-occurrences of these modes. If one mode occurs in a short period after the occurrence of another mode, we may regard them related; otherwise, their occurrences are not related. Accordingly, the co-occurrence of multiple operating modes are defined as follows:

Definition 9. The n modes $X^{(n)} = \{x_i \in \mathcal{X} : i = 1, 2, \cdots, n\}$ are said to co-occur within a time window W_{th}, if $\exists S = \{t_l^{(x_i)} : i = 1, 2, \cdots, n; l \in \mathbb{N}^+\}$ such that $\Delta S = \max S - \min S \leq W_{th}$. The frequencies $\xi(X^{(n)})$ and $\xi(X^{(n)} \to y_j)$ indicate the number of co-occurrences of $x_1, x_2, \cdots, x_n$ and the number of times that their co-occurrences are followed by $y_j \in \mathcal{Y}$ in $\mathbb{D}$, respectively.

Analogous to Definition 8, the alarm $y_j \in \mathcal{Y}$ is said to be a mode-dependent alarm of $X^{(n)} = \{x_i \in \mathcal{X} : i = 1, 2, \cdots, n\}$, notated as $X^{(n)} \Rightarrow y_j$, if (i) $\xi(X^{(n)} \to y_j) \geq F_{th}$, and (ii) $p(y_j|X^{(n)}) = 1$; otherwise, $X^{(n)} \not\Rightarrow y_j$. The hypothesis test is the same as that in eqn. (5.2) by checking the population parameter $p(y_j|X^{(n)})$. As a result, the set of detected association rules does not just include mode-dependent alarms of a single operating mode, but also contains those of multiple operating modes.

5.4 Methodology

This section discusses how to identify mode-dependent alarms from historical A&E Logs. Algorithms to detect the frequent patterns of operating modes and the association rules of mode-dependent alarms are proposed.

5.4.1 Frequent Patterns of Operating Modes

According to the previous section, the determination of association rules involves two cases, namely, the mode-dependent alarms for a single mode and multiple modes. Thus, prior to detecting $\psi : \mathcal{X} \to \mathcal{Y}$, it is required to find the frequent patterns of operating modes, which is defined as follows.

Definition 10. A frequent n-pattern $\tilde{X}^{(n)}$ of operating modes is an itemset of n modes $X^{(n)} = \{x_i \in \mathcal{X} : i = 1, 2, \cdots, n\}$, the co-occurrence frequency of which satisfies $\xi(X^{(n)}) \geq F_{th}$. The collection of all n-patterns $\tilde{X}^{(n)}$ is notated as L_n. The collection of all frequent patterns is notated as $\mathcal{L} = \bigcup_{n=1}^{N} L_n$.

Given a sub-database $\mathbb{D}(\mathcal{X})$, the objective is to detect $\mathcal{L}$ that contains all frequent mode patterns. A prevalent method for frequent pattern mining is the *a-priori* algorithm developed by Agrawal & Srikant [2]. It detects frequent itemsets from transactional database in a join-and-prune manner. In the joining step, all possible candidate itemsets of k items are generated as C_k. In the pruning step, all infrequent itemsets are excluded from C_k by examining their frequencies in the database. In this study, a modified *a-priori* method is proposed for the detection of frequent mode patterns. The framework of the algorithm remains the same as that of the *a-priori* algorithm. The difference is the calculation of frequencies, which serves as a critical step for the determination of frequent patterns.

Algorithm 3 Discovery of frequent patterns of operating modes.

```
Input Arguments: $\mathbb{D}(\mathcal{X}), \mathcal{X}, W_{th}, F_{th}$.
Output Argument: $\mathcal{L}$.
Extract all $\mathbf{t}^{(x_i)}, \forall x_i \in \mathcal{X}$
$C_1 = \mathcal{X}$;
$L_1 = \{x \in C_1 : \xi(x) \geq F_{th}\}$;
$k = 2$;
while $L_{k-1} \neq \emptyset$ do
    $C_k =$ Fun-candidate$(L_{k-1})$;
    for each candidate $c \in C_k$ do
        $\mathbf{T} = \{\mathbf{t}^{(x_i)} : \forall x_i \in c\}$
        $\xi(c) =$ Fun-frequency$(\mathbf{T}, W_{th})$;
    end for
    $L_k = \{c \in C_k | \xi(c) \geq F_{th}\}$;
    $k = k + 1$;
end while
$\mathcal{L} = L_1 \cup L_2 \cup \cdots \cup L_{k-1}$.
```

Algorithm 1 is the main algorithm for the detection of frequent patterns of operating modes. Initially, the collection of 1-item candidates is taken as the set of all distinct modes, namely, $C_1 = \mathcal{X}$. Then, the frequent 1-patterns are found as $L_1 = \{X^{(1)} \in C_1 : \xi(X^{(1)}) \geq F_{th}\}$. Iteratively, the frequent k-patterns L_k are found from k-mode candidates C_k. In each iteration, the candidates C_k are generated based on L_{k-1} using Algorithm 2.

For any two $k-1$ frequent patterns $l_1 = \{x_1, \cdots, x_{k-2}, x_{k-1}\} \in L_{k-1}, l_2 = \{x_1, \cdots, x_{k-2}, \tilde{x}_{k-1}\} \in L_{k-1}, x_{k-1} \neq \tilde{x}_{k-1}$, a k-item candidate is generated as $c = X^{(k)} = \{x_1, \cdots, x_{k-2}, x_{k-1}, \tilde{x}_{k-1}\}$. However, if there is any $k-1$ subset $s = X^{(k-1)} \subset c$ not frequent, namely, $s \notin L_{k-1}$, c will not be included in C_k. To determine whether the k-item candidate is frequent or not, the frequency of c is calculated using Algorithm 3.

Algorithm 4 Fun-candidate (L_{k-1}): Generate candidates.

```
Input Argument: $L_{k-1}$.
Output Argument: $C_k$.
for $\forall l_1 = \{x_1, \cdots, x_{k-2}, x_{k-1}\} \in L_{k-1}, \forall l_2 = \{x_1, \cdots, x_{k-2}, \tilde{x}_{k-1}\} \in L_{k-1}, x_{k-1} \neq \tilde{x}_{k-1}$ do
    $c = \{x_1, \cdots, x_{k-2}, x_{k-1}, \tilde{x}_{k-1}\}$;
    $C_k = C_k \cup c$;
    for $\forall$ (($k-1$)-subset $s \subset c$) do
        if $s \notin L_{k-1}$ then
            $C_k = C_k \backslash c$;
        end if
    end for
end for
```

According to Definition 9, the frequency of a k-item candidate c is counted as the number of times all the operating modes $X^{(k)} = \{x_i \in c : i = 1, 2, \cdots, k\}$ co-occur in the database $\mathbb{D}(\mathcal{X})$. This requires the time vector $\mathbf{t}^{(x_i)}$ of each mode $x_i \in c$ and the time window W_{th}. Firstly, all the time vectors are stacked into one vector as

$$\tilde{\mathbf{t}} = [\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_{|\tilde{\mathbf{t}}|}], \tilde{t}_l \in \bigcup_{i=1}^{k} \mathbf{t}^{(x_i)}, \tag{5.3}$$

where the elements are sorted in an ascending order, namely, $\tilde{t}_l < \tilde{t}_{l+1}$; and, $|\tilde{\mathbf{t}}| = \sum_{i=1}^{k} |\mathbf{t}^{(x_i)}|$. Then, an index vector indicating that $\tilde{t}_l$ belongs to the set $\mathbf{t}^{(x_i)}$ for $x_i \in c$ is created as

$$\mathbf{h} = [h_1, h_2, \ldots, h_{|\mathbf{h}|}], \tag{5.4}$$

where $|\mathbf{h}| = |\tilde{\mathbf{t}}|$; the l-th element h_l is assigned to be i, if the corresponding $\tilde{t}_l$ in $\tilde{\mathbf{t}}$ comes from the i-th time vector $\mathbf{t}^{(x_i)}$. Afterwards, we use a time window of size W_{th} to scan from the first element $\tilde{t}_1$ to the last one $\tilde{t}_{|\tilde{\mathbf{t}}|}$ of $\tilde{\mathbf{t}}$. The frequency of c is initialized as $\xi(c) = 0$. If the events for all operating modes

Algorithm 5 Fun-frequency $(\mathbf{T}, W_{th})$: Calculate the frequency of a pattern with multiple modes.

```
Input Argument: $\mathbf{t}^{(x_1)}, \mathbf{t}^{(x_2)}, \cdots, \mathbf{t}^{(x_k)}, W_{th}$.
Output Argument: $\xi$.
$\tilde{\mathbf{t}} = [\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_{|\tilde{\mathbf{t}}|}] = \mathbf{t}^{(x_1)} \cup \mathbf{t}^{(x_2)} \cup \cdots \cup \mathbf{t}^{(x_k)}$;
$\mathbf{h} = [h_1, h_2, \ldots, h_{|\tilde{\mathbf{t}}|}]$, where $h_l = i$, if $\tilde{t}_l \in \mathbf{t}^{(x_i)}, i = 1, 2, \cdots, k$;
$[\tilde{\mathbf{t}}, la]$ = sort($\tilde{\mathbf{t}}$,'ascend');
$\mathbf{h} = \mathbf{h}(la)$;
$d = 1$;
$\xi = 0$;
for $i = 1$ to $|\tilde{\mathbf{t}}|$ with step $d$ do
    $lb = (\tilde{\mathbf{t}} \geq \tilde{t}_l) \wedge (\tilde{\mathbf{t}} < \tilde{t}_l + W_{th})$;
    $b$ = unique($\mathbf{h}(lb)$)
    if length($b$)== $N$ then
        $\xi = \xi + 1$;
        $d$ = sum($lb$);
    else
        $d = 1$;
    end if
end for
```

$x_i \in c, i = 1, 2, \cdots, k$, are found within the same time window, the frequency will be added 1, namely, $\xi(c) = \xi(c) + 1$, as shown in Algorithm 5. Then, the calculation will move to the next time window starting from the mode event next to the previous time window. Finally, the frequency $\xi(c)$ of c is obtained when the scan ends at the last event. The collection of all frequent k-patterns is found as $L_k = \{c \in C_k | \xi(c) \geq F_{th}\}$.

Remark 1. The method in Agrawal & Srikant [2] cannot be directly applied to the problem in this section, owing to the difference in database structures. It requires transactional database which is a database consisting of many transactions. Each transaction is a sequence of records. Thus, the frequencies of items can be easily counted as their occurrences in the transactions. In contrast, the database $\mathbb{D}(\mathcal{X})$ of operating modes is essentially a whole sequence of records. Thus, to adapt the *a-priori* algorithm to such database, the major problem is the calculation of frequencies of items, which is solved using Algorithm 5.

5.4.2 Mode-Dependent Alarms for Single Operating Mod-e

This subsection elaborates how to detect mode-dependent alarms for a single operating mode from the A&E log $\mathbb{D}$ on $\mathcal{X} \cup \mathcal{Y}$. The detection is implemented for each pair of $x_i \in L_1$ and $y_j \in \mathcal{Y}$. An association rule of mode-dependent alarm must satisfy two conditions in Definition 8. Accordingly, the detection includes two steps: First, the frequency $\xi(x_i \to y_j)$ should be calculated and compared with F_{th}; second, the hypothesis test in eqn. (5.2) should be implemented by examining the population parameter $p(y_j|x_i)$.

The calculation of $\xi(x_i \to y_j)$ is based on their time vectors $\mathbf{t}^{(x_i)}$ and $\mathbf{t}^{(y_j)}$. The frequency is counted as the number of times x_i and y_j occur simultaneously within a time window of size W_{th}. By comparing $\mathbf{t}^{(x_i)}$ and $\mathbf{t}^{(y_j)}$, a matrix of time differences is formulated as

$$\triangle T = [\triangle t]_{|\mathbf{t}^{(y_j)}|\times|\mathbf{t}^{(x_i)}|} = \mathbf{t}^{(y_j)} \cdot \mathbf{I}_{1\times|\mathbf{t}^{(x_i)}|} - \mathbf{I}_{|\mathbf{t}^{(y_j)}|\times 1} \cdot \left(\mathbf{t}^{(x_i)}\right)^T, \tag{5.5}$$

where $\mathbf{I}_{1\times|\mathbf{t}^{(x_i)}|}$ and $\mathbf{I}_{|\mathbf{t}^{(y_j)}|\times 1}$ are all one vectors. By truncating the matrix $\triangle T$ with a widow $[0, W_{th}]$, we have

$$\bigtriangledown T = [\bigtriangledown t]_{|\mathbf{t}^{(y_j)}|\times|\mathbf{t}^{(x_i)}|}, \tag{5.6}$$

where

$$\bigtriangledown t_{ij} = \begin{cases} \triangle t_{ij} & \text{if } \triangle\, t_{ij} \in [0, W_{th}], \\ 0 & \text{otherwise .} \end{cases} \tag{5.7}$$

The transition from $\triangle T$ to $\bigtriangledown T$ filters out the time differences for the pairs of occurrences not satisfying Definition 6. Then, the $\xi(x_i \to y_j)$ is estimated as

$$\xi(x_i \to y_j) = \text{rank } (\bigtriangledown T). \tag{5.8}$$

If $\xi(x_i \to y_j) \geq F_{th}$, the computation proceeds to the next step of hypothesis test to examine the second condition of Definition 8. Otherwise, the computation terminates and we conclude that $x_i \nRightarrow y_j$.

To implement the hypothesis test, we need to check whether the population parameter $p(y_j|x_i)$ is equal to 1 or not. Based on $p(y_j|x_i) = \frac{p(y_j,x_i)}{p(x_i)}$, the hypothesis in eqn. (5.2) is converted to

$$\begin{cases} H_0: & p(y_j, x_i) = p(x_i), \\ H_A: & p(y_j, x_i) \neq p(x_i). \end{cases} \tag{5.9}$$

In this vein, the frequency $\xi(x_i)$ of x_i can be treated as a random variable K following the binomial distribution [36]. The probability function is

$$p(K = k) = \binom{N}{k} p^k (1-p)^{(N-k)}, \tag{5.10}$$

where N is the number of samples and $p = p(x_i)$. Analogously, the frequency $\xi(x_i \rightarrow y_j)$ of the co-occurrence of x_i and y_j also follows the binomial distribution in eqn. (5.19) with $p = p(y_j, x_i)$. Denote the population parameters as $p_1 = p(x_i)$ and $p_2 = p(y_j, x_i)$, and the observation values as $k_1 = \xi(x_i)$, $k_2 = \xi(x_i \rightarrow y_j)$, $\hat{p}_1 = \frac{k_1}{N_1}$, $\hat{p}_2 = \frac{k_2}{N_2}$, and $\hat{p} = \frac{k_1+k_2}{N_1+N_2}$. According to [27], the alternative likelihood function is

$$\begin{aligned} H(\hat{p}_1, \hat{p}_2; k_1, N_1, k_2, N_2) = \binom{N_1}{k_1} p_1^{k_1} (1-p_1)^{(N_1-k_1)} \cdot \binom{N_2}{k_2} \\ p_2^{k_2} (1-p_2)^{(N_2-k_2)}. \end{aligned} \tag{5.11}$$

The null likelihood function $H(\hat{p}, \hat{p}; k_1, N_1, k_2, N_2)$ has the same form of eqn. (5.11) by substituting $\hat{p}_1$ and $\hat{p}_2$ for $\hat{p}$. Then, the logarithmic likelihood statistic is

$$\Lambda = -2 \log \frac{H(\hat{p}, \hat{p}; k_1, N_1, k_2, N_2)}{H(\hat{p}_1, \hat{p}_2; k_1, N_1, k_2, N_2)}. \tag{5.12}$$

By substituting the observation values into eqn. (5.12), the logarithmic likelihood statistic is calculated as

$$\begin{aligned} \Lambda = -2 \Bigg[& k_1 \log \frac{k_1 + k_2}{2k_1} + (N_1 - k_1) \log \frac{N_1 - \frac{k_1+k_2}{2}}{N_1 - k_1} \\ & + k_2 \log \frac{k_1 + k_2}{2k_2} + (N_2 - k_2) \log \frac{N_2 - \frac{k_1+k_2}{2}}{N_2 - k_2} \Bigg]. \end{aligned} \tag{5.13}$$

The parameters N_1 and N_2 are referred to as the time length of the database $\mathbb{D}$. Thus, $N_1 = N_2$, $N_1 \gg k_1$, $N_2 \gg k_2$, $N_1 \gg \frac{k_1+k_2}{2}$, and $N_1 \gg \frac{k_1+k_2}{2}$. Then, we have

$$\lim_{N_1 \rightarrow +\infty} (N_1 - k_1) \log \frac{N_1 - \frac{k_1+k_2}{2}}{N_1 - k_1} = \frac{k_1 - k_2}{2}, \tag{5.14}$$

$$\lim_{N_2 \rightarrow +\infty} (N_2 - k_2) \log \frac{N_2 - \frac{k_1+k_2}{2}}{N_2 - k_2} = \frac{k_2 - k_1}{2}. \tag{5.15}$$

Subsequently, Λ is approximated as

$$\Lambda \approx -2 \left[k_1 \log \frac{k_1 + k_2}{2k_1} + k_2 \log \frac{k_1 + k_2}{2k_2} \right]. \tag{5.16}$$

According to the work in [27], Λ is asymptotically χ^2 distributed with one degree of freedom. The Chi-squared value at the significance level of 0.05 is about 3.84. If $\Lambda > 3.84$, we say that the null hypothesis H_0 is rejected at the significance level of 0.05. Accordingly, $p(y_j, x_i) \neq p(x_i)$, or equally, $p(y_j|x_i) \neq 1$, is accepted. Then, we conclude $x_i \not\Rightarrow y_j$. Otherwise, $x_i \Rightarrow y_j$, if $\Lambda \leq 3.84$. The smaller value Λ implies more significant association rule of $x_i \Rightarrow y_j$. Eventually, the association rules detected for $\forall x_i \in \mathcal{X}$ and $\forall y_j \in \mathcal{Y}$ can be ranked based on the ascending order of Λ.

5.4.3 Mode-Dependent Alarms for Multiple Operating Modes

The detection of mode-dependent alarms for multiple operating modes differs from that for a single operating mode by the calculation of frequencies. The frequent patterns $\tilde{X}^{(k)} \in L_k, k \geq 2$, of multiple operating modes are found in subsection 5.4.1. The frequency $\xi(\tilde{X}^{(k)})$ of each pattern is calculated using Algorithm 5. To determine the association rule of mode-dependent alarms, the first condition in Definition 8 must be checked first. The problem is how to calculate the frequency $\xi(\tilde{X}^{(k)} \rightarrow y_j)$ for each pair of $\tilde{X}^{(k)} \in L_k, k \geq 2$, and $y_j \in \mathcal{Y}$. In general, if one alarm y_j is dependent on a multi-mode pattern $\tilde{X}^{(k)}$, y_j should always occur after the last mode variable $x_i \in \tilde{X}^{(k)}$ occurs. Based on this fact, the frequency $\xi(\tilde{X}^{(k)} \rightarrow y_j)$ can be calculated in the following way.

At first, we need to determine the time vector $\mathbf{t}^{(\tilde{X}^{(k)})}$ with each element indicating the occurrence of the last event for all $x_i \in \tilde{X}^{(k)}$. According to Definition 9, a sub-vector S can be extracted from the incrementally sorted time vector $\tilde{\mathbf{t}}$ in eqn. (5.3), using a sliding time window of size W_{th}. The sub-vector S is formulated as

$$S = [\tilde{t}_{l+1}, \tilde{t}_{l+2}, \cdots, \tilde{t}_{l+|S|}] \tag{5.17}$$

such that

1. $\Delta S = \tilde{t}_{l+|S|} - \tilde{t}_{l+1} \leq W_{th}$;

2. $\mathfrak{S}([h_{l+1}, h_{l+2}, \cdots, h_{l+|S|}]) = \{1, 2, \cdots, k\}$.

where $[h_{l+1}, h_{l+2}, \cdots, h_{l+|S|}]$ in $\mathbf{h}$ is the index vector corresponding to S in $\tilde{\mathbf{t}}$; $\mathfrak{S}(\mathbf{v})$ indicates the set of unique elements in $\mathbf{v}$. The first condition makes sure all events are captured within a time window W_{th}. The second condition guarantees the occurrence of $\forall x_i \in \tilde{X}^{(k)}$ can be found within the time window. Then, $t_{l+|S|}$ is selected as an element of $\mathbf{t}^{(\tilde{X}^{(k)})}$. Finally, the frequency of co-occurrences of $\forall x_i \in \tilde{X}^{(k)}$ is achieved as $\xi(\tilde{X}^{(k)}) = |\mathbf{t}^{(\tilde{X}^{(k)})}|$. A graphical example is shown in Fig. 5.6 to demonstrate how the time vector $\mathbf{t}^{(\tilde{X}^{(2)})}$ is determined for two mode variables. For example, we find the first occurrences of x_1 and x_2 with $\Delta S = t_1^{(x_2)} - t_1^{(x_1)} < W_{th}$ and x_2 happens after x_1, then $t_1^{(x_2)}$ is selected as the first element of $\mathbf{t}^{(\tilde{X}^{(2)})}$. Eventually, $\mathbf{t}^{(\tilde{X}^{(2)})} = [t_1^{(x_2)}, t_2^{(x_1)}, t_4^{(x_1)}, t_5^{(x_2)}]$.

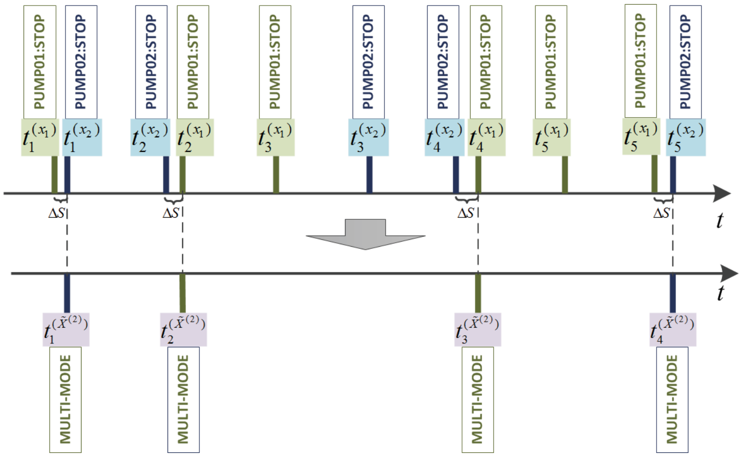


Figure 5.6: Graphical example for producing the time vector of the multi-mode variable $\tilde{X}^{(2)} = \{x_1, x_2\}$ based on $\mathbf{t}^{(x_1)}$ and $\mathbf{t}^{(x_2)}$.

Subsequently, for each pair of $\tilde{X}^{(k)} \in L_k, k \geq 2$, and $y_j \in \mathcal{Y}$, a matrix of time differences analogous to eqn. (5.5) is calculated as

$$\triangle T = [\triangle t]_{|\mathbf{t}^{(y_j)}| \times |\mathbf{t}^{(\tilde{X}^{(k)})}|} = \mathbf{t}^{(y_j)} \cdot \mathbf{I}_{1 \times |\mathbf{t}^{(\tilde{X}^{(k)})}|} - \mathbf{I}_{|\mathbf{t}^{(y_j)}| \times 1} \cdot \left(\mathbf{t}^{(\tilde{X}^{(k)})}\right)^T \tag{5.18}$$

The calculation of $\xi(\tilde{X}^{(k)} \to y_j)$ can be done using eqns. (5.6-5.8). If $\xi(\tilde{X}^{(k)} \to y_j) \geq F_{th}$, the computation proceeds to hypothesis test; otherwise, the computation terminates and we conclude that $\tilde{X}^{(k)} \nrightarrow y_j$.

To implement the hypothesis test, the method is similar to that for a single operating mode in subsection 5.4.2. The hypothesis to determine the association rule $\tilde{X}^{(k)} \Rightarrow y_j$ is formulated as

$$\begin{cases} H_0: & p(y_j, \tilde{X}^{(k)}) = p(\tilde{X}^{(k)}), \\ H_A: & p(y_j, \tilde{X}^{(k)}) \neq p(\tilde{X}^{(k)}). \end{cases} \tag{5.19}$$

The frequency $\xi(\tilde{X}^{(k)})$ of the co-occurrence of multiple variables $\tilde{X}^{(k)}$ can be treated as a random variable following the binomial distribution with the probability function as shown in eqn. (5.10). So it is with the frequency $\xi(\tilde{X}^{(k)} \to y_j)$. Denote the population parameters as $p_1 = p(\tilde{X}^{(k)})$ and $p_2 = p(y_j, \tilde{X}^{(k)})$, and the observation values as $k_1 = \xi(\tilde{X}^{(k)})$, $k_2 = \xi(\tilde{X}^{(k)} \to y_j)$, $\hat{p}_1 = \frac{k_1}{N_1}$, $\hat{p}_2 = \frac{k_2}{N_2}$, and $\hat{p} = \frac{k_1+k_2}{N_1+N_2}$. The rest of the hypothesis test is the same as eqns. (5.11-5.16). The null hypothesis H_0 is rejected at the significance level of 0.05 if $\Lambda > 3.84$. Then, we conclude $\tilde{X}^{(k)} \not\Rightarrow y_j$. Otherwise, $\tilde{X}^{(k)} \Rightarrow y_j$, if $\Lambda \leq 3.84$.

5.4.4 Detection Procedures

Given an A&E log $\mathbb{D}$ on $\mathcal{X} \cup \mathcal{Y}$, the detection procedure of association rules $\psi : \mathcal{X} \to \mathcal{Y}$ is composed of the following steps (and depicted as a flowchart in Fig. 5.7):

1. The A&E log $\mathbb{D}$ on $\mathcal{X}$ and $\mathcal{Y}$ is taken as the input. Two parameters W_{th} and F_{th} are predefined.

2. Extract the sub-database of operating modes $\mathbb{D}(\mathcal{X})$ from $\mathbb{D}$ and detect the frequent k-patterns of operating modes $\tilde{X}^{(k)}$ using Algorithm 3. The frequency of $\tilde{X}^{(k)}$ is calculated as $\xi(\tilde{X}^{(k)}) > F_{th}$. All frequent k patterns $\tilde{X}^{(k)}$ of operating modes are stored in L_k. The collection of all frequent patterns is $\mathcal{L} = L_1 \cup L_2 \cup \cdots \cup L_K$, where K is the maximum size of frequent patterns.

3. The iterative detection of mode-dependent alarms starts from the collection of frequent 1-pattern L_1, and runs till completion of the detection for all frequent K-patterns in L_K.

4. For each collection L_k of frequent k-patterns, the detection is iteratively implemented for each pair of $\tilde{X}_i^{(k)} \in L_k$ and $y_j \in \mathcal{Y}$, from $i = 1, j = 1$ to $i = |L_k|, j = |\mathcal{Y}|$, where $|L_k|$ indicates the number of all frequent k-patterns.

5. To determine the association rule for $\tilde{X}_i^{(k)} \in L_k$ and $y_j \in \mathcal{Y}$, the frequency $\xi(\tilde{X}_i^{(k)} \to y_j)$ is calculated. If $k = 1$, $\tilde{X}_i^{(k)}$ is essentially $x_i \in \mathcal{X}$. Then, $\xi(\tilde{X}_i^{(1)} \to y_j)$ is calculated using eqns. (5.5-5.8). If $k \geq 2$, the time vector $\mathbf{t}^{(\tilde{X}_i^{(k)})}$ should be determined via eqn. (5.17). Eventually, $\xi(\tilde{X}_i^{(k)} \to y_j), k \geq 2$, is calculated using eqns. (5.18,5.6-5.8).

6. Compare $\xi(\tilde{X}_i^{(k)} \to y_j)$ with F_{th}. If $\xi(\tilde{X}^{(k)} \to y_j) \geq F_{th}$, the computation proceeds to the next step of hypothesis test; otherwise, the computation for the current pair of $\tilde{X}_i^{(k)}$ and y_j terminates and we conclude that $\tilde{X}^{(k)} \nRightarrow y_j$.

7. Substitute $k_1 = \xi(\tilde{X}_i^{(k)})$ and $k_2 = \xi(\tilde{X}_i^{(k)} \to y_j)$ into eqn. (5.16). Then, Λ is calculated and compared with the Chi-squared value 3.84 at the significance level of 0.05. The null hypothesis H_0 is rejected if $\Lambda > 3.84$. Then, we conclude $\tilde{X}^{(k)} \nRightarrow y_j$; otherwise, $\tilde{X}^{(k)} \Rightarrow y_j$, if $\Lambda \leq 3.84$.

8. All association rules of mode-dependent alarms $\tilde{X}^{(k)} \Rightarrow y_j$ are discovered and sorted based on the ascending order of Λ and output as the final result $\psi : \mathcal{X} \to \mathcal{Y}$.

The number of detected association rules is eventually decided by the two parameters W_{th} and F_{th}. To set W_{th} and F_{th}, the following principles could be considered. (1) The time window W_{th} is used to determine whether the occurrence of one event is related to another based on Definition 6. If W_{th} is set relatively large, then the unrelated occurrences will be falsely regarded as related. If W_{th} is set relatively small, then some related events will be missed. The best way to set W_{th} should be based on the process. For instance, a small W_{th} is needed for fast processes, while a large W_{th} can be applied to slow processes. (2) The frequency threshold F_{th} is used to make sure the detected association rules are common enough. A small F_{th} may lead to the burst of association rules while an extremely large F_{th} may provide less association rules than expected. As a rule of thumb, F_{th} can be set to be at the median, 1-quantile, or 3-quantiles of $\xi(x_i), \forall x_i \in \mathcal{X}$, depending on how common the consequential nuisance alarms are in the alarm system.

5.5 Industrial Case Studies

This section provides two industrial case studies to demonstrate the practicability and effectiveness of the proposed method in detecting mode-dependant alarms from historical A&E logs.

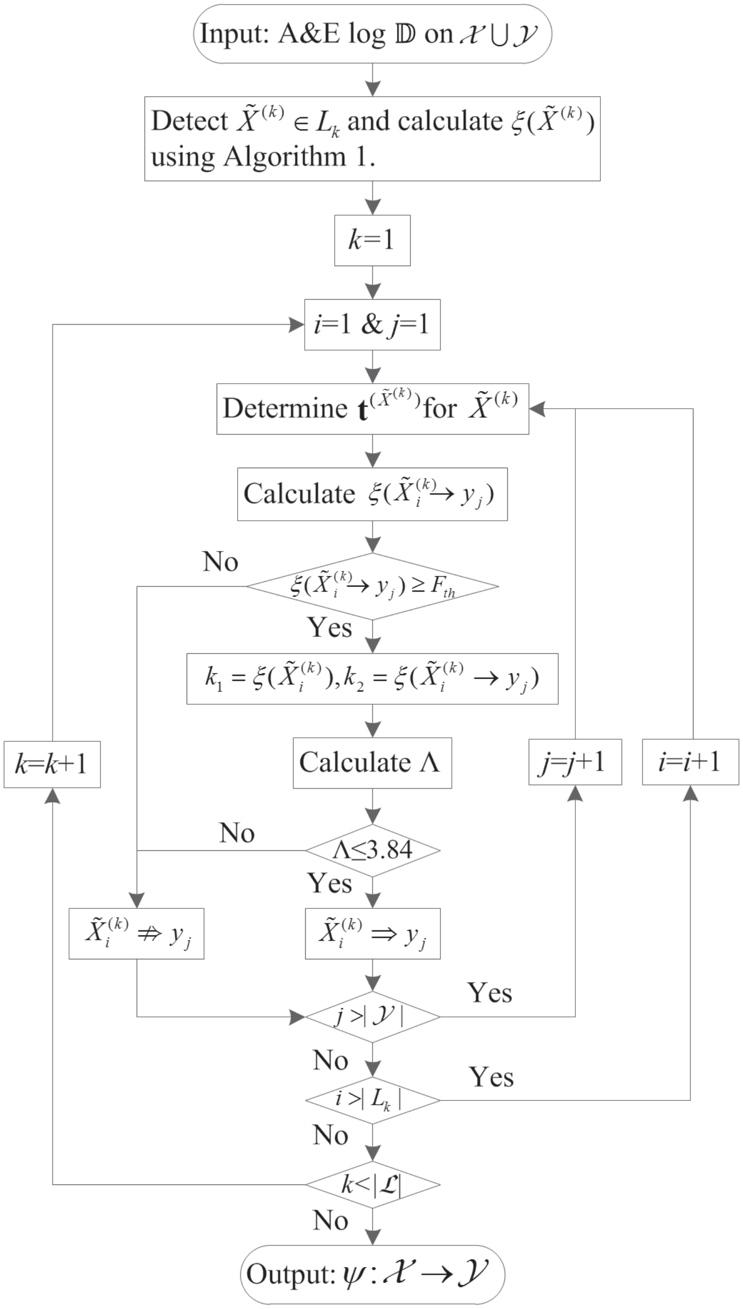

Input: A&E log $\mathbb{D}$ on $\mathcal{X} \cup \mathcal{Y}$
Detect $\tilde{X}^{(k)} \in L_k$ and calculate $\xi(\tilde{X}^{(k)})$ using Algorithm 1.
k=1
i=1 & j=1
Determine $\mathbf{t}^{(\tilde{X}^{(k)})}$ for $\tilde{X}^{(k)}$
Calculate $\xi(\tilde{X}_i^{(k)} \to y_j)$
No
$\xi(\tilde{X}_i^{(k)} \to y_j) \geq F_{th}$
Yes
$k_1 = \xi(\tilde{X}_i^{(k)}), k_2 = \xi(\tilde{X}_i^{(k)} \to y_j)$
k=k+1
Calculate Λ
j=j+1
i=i+1
No
$\Lambda \leq 3.84$
Yes
$\tilde{X}_i^{(k)} \not\Rightarrow y_j$
$\tilde{X}_i^{(k)} \Rightarrow y_j$
Yes
$j > |\mathcal{Y}|$
No
Yes
$i > |L_k|$
No
$k < |\mathcal{L}|$
No
Output: $\psi : \mathcal{X} \to \mathcal{Y}$

Figure 5.7: Flow chart for the procedures to determine association rules of mode-dependent alarms.

5.5.1 Case Study I

The industrial process for this case study is an oil sands extraction plant, which refines bitumen from oil sands and upgrades it into crude oil products. Thousands of unique alarms are configured to monitor the processes of the entire plant. Actions, such as the changes of operating modes are made by operators through the human-machine interface (HMI) and recorded via data acquisition systems. The alarm and event data collected for case studies covered three consecutive months. The total number of alarm occurrences is about 50 thousand and the alarm rate for the entire plant is 3.7 alarms/10 min. Moreover, about 14 thousand events of operating modes were recorded in the A&E log. The sets $\mathcal{X}$ and $\mathcal{Y}$, contains 1061 distinct alarm variables and 1231 distinct mode variables, respectively. This subsection presents the overall results for the implementation of the proposed method.

At first, the time window W_{th} is determined in a practical manner: (1) The time difference Δt_i for each event of operating mode and the nearest followed alarm occurrence is calculated and shown in Fig. 5.8. (2) As a conservative choice, the time window is set to $W_{th} = 109$ sec, which is the median of time differences in Fig. 5.8. Accordingly, about half of the time differences will be used in the calculation of frequencies $\xi(x_i \to y_j)$. If the mode-dependent alarms are very common in an alarm system, the 3-quantiles could be used as W_{th}, so as to tolerate more mode-alarm pairs. In contrast, the 1-quantile might be a choice if the mode-dependent alarms are expected to be rare. In the rest of the case study, $W_{th} = 109$ sec is used.

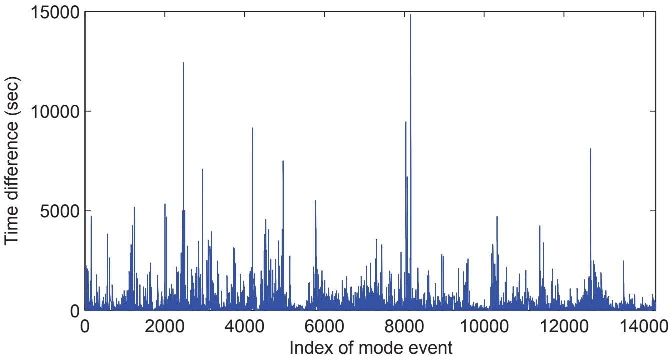


Figure 5.8: The time difference for each event of operating mode and the nearest alarm occurrence following it.

Tests were conducted based on different threshold values for F_{th}: [5, 10, 15, 20, 25, 30]. Prior to the detection of association rules, the frequent patterns of operating modes are extracted first. Then, the association rules of mode-dependent alarms are detected for each mode pattern. It can be found from Tables 5.2 and 5.3 that the number of frequent k-patterns of operating modes and the number of association rules of mode-dependent alarms grow with the decreasing values of F_{th}. For example, for the case with $F_{th} = 5$, there are 1023 mode patterns and 273 association rules. A small F_{th} does not just increase the computational burden, but also causes the difficulty for the users to select association rules of interests. To determine a proper frequency threshold F_{th}, the mean of the Chi-squared values are calculated and compared in Fig. 5.9. A smaller Chi-squared value Λ indicates a more significant association rule. Based on this principle, the frequency threshold can be determined as $F_{th} = 10$ corresponding to the minimum mean value of Λ. This threshold can be used as a recommendation for the same alarm system.

Table 5.2: Number of frequent mode patterns based on different F_{th}.

Frequency	Num. of mode patterns in L_k					
threshold	L_1	L_2	L_3	L_4	L_5	L_6
$F_{th} = 5$	714	426	151	41	8	1
$F_{th} = 10$	396	127	18	2	0	0
$F_{th} = 15$	232	48	3	0	0	0
$F_{th} = 20$	173	24	1	0	0	0
$F_{th} = 25$	126	12	1	0	0	0
$F_{th} = 30$	91	8	1	0	0	0

Table 5.3: Number of association rules based on different F_{th}.

Frequency	Num. of association rules		
threshold	$\tilde{X}_i^{(1)} \rightarrow y_j$	$\tilde{X}_i^{(2)} \rightarrow y_j$	$\tilde{X}_i^{(3)} \rightarrow y_j$
$F_{th} = 5$	129	121	23
$F_{th} = 10$	70	39	10
$F_{th} = 15$	48	18	2
$F_{th} = 20$	25	12	2
$F_{th} = 25$	16	4	1
$F_{th} = 30$	8	3	1

5.5.2 Case Study II

To show how the mode-dependent alarms are determined, two specific examples are selected from Case I. The first example gives the mode-dependent alarms for a single operating mode tagged as "Mode #68", which indicates the stopping of the tailing pump. Over the time period of three months,

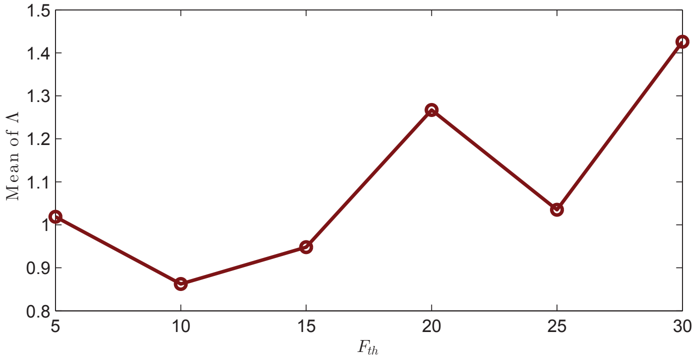


Figure 5.9: The mean of Chi-squared values for all association rules based on different F_{th}.

this pump was stopped 38 times. Using the proposed method, the frequency $\xi(x_i \to y_j)$ and the Chi-squared value Λ are calculated for each alarm and "Mode #68". The metrics for the top four hundred alarms are shown in Fig. 5.10. The association rules of mode-dependent alarms are eventually found as those with $\xi(x_i \to y_j) \geq F_{th} = 10$ and $\Lambda \leq 3.84$.

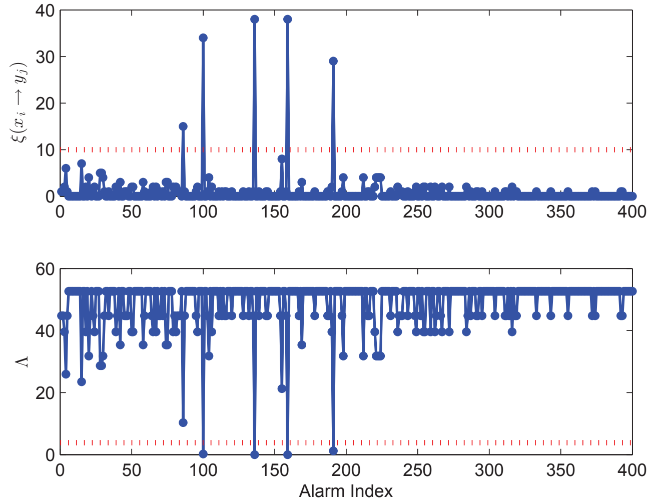


Figure 5.10: The frequencies $\xi(x_i \to y_j)$ and Chi-squared values Λ for the top four hundred alarms and "Mode #68". (The dashed lines in the upper plot and lower plot indicate the frequency threshold $F_{th} = 10$ and the Chi-square threshold 3.84, respectively.)

Four association rules are shown in top rows of Table 5.4 while the last row is presented for comparison. The occurrences of "Mode #68" and the five alarms are shown as the vertical lines on the time axis in Fig. 5.11. It can be seen that "Alarm #159", "Alarm #136", "Alarm #100", and "Alarm #191" were very likely annunciated simultaneously with the occurrences of "Mode #68". In contrast, the occurrences of "Alarm #96" had a very different distribution on the time axis.

Table 5.4: Association rules detected for Mode #68.

Association rules	$\xi(x_i \to y_j)$	$\hat{p}(y_j \vert x_i)$	Λ
Mode#68⇒Alarm#159	38	1	0
Mode#68⇒Alarm#136	38	1	0
Mode#68⇒Alarm#100	34	0.89	0.22
Mode#68⇒Alarm#191	29	0.76	1.21
Mode#68⇏Alarm#96	2	0.05	39.57

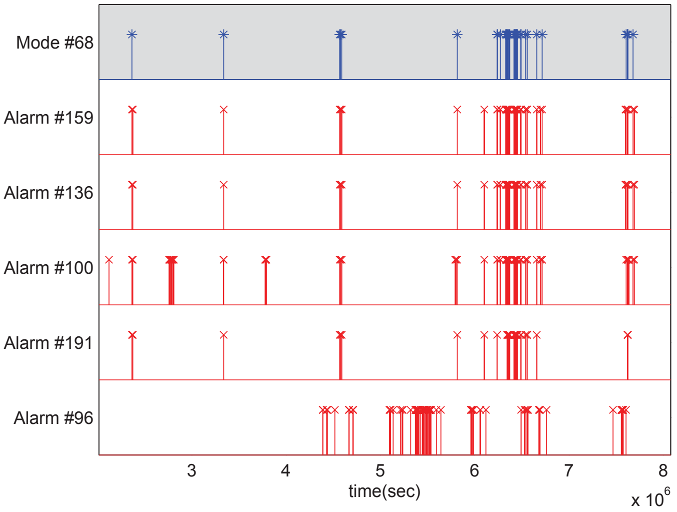


Figure 5.11: Occurrences of "Mode #68" and the five alarms on the time axis. (The vertical lines indicate the time instants when the mode or alarm occurred.)

Table 5.4 gives quantitative indicators for the detected association rules. For instance, "Alarm #159" was found to occur 38 times following the shutdown of the tailing pump. The probability for "Alarm #159" following "Mode #68" is 1. The Chi-squared value $\Lambda = 0$ indicates the association rule "Mode #68 ⇒ Alarm #159" is very significant. The time differences between the occurrence of "Mode #68" and that of "Alarm #159" are shown in Fig. 5.12-(a). It can be seen that "Alarm #159" always occurred within five seconds after the occurrence of “Mode #68”. Comparing the Chi-squared values, “Mode #68 ⇒ Alarm #136” was found as significant as the first rule; while “Mode #68 ⇒ Alarm #100” and “Mode #68 ⇒ Alarm #191” are less significant but nevertheless indicating mode-dependent alarms. In contrast, the association rule for “Alarm #96” has a Chi-squared value of 39.57 which is much larger than the threshold of 3.84. Thus, “Mode #68 $\not\Rightarrow$ Alarm #96” is concluded.

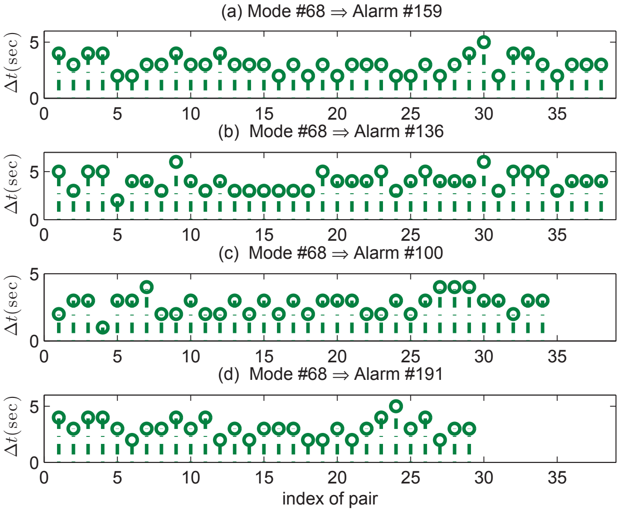


Figure 5.12: Time differences between the occurrences of each mode-alarm pair.

The second example presents mode-dependent alarms for multiple operating modes. The operating modes “Mode #161”, “Mode #179”, and “Mode #188”, represent the shut-down of a booster seal water pump, the shut-down of a lube oil pump, and the shut-down of a lube oil cooling fan, respectively. These operating modes are related to the same lube oil supply system. Over the time period of three months, the three facilities were stopped 21, 19, and 18 times, respectively. Using Algorithm 3, “Mode #161”, “Mode #179”, and “Mode #188” were found to co-occur 14 times, larger than the threshold $F_{th} = 10$. Thus, {Mode #161, Mode #179, Mode #188} is determined to be a frequent 3-pattern. By calculating the Λ for each alarm and the mode pattern, and comparing them with the threshold of 3.84, eventually one association rule is determined as “{Mode #161, Mode #179, Mode #188} ⇒ Alarm #293” with $\xi(x_i \to y_j) = 14$, $\hat{p}(y_j|x_i) = 1$, and $\Lambda = 0$. The alarm “Alarm #293” is the “LOLO” alarm of the discharge lube oil flow. Fig. 5.13

displays the occurrences of the three operating modes and “Alarm #293” on the same time axis. Obviously, “Alarm #293” was annunciated almost simultaneously with the co-occurrences of the three operating modes. Fig. 5.14 shows the time differences, indicating that “Alarm #293” usually occurred within 14 seconds after the co-occurrence of “Modes #161”, “Modes #179”, and “Modes #188”.

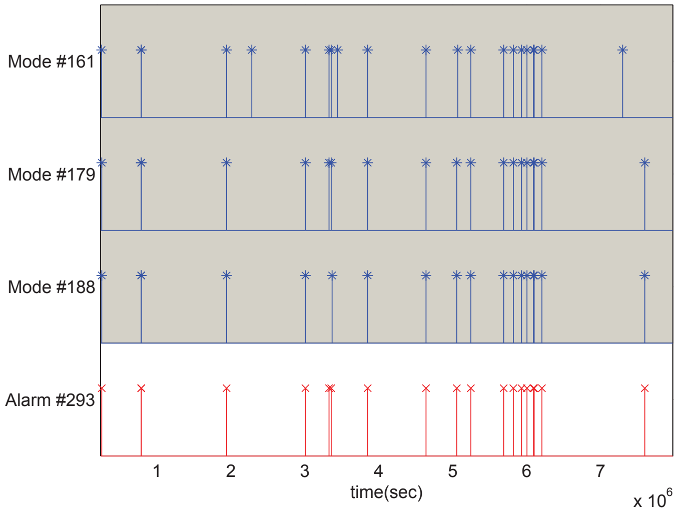


Figure 5.13: Occurrences of “Mode #161”, “Mode #179”, “Mode #188”, and “Alarm #293” on the time axis.

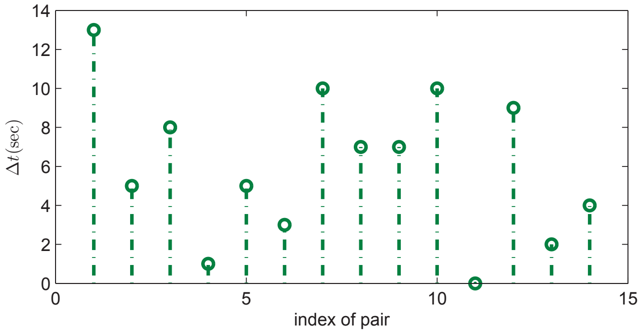


Figure 5.14: Time differences between the occurrence of “Alarm #293” and the co-occurrence of “Modes #161”, “Modes #179”, and “Modes #188”.

5.6 Summary

State-based alarming is an effective strategy to reduce nuisance alarms, and thus should become a routine part of alarm rationalization strategies in process industries. To help with the configuration of state-based alarming in practice, this chapter proposed a data-driven method to detect association rules of mode-dependent alarms from A&E logs. The feasibility is based on the principle that the switching to a specific operating mode is always followed by the occurrence of an alarm. Accordingly, the definitions of mode-dependent alarms and related preliminary concepts are presented. Algorithms to detect frequent patterns of operating modes and association rules of mode-dependent alarms are proposed. The effectiveness of the proposed method is demonstrated by industrial case studies based on real historical A&E data sets from an oil sands plant.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis proposes a variety of advanced techniques for the analysis of alarm data, targeting at improving the efficiency of industrial alarm systems and assisting process operators in managing hazardous situations. The outcomes of the studies in this thesis are summarized as follows:

1. A new method is developed to detect and quantify correlated alarms in the presence of occurrence delays, which are identified as the main causes leading to erroneous conclusions from existing methods. The correlated alarms are determined using a statistical test for random occurrence delays and the correlation level is calculated based on continuous-valued pseudo alarm sequences. The method can help process engineers to detect and remove duplicated alarms, or discover improper alarm system designs.

2. An accelerated local alignment method is proposed to find similar alarm flood sequences, based on the motivation that similar alarm floods are very likely caused by the same root cause. To improve the computational efficiency and accuracy, three novel strategies are incorporated, including the priority-based similarity scoring strategy, the set-based prematching mechanism, and the modified seeding-extending steps. With the proposed method, it is ready for further pursuing the predication and prevention of alarm floods based on similar alarm sequences in historical alarm floods.

3. A causality inference method based on binary-valued alarm data is developed to identify abnormality propagation paths and detect root causes, in the absence of continuous-valued process data. A modified transfer entropy (TE) and a direct transfer entropy (DTE) are formulated based on the two characteristics of alarm signals, namely, the random occurrence delays and the mutual independence of alarm occurrences. The proposed method provides an offline cause and effect analysis, which is valuable in assisting plant operators in the online diagnosis of root causes of abnormalities. The causality relations constructed may also reveal fault propagation pathways, and can be used to determine root causes in alarm floods.

4. A data-driven approach is proposed to discover association rules of mode-dependent alarms from historical A&E logs. The method consists of two major steps: the frequent patterns of operating modes are firstly detected by a modified a-priori algorithm, and then the mode-dependent alarms are determined by a hypothesis test for either a single operating mode or multiple operating modes. The proposed method can help process engineers in discovering consequential nuisance alarms and configuring state-based alarming strategies.

The effectiveness and applicability of the proposed methods are validated by case studies of real industrial alarm data.

6.2 Future Work

The results achieved so far in this thesis are of great value for the improvement of alarm monitoring in large-scale process industries. Nonetheless, there remain many opportunities for further exploration, so as to further improve alarm systems and make them compliance with the benchmark requirements in ISA-18.2 [47] and EEMUA-191 [28]. The following promising directions deserve efforts for future work.

- **Prediction and Prevention of Alarm Floods for Online Application**

For the prediction and prevention of alarm floods, the very first step is to analyze and classify the historical alarm floods in some manner in order to see if alarm floods are predictable. The method proposed in Chapter 3 is able to find similar alarm floods in a quick and accurate way. The unsolved problems are: what are the abnormalities leading to similar alarm floods? Would two different abnormalities result in similar alarm floods? The answers to the questions are closely related to process knowledge that may be hard to obtain and express in a mathematical form. Next, the prediction of alarm floods is meaningful only for online applications. That is, more alarms are coming, similar alarm floods in the historical database are found, and the prediction is made. Thus, the prediction has to be done in an online and recursive manner. Could the proposed algorithm in Chapter 3 be adapted to a recursive form in order to further reduce the computation cost? These questions should be studied in order to achieve the online prediction and prevention of alarm floods.

- **Process Discovery of Operator Actions in Response to Alarms**

The existing research on alarm management only focused on alarms whereas the human factors during abnormal situations are vital but rarely studied. Improper operational procedures can cause further abnormalities and faults, which can potentially result in alarm chattering and floods, accidents, and economical loss. It has been reported that about 80% of the catastrophes are linked to human errors according to studies in [72]. The worker fatigue costs U.S. companies \$77 - \$150 billion per year in increased health care costs, lost production and damage sustained during accidents [21, 89]. In view of these facts, it is of significant importance to study the relations between alarms and operator actions. Specifying proper operational procedures provides the means for decision support capabilities which can potentially result in further enhancement in the efficiency and productivity of the processes as well as in reducing fatigue of operators. There are a variety of questions to be explored: How can the historized information be used to discover the sequences of alarms and the corresponding operator actions? How can the models of proper operational procedures be developed from the discovered sequences? Finally, what potential findings can be specified from the developed models for use as training and enhancement benchmarks? We have made an attempt in this research direction and have a piece of preliminary work published as [44]. This study focuses on a simple case of univariate alarms, whereas the more complicated case of multiple alarms remains unsolved. To make further exploration, the following technical problems should be thought thoroughly: How can A&E logs be decomposed based on the state transitions of multiple alarms? How can the uncertainty of orders of multiple alarms be dealt with in process discovery? Can the time information be extracted and incorporated into the representation models of captured operational procedures?

- **Root Cause Detection with A Combination of Multiple Resources**

As explained in Chapter 1, there are different ways to find causal relations for complex industrial processes depending on the resources. The graph-based and process history based methods have been commonly studied in the past years. Chapter 4 provides an alternative means for causality inference using another type of resource, namely, alarm data. However, no single method can provide satisfactory results for all cases. Each method has its own merits and limitations. Thus, one question to explore is: Can we make effective use of multiple resources for the identification of abnormality propagation paths and detection of root causes? Thambirajah et al. [86] conducted cause-and-effect analysis in chemical processes by combining basic and readily available information about the connectivity of the process with the results from causal measurement-based analysis. Schleburg et al. [79] utilized two resources, namely, the plant connectivity and alarm logs, to analyze the causal relations of alarms and group alarms associated with a common root cause. Furthermore, there are three types of resources, namely, process connectivity information, historical process data, and alarm data. How could we combine them to reduce spurious conclusions and deliver more reliable causality inference? How can the results be used to assist operator in perceiving the true abnormalities and making prompt and correct responses, especially during alarm floods?

Bibliography

[1] Adnan, N.A., Izadi, I., & Chen, T. (2011). On expected detection delays for alarm systems with deadbands and delay-timers. *Journal of Process Control*, 21, 1318-1331.

[2] Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, 1215, 487-499.

[3] Ahmed, K., Izadi, I., Chen, T., Joe, D., & Burton, T. (2013). Similarity analysis of industrial alarm flood data. *IEEE Transactions on Automation Science and Engineering*, 10, 452-457.

[4] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215, 403-410.

[5] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25, 3389-3402.

[6] Babich, I. V., & Moulijn, J. A. (2003). Science and technology of novel processes for deep desulfurization of oil refinery streams: a review. *Fuel*, 82(6), 607-631.

[7] Bauer, M., & Thornhill, N. F. (2008). A practical method for identifying the propagation path of plant-wide disturbances. *Journal of Process Control*, 18(7), 707-719.

[8] Bauer, M., Cox, J. W., Caveness, M. H., Downs, J. J., & Thornhill, N. F. (2007). Finding the direction of disturbance propagation in a chemical

process using transfer entropy. *IEEE Trans. Control Systems Technology*, 15(1), 12-21.

[9] Beebe, D., Ferrer, S., & Logerot, D. (2013). The Connection of peak alarm rates to plant incidents and what you can do to minimize. *Process Safety Progress*, 32, 72-77.

[10] Bhaumik, S., MacGowan, J., & Doraj, V. (2015). Mode based alarm solutions at Syncrude Canada. In *Proceedings of 9th IFAC Symposium on Advanced Control of Chemical Processes*, Whistler, Canada, 653-656.

[11] Bouamama, B. O., Biswas, G., Loureiro, R., & Merzouki, R. (2014). Graphical methods for diagnosis of dynamic systems: Review. *Annual Reviews in Control*, 38(2), 199-219.

[12] Bransby, M. L., & Jenkinson, J. (1998). *The management of alarm systems*. HSE Books.

[13] Brown, N. (2003). Alarm Management: The EEMUA guidelines in practice. *Measurement and Control*, 36(4), 114-119.

[14] Bullemer, P. T., Tolsma, M., Reising, D. V. C., & Laberge, J. C. (2011). Towards improving operator alarm flood responses: alternative alarm presentation techniques, in *ISA Automation Week*, Mobile, Alabama.

[15] Charbonnier, S., Bouchair, N., & Gayet, P. (2015). A weighted dissimilarity index to isolate faults during alarm floods. *Control Engineering Practice*, 45, 110-122.

[16] Charbonnier, S., Bouchair, N., & Gayet, P. (2016). Fault template extraction to assist operators during industrial alarm floods. *Engineering Applications of Artificial Intelligence*, 50, 32-44.

[17] Cheng, Y., Izadi, I., & Chen, T. (2013). Pattern matching of alarm flood sequences by a modified Smith-Waterman algorithm. *Chemical Engineering Research and Design*, 91, 1085-1094.

[18] Cheng, Y., Izadi, I., & Chen, T. (2013). Optimal alarm signal processing: filter design and performance analysis. *IEEE Transactions on Automation Science and Engineering*, 10, 446-451.

[19] Chiang, L. H., & Braatz, R. D. (2003). Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics and Intelligent Laboratory Systems*, 65(2), 159-178.

[20] Císař, P., Hošt'álková, E., & Stluka, P. (2009). Data mining techniques for alarm rationalization. *19th European Symposium on Computer Aided Process Engineering*, Cracow, Poland, 1457-1462.

[21] Coburn, E. (1997). Shiftworker fatigue: The $77 billion problem. *Cost Engineering*, 39(4), 26-28.

[22] Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4), 309-347.

[23] Couper, J. R., Penney, W. R., & Fair, J. R. (2009). *Chemical Process Equipment revised 2E: Selection and Design*. Gulf Professional Publishing.

[24] Dash, S., & Venkatasubramanian, V. (2000). Challenges in the industrial applications of fault diagnostic systems. *Computers & Chemical Engineering*, 24(2), 785-791.

[25] Duan, P., Yang, F., Chen, T., & Shah, S. L. (2013). Direct causality detection via the transfer entropy approach. *IEEE Trans. Control Systems Technology*, 21(6), 2052-2066.

[26] Duan, P., Yang, F., Shah, S.L., & Chen, T. (2015). Transfer zero-entropy and its application for capturing cause and effect relationship between variables. *IEEE Trans. Control Systems Technology*, 23(3), 855-867.

[27] Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1), 61-74.

[28] EEMUA (Engineering Equipment and Materials Users' Association) (2013). *Alarm Systems: A Guide to Design, Management and Procurement, Edition 3*. London: EEMUA Publication 191.

[29] Folmer, J., & Vogel-Heuser, B. (2012). Computing dependent industrial alarms for alarm flood reduction. *The 9th Int. Multi-Conf. Systems, Signals and Devices*, March 20-23, 2012, pp.1-6.

[30] Folmer, J., Schuricht, F., & Vogel-Heuser, B. (2014). Detection of temporal dependencies in alarm time series of industrial plants, in *Proceedings of the 19th IFAC World Congress*, 19, 1802-1807.

[31] Foong, O.M., Sulaiman, S.B., Rambli, D.R.B.A., & Abdullah, N.S.B. (2009). ALAP: alarm prioritization system for oil refinery, in *Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, USA.

[32] Gandy, A. (2009). Sequential implementation of Monte Carlo tests with uniformly bounded resampling risk. *J. American Statistical Association*, 104(488), 1504-1511.

[33] Gourévitch, B., & Eggermont, J. J. (2007). Evaluating information transfer between auditory cortical neurons. *J. Neurophysiology*, 97(3), 2533-2543.

[34] Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, 424-438.

[35] Gupta, A., Giridhar, A., Venkatasubramanian, V., & Reklaitis, G. V. (2013). Intelligent alarm management applied to continuous pharmaceutical tablet manufacturing: An integrated approach. *Industrial & Engineering Chemistry Research*, 52(35), 12357-12368.

[36] Hämäläinen, W. (2010). StatApriori: an efficient algorithm for searching statistically significant association rules. Knowledge and information systems, 23(3), 373-399.

[37] Hlaváčková-Schindler, K., Paluš, M., Vejmelka, M., & Bhattacharya, J. (2007). Causality detection based on information-theoretic approaches in time series analysis. *Physics Reports*, 441(1), 1-46.

[38] Hollifield, B., & Habibi, E. (2011). *Alarm Management: A Comprehensuve Guide*. Research Traingle Park, NC: ISA.

[39] Hope, A. C. (1968). A simplified Monte Carlo significance test procedure. *Journal of the Royal Statistical Society. Series B (Methodological)*, 582-598.

[40] HSE (1997). *The explosion and fires at the Texaco Refinery, Milford Haven, 24 July 1994: A report of the investigation by the Health and Safety Executive into the explosion and fires on the Pembroke Cracking Company Plant at the Texaco Refinery, Milford Haven on 24 July 1994*. Health and Safety Executive.

[41] Hu, W., Wang, J., & Chen, T. (2015). A new method to detect and quantify correlated alarms with occurrence delays. *Computers & Chemical Engineering*, 80, 189-198.

[42] Hu, W., Wang, J., & Chen, T. (2016). A local alignment approach to similarity analysis of industrial alarm flood sequences. *Control Engineering Practice*, 55, 13-25.

[43] Hu, W., Wang, J., & Chen, T. (2015). Fast sequence alignment for comparing industrial alarm floods. In *Proceedings of 9th IFAC Symposium on Advanced Control of Chemical Processes*, Whistler, Canada, 647-652.

[44] Hu, W., Ahmad W. A., Chen, T., & Shah, S. L. (2016). Process discovery of operator actions in response to univariate alarms. In *11th IFAC Symposium on Dynamics and Control of Process SystemsIncluding Biosystems*, Trondheim, Norway, 1026-1031.

[45] Hu, W., Afzal, M. S., Brandt, G., Lau, E., Chen, T., & Shah, S. L. (2015). An application of advanced alarm management tools to an oil sand extraction plant. In *Proceedings of 9th IFAC Symposium on Advanced Control of Chemical Processes*, Whistler, Canada, 641-646.

[46] IEC (International Electrotechnical Commission) (2014). *Management of Alarm Systems for the Process Industries*. IEC 62682.

[47] ISA (International Society of Automation) (2009). *Management of Alarm Systems for the Process Industries*. North Carolina: ISA 18.02.

[48] Ito, S., Hansen, M. E., Heiland, R., Lumsdaine, A., Litke, A. M., & Beggs, J. M. (2011). Extending transfer entropy improves identification of effective connectivity in a spiking cortical network model. *PloS one*, 6(11), e27431.

[49] Izadi, I., Shah, S. L., & Chen, T. (2010). Effective resource utilization for alarm management. *The 49th IEEE Conference on Decision and Control (CDC2010)*, 6803-6808.

[50] Jang, G. S., Suh, S. M., Kim, S. K., Suh, Y. S., & Park, J. Y. (2013). A proactive alarm reduction method and its human factors validation test for a main control room for SMART. *Annals of Nuclear Energy*, 51, 125-134.

[51] Jerhotova, E., Sikora, M., & Stluka, P. (2013). Dynamic alarm management in next generation process control systems. *In Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services*. Berlin Heidelberg: Springer.

[52] Jiang, H., Patwardhan, R., & Shah, S. L. (2009). Root cause diagnosis of plant-wide oscillations using the concept of adjacency matrix. *J. Process Control*, 19(8), 1347-1354.

[53] Jiang, H., Choudhury, M. S., & Shah, S. L. (2007). Detection and diagnosis of plant-wide oscillations from industrial data using the spectral envelope method. *J. Process Control*, 17(2), 143-155.

[54] Johnson, R.A., & Wichern, D.W. (2002). *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ: Prentice hall.

[55] Kaiser, A., & Schreiber, T. (2002). Information transfer in continuous processes. *Physica D: Nonlinear Phenomena*, 166(1), 43-62.

[56] Kent, W. J. (2002). BLAT - the BLAST-like alignment tool. *Genome research*, 4, 656-664.

[57] Kondaveeti, S.R., Izadi, I., Shah, S.L., Black, T., & Chen, T. (2012). Graphical tools for routine assessment of industrial alarm systems. *Computers and Chemical Engineering*, 46, 39-47.

[58] Kondaveeti, S. R., Izadi, I., Shah, S. L., Shook, D. S., Kadali, R., & Chen, T. (2013). Quantification of alarm chatter based on run length distributions. *Chemical Engineering Research and Design*, 91, 2550-2558.

[59] Kordic, S., Lam, P., Xiao, J., & Li, H. (2007). Associative data mining for alarm groupings in chemical processes. *International Conference on Intelligent Systems and Knowledge Engineering 2007*. Atlantis Press.

[60] Koscielny, J. M., Bartys, M., & Syfert, M. (2012). Method of multiple fault isolation in large scale systems. *IEEE Trans. Control Systems Technology*, 20(5), 1302-1310.

[61] Kourti, T. (2002). Process analysis and abnormal situation detection: from theory to practice. *IEEE Control Systems Magazine*, 22(5), 10-25.

[62] Laberge, J.C., Bullermer, P., Tolsma, M., & Reising, D.V.C. (2014). Addressing alarm flood situations in the process industries through alarm summary display design and alarm response strategy. *Int. J. Industrial Ergonomics*, 44, 395-406.

[63] Lee, Y. H., Jin, H. D., & Han, C. (2006). On-line process state classification for adaptive monitoring. *Industrial & engineering chemistry research*, 45(9), 3095-3107.

[64] Li, D., Hu, J., Wang, H., & Huang, W. (2015). A distributed parallel alarm management strategy for alarm reduction in chemical plants. *Journal of Process Control*, 34, 117-125.

[65] Mahadevan, N., Dubey, A., Karsai, G., Srivastava, A., & Liu, C. C. (2014). Temporal causal diagrams for diagnosing failures in cyber-physical systems. In *Annual Conference of the Prognostics and Health Management Society*.

[66] Montgomery, D. C., & Runger, G. C. (2010). *Applied statistics and probability for engineers*. John Wiley & Sons.

[67] Mosterman, P. J., & Biswas, G. (1999). Diagnosis of continuous valued systems in transient operating regions. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 29(6), 554-565.

[68] Naghoosi, E., Huang, B., Domlan, E., & Kadali, R. (2013). Information transfer methods in causality analysis of process variables with an industrial application. *J. Process Control*, 23(9), 1296-1305.

[69] Naghoosi, E., Izadi, I., & Chen, T. (2011). Estimation of alarm chattering. *Journal of Process Control*, 21, 1243-1249.

[70] Nair, G. N. (2013). A nonstochastic information theory for communication and state estimation. *IEEE Trans. Automatic Control*, 58(6), 1497-1510.

[71] Nihlwing, C., Kaarstad, M(2012). The development and usability test of a state based alarm system for a nuclear power plant simulator. *8th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies 2012*, 3, pp. 1904-1913.

[72] Nimmo, I. (2002). It's time to consider human factors in alarm management. *Chemical engineering progress*, 98(11), 30-38.

[73] Nimmo, I. (2005). Rescue your plant from alarm overload. *Chemical Processing*, 28-33.

[74] Noda, M., Higuchi, F., Takai, T., & Nishitani, H. (2011). Event correlation analysis for alarm system rationalization. *Asia-Pacific Journal of Chemical Engineering*, 6, 497-502.

[75] Parker, B. (2010). How to avoid alarm overload with centralized alarm management. *Power*, 154(2), 38-41.

[76] Phipson, B., & Smyth, G. K. (2010). Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*, 9(1).

[77] Qin, S. J. (2012). Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, 36(2), 220-234.

[78] Rothenberg, D.H. (2009). *Alarm Management for Process Control: A Best-practice Guide for Design, Implementation, and Use of Industrial Alarm Systems*. NewYork: Momentum Press.

[79] Schleburg, M., Christiansen, L., Thornhill, N.F., & Fay, A. (2013). A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system. *Journal of Process Control*, 23, 839-851.

[80] Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, 85(2), 461.

[81] Schreiber, T., & Schmitz, A. (2000). Surrogate time series. *Physica D: Nonlinear Phenomena*, 142(3), 346-382.

[82] Silverman, B.W. (1998). *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.

[83] Simeu-Abazi, Z., Lefebvre, A., & Derain, J. P. (2011). A methodology of alarm filtering using dynamic fault tree. *Reliability Engineering & System Safety*, 96(2), 257-266.

[84] Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147, 195-197.

[85] Stauffer, T., Bogdan, J., & Booth, S. (2011). Managing alarms using rationalization. *Control Engineering*, 58(3), 30-35.

[86] Thambirajah, J., Benabbas, L., Bauer, M., & Thornhill, N. F. (2009). Cause-and-effect analysis in chemical processes utilizing XML, plant connectivity and quantitative process history. *Computers & Chemical Engineering*, 33(2), 503-512.

[87] Timms, C. (2009). Hazards equal trips or alarms or both. *Process Safety and Environmental Protection*, 87(1), 3-13.

[88] Vogel-Heuser, B., Schütz, D., & Folmer, J. (2015). Criteria-based alarm flood pattern recognition using historical data from automated production systems (aPS). *Mechatronics*.

[89] Walker, B. A., SMITH, K. D., & KEKICH, M. D. (2003). Limiting shiftwork fatigue in process control. *Chemical engineering progress*, 99(4), 54-57.

[90] Wang, J., & Chen, T. (2013). An online method for detection and reduction of chattering alarms due to oscillation. *Computers and Chemical Engineering*, 54, 140-150.

[91] Wang, J., & Chen, T. (2014). An online method to remove chattering and repeating alarms based on alarm durations and intervals. *Computers and Chemical Engineering*, 67, 43-52.

[92] Wang, J., Li, H., Huang, J., & Su, C. (2015). A data similarity based analysis to consequential alarms of industrial processes. *Journal of Loss Prevention in the Process Industries*, 35, 29-34.

[93] Wang, J., Yang, F., Chen, T., & Shah, S.L.(2015). An Overview of Industrial Alarm Systems: Main Causes for Alarm Overloading, Research Status, and Open Problems. *Automation Science and Engineering, IEEE Transactions on*, PP(99), 1-17.

[94] Wyner, A.D., & Ziv, J. (1973). A theorem on the entropy of certain binary sequences and applications–I. *IEEE Trans. Information Theory*, 19(6), 769-772.

[95] Xu, J., Wang, J., Izadi, I., & Chen, T. (2012). Performance assessment and design for univariate alarm systems based on FAR, MAR, and AAD. *IEEE Transactions on Automation Science and Engineering*, 9, 296-307.

[96] Yang, F., Duan, P., Shah, S. L., & Chen, T. (2014). *Capturing Connectivity and Causality in Complex Industrial Processes*. Springer Science & Business Media.

[97] Yang, F., Xiao, D., & Shah, S. L. (2010).Qualitative fault detection and hazard analysis based on signed directed graphs for large-scale complex systemsm. *Fault Detection*. INTECH Open Access Publisher.

[98] Yang, F., Shah, S.L. & Xiao D. (2010). Correlation analysis of alarm data and alarm limit design for industrial processes. *In Proc. American Control Conference (ACC)*, 5850-5855.

[99] Yang, F., Shah, S.L., Xiao, D., & Chen, T. (2012). Improved correlation analysis and visualization of industrial alarm data. *ISA Transactions*, 51, 499-506.

[100] Yang, Z., Wang, J., & Chen, T. (2013). Detection of correlated alarms based on similarity coefficients of binary data. *IEEE Transactions on Automation Science and Engineering/*, 10, 1014-1025.

[101] Yu, W., & Yang, F. (2014). Detection of causality between industrial alarm data based on transfer entropy, in *Proceedings of the International*

Conference on Chemical Engineering 2014, ICChE2014, 29-30 December, Dhaka, Bangladesh.

[102] Zhu, J., Shu, Y., Zhao, J., & Yang, F. (2013). A dynamic alarm management strategy for chemical process transitions. *Journal of Loss Prevention in the Process Industries*, 30, 207-218.