Advanced Methods for Alarm Monitoring and Alarm Flood Analysis Based on Industrial Data

by

Boyuan Zhou

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 in process industries to ensure process safety and efficiency. However, the number of alarms has grown exponentially in the past decades due to easy alarm configuration in computerized systems and lack of proper alarm rationalization. As a consequence, the presence of nuisance alarms and alarm floods severely compromise the performance of alarm systems: nuisance alarms could distract operators with incorrect indications of abnormalities; while alarm floods lead to increased operational risks as operators are overwhelmed by massive alarms and thus may overlook critical alarms. Motivated by this, this thesis focuses on the development of data-driven methods for alarm monitoring and alarm flood analysis.

Three research topics are considered. First, to address nuisance alarms during the start-up operations of industrial equipment, a new alarm monitoring method is proposed, which comprises an offline design stage to capture equipment dynamics during start-ups and an online algorithm for alarm monitoring. Second, a systematic pattern matching method is proposed to capture similar alarm floods across different processes, where the alarms are associated with the same fault types, but configured with different tag names. The obtained results could facilitate root cause analysis and lead to generalized solutions. Last, a pattern mining method is proposed to extract compact alarm sequential patterns from Alarm & Event (A&E) logs with the incorporation of time stamps, tolerance of alarm order switchings, and distillation of compact

results. Therefore, the proposed method is capable of avoiding influences of order ambiguities and also minimizing the redundancy of extracted patterns.

The effectiveness and practicality of the proposed methods are demonstrated by case studies using alarm data from a large-scale industrial facility. Based on the proposed methods, equipment start-ups are effectively monitored while nuisance alarms are significantly reduced; notable alarm sequential patterns are discovered from historical alarm data, which could facilitate alarm suppression, root cause analysis, and decision support for operators.

Preface

The research work in Chapters 2–4 of the thesis was part of an international research collaboration with Dr. Wenkai Hu at China University of Geosciences. The ideas in Chapters 2–3 were from discussions with Dr. Wenkai Hu. The ideas in Chapter 4 were my original ideas. The algorithms, mathematical derivations, and industrial case studies were my original work, as well as the introduction in Chapter 1.

- Chapter 2 has been published as Boyuan Zhou, Wenkai Hu, and Tongwen Chen, A new method for alarm monitoring of equipment start-up operations with applications to pumps. *Industrial & Engineering Chemistry Research*, 58(26), 11251–11260, 2019.

- Chapter 3 has been published as Boyuan Zhou, Wenkai Hu, Kevin Brown, and Tongwen Chen, Generalized pattern matching of industrial alarm flood sequences via word processing and sequence alignment. *IEEE Transactions on Industrial Electronics*, early access. A short version has been published as Boyuan Zhou, Wenkai Hu, and Tongwen Chen, Cross-process alarm flood similarity analysis based on abstracted alarm descriptors. *28th IEEE International Symposium on Industrial Electronics (ISIE)*, 1749–1754, Vancouver, Canada, 2019.

- Chapter 4 has been submitted for publication as Boyuan Zhou, Wenkai Hu, and Tongwen Chen, Pattern extraction from industrial alarm flood sequences by a modified CloFAST algorithm. *IEEE Transactions on Industrial Informatics*.

Acknowledgements

I would like to express my sincere gratitude to many people around me for their help and support. First and foremost, I would like to thank my supervisor, Prof. Tongwen Chen, for his encouragement, guidance, and support. He gave me many valuable opportunities and large freedom to explore interesting research topics and participate in industrial collaborations for getting practical experience. The past years of working under his supervision are unforgettable and irreplaceable in my life.

I would also like to thank my committee members Drs. Henry Leung, Zukui Li, Qing Zhao, Mahdi Tavakoli Afshari for their precious time and valuable feedback.

Furthermore, I am deeply thankful to those who have contributed to the collaborative research and discussions, with a special mention of Dr. Wenkai Hu (China University of Geosciences) and Kevin Brown.

I also wish to acknowledge the financial support from Natural Sciences and Engineering Research Council of Canada (NSERC) and many industrial partners, including Enbridge, Suncor, and Syncrude.

Moreover, I would like to thank all current and previous members of the research group, especially Jun Shang, Mohammad Hossein Roohi, Harikrishna Rao Mohan Rao, and Md Rezwan Parvez for their kindness, encouragement, and help.

Last but not the least, greatest thanks are also to my family members and friends for their companion mentally or physically over the past years.

Again, to people mentioned or not, my thankfulness and gratitude are beyond words and can never be written well enough.

Contents

List of Tables

List of Figures

List of Symbols

$a(t)$	Alarm Signal
$\overline{f}_i(t)$ ($\underline{f}_i(t)$)	Upper (Lower) Bound of Dynamic Alarm Limits
Δt_s	Exact Alarm Unsuppress Delay Time
$\mathbf{X}_i$	Data Matrix of Process Variable Measurement
$\zeta(X_{is}, \mathcal{N}(X_{is}))$	Distance Metric
$d(X_{is}, X_{ik})$	Manhattan Distance
P_{is}	Local Outlier Probability (LoOP) Value
$a = (\epsilon_a, \zeta_a, \chi_a)$	Description of Alarm
$E = (a, t, s)$	Event Record in A&E log
$\tilde{\mathbb{X}}$	Set Containing Bag-of-Words Models
ξ_1, ξ_2, ξ_3	Three Criteria to Remove Stop Words
$\mathcal{E}$	Abstracted Alarm Descriptor
Φ_0	Szymkiewicz-Simpson Coefficient
$\mathbf{W}_{\mathcal{F}}$	Time Weighting Matrix
$\mathbf{H}$	Score Matrix for Dyanmic Programming
μ	Mismatch Penalty
δ	Gap Penalty
$\mathbb{F}$	Dataset of Alarm Flood Sequences
$\mathcal{I}$	Alarm Itemset
$\bar{\sigma}_{\mathcal{I}}$	Minimum Itemset Support

$\mathcal{S}$	Alarm Sequence
$\bar{\sigma}_{\mathcal{S}}$	Minimum Sequence Support
$J_{i,j}$	Element of a Binary Matrix
$\Psi(\mathcal{I})$	Itemset Time-List
$\sigma(\mathcal{I})$	Support Value of Alarm Itemset
$\Omega(\mathcal{S})$	Sequence Time-List
$\sigma(\mathcal{S})$	Support of Alarm Sequence
$\mathcal{P}_n$	Compact Alarm Sequential Pattern

List of Acronyms

A&E	Alarm & Event
AAD	Averaged Alarm Delay
BPCS	Basic Process Control System
DCS	Distributed Control System
FAR	False Alarm Rate
HMI	Human-Machine Interface
KDE	Kernel Density Estimation
LoOP	Local Outlier Probability
MAR	Missed Alarm Rate
PCA	Principal Component Analysis
PDF	Probability Density Function
SCADA	Supervisory Control and Data Acquisition
SIS	Safety Instrumented System
VFD	Variable Frequency Drive

Chapter 1

Introduction

In this chapter, the research background for alarm systems and alarm management is introduced and a literature survey is provided to summarize the recent development of alarm management methods. Thereafter, the contributions of the thesis are listed, followed by thesis outline.

1.1 Research Background

Alarm systems are core assets for complex industrial facilities to ensure process safety and efficiency. The development and applications of new digital technologies make traditional process industries much improved in terms of productivity, safety, and operational costs in the era of industry 4.0 [90]. However, due to easy alarm configuration in computerized systems and lack of proper alarm rationalization, the number of alarms has grown exponentially in the past decades, causing many practical problems for alarm management, such as nuisance alarms and alarm floods. In practice, the key functionality of alarm systems is to help monitor process operations by clearly indicating abnormalities to operators. But the efficiency of alarm systems could be severely impaired by nuisance alarms, which are annunciated excessively, unnecessarily, or even do not return to normal after correct responses are taken [49]. Moreover, alarm floods are regarded as the most challenging issue among the problems for alarm management. According to industrial standards [32, 49], an alarm flood is a situation that the number of annunciated alarms is more than what an operator can effectively manage. As a result, alarm floods lead to increased operational risks as operators are overwhelmed by massive alarms and thus may overlook critical alarms. For instance, alarm floods were identified as the main culprits for accidents in Texaco Refinery [37] and Three Mile Island Nuclear Plant [76], which ended up with huge losses in economy, health, environment, and reputation. To increase the safety of industrial process operations with better alarm performance, data-driven methods for alarm monitoring and alarm flood analysis are developed in this thesis.

1.1.1 Alarm Systems and Alarm Management

With the advancement of Distributed Control Systems (DCS) and Supervisory Control and Data Acquisition (SCADA) systems over the past decades, configurations of complicated alarm systems in process industry become feasible, making alarm systems critical components in modern industrial facilities. For advanced control and monitoring of industrial facilities, abnormalities in process operations are indicated to plant operators by corresponding alarms in audible and/or visible means based on the configured alarm systems, which have many sensors for process measurement and different thresholds/mechanisms for abnormality detection. As a result, an alarm system is comprised of a complex collection of hardware and software that are interconnected hierarchically. The typical configuration of an alarm system is shown in Fig. 1.1 [49], where the major components are: 1) Basic Process Control System (BPCS) and Safety Instrumented System (SIS) that generate alarm signals based on sensor measurements and associated logic conditions; 2) Human-Machine Interface (HMI) that presents the generated alarm signals and receives the operation commands from operators; and 3) alarm logs that record the historical alarms and events, which are extensively used for data analysis to evaluate alarm system performance and design better alarming techniques.

However, many industrial alarm systems suffer from poor performance in

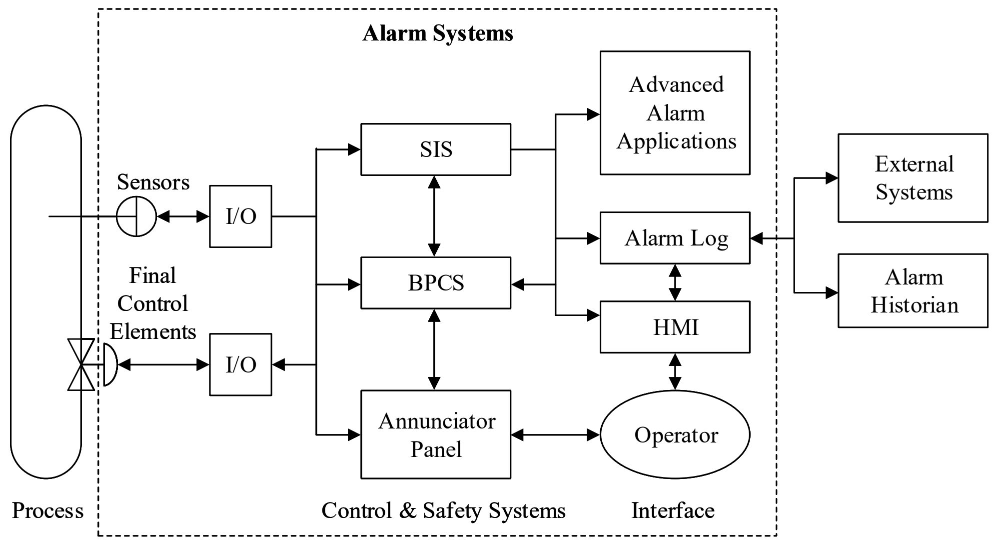

Figure 1.1: Alarm system dataflow [49].

Table 1.1: Alarm system performance survey [75].

Performance Measurement	EEMUA	Oil & Gas	Petrochemical	Power
Average alarms per hour	6	36	54	48
Average standing alarms	9	50	100	65
Peak alarms per hour	60	1320	1080	2100
Priority distribution % (low/med/high)	80/15/5	25/40/35	25/40/35	25/40/35

practice, as reported in a comprehensive survey [75] (page 125) that covers 39 industrial plants ranging from oil and gas, petrochemical, power, and other industries; for further demonstration, the list of several basic performance metrics of alarm systems are summarized in Table 1.1. This table clearly indicates that there still exist significant gaps in alarm performance metrics between the studied industrial systems and the benchmark provided by industrial standard EEMUA-191 [32].

Moreover, due to the practical requirement for process operation and high expectations for safety and efficiency of industrial facilities, effective management of alarm systems becomes sophisticated and time-consuming. Therefore, the complicated problems for alarm management are usually divided into several smaller stages, so as to determine clarified objectives with reduced complexity. To capture the comprehensive stages for alarm system management associated with different objectives and necessary resources, the typical lifecycle of alarm management is summarized in Fig. 1.2 [49], where these stages are connected with directed arrows that indicate work flow. In this alarm management lifecycle, many useful stages are covered, such as rationalization, detailed design, monitoring and assessment, and management of change. Following such divided stages in the provided lifecycle, the complicated tasks of alarm management become more handy and easier to achieve. For instance, in the design stage, more attentions could be paid to problems associated with alarm configurations using basic or advanced alarming techniques, and the improvement of HMI for alarm presentation. In addition, alarm systems are highly associated with safe and efficient operations of industrial facilities, and thus the performance evaluation and functionality improvement of alarm systems should be performed regularly. For this purpose, historical operational data (including alarm data and process data) are commonly used to help with comprehensive analysis, such that better alarming strategies could be identified and meaningful alarm patterns could be discovered.

1.1.2 Current Status of Alarm Management

Alarm systems with satisfactory performance must promptly detect abnormalities in process operation and effectively send out warning information to operators while such information should not mislead, overload, or distract the operators [75]. However, nuisance alarms and alarm floods commonly exist in industrial facilities, severely compromising the performance of alarm systems and causing many challenging problems for alarm management [12, 38].

Nuisance alarms are general descriptions for alarms that annunciate excessively, unnecessarily, or even do not return to normal after the correct responses are taken [49]. Consequently, nuisance alarms are taken as one of the major culprits for alarm overloading and thus severely compromise the performance of alarm systems [96]. In practice, nuisance alarms could be caused by many factors, such as process noises, operational disturbances, incorrect

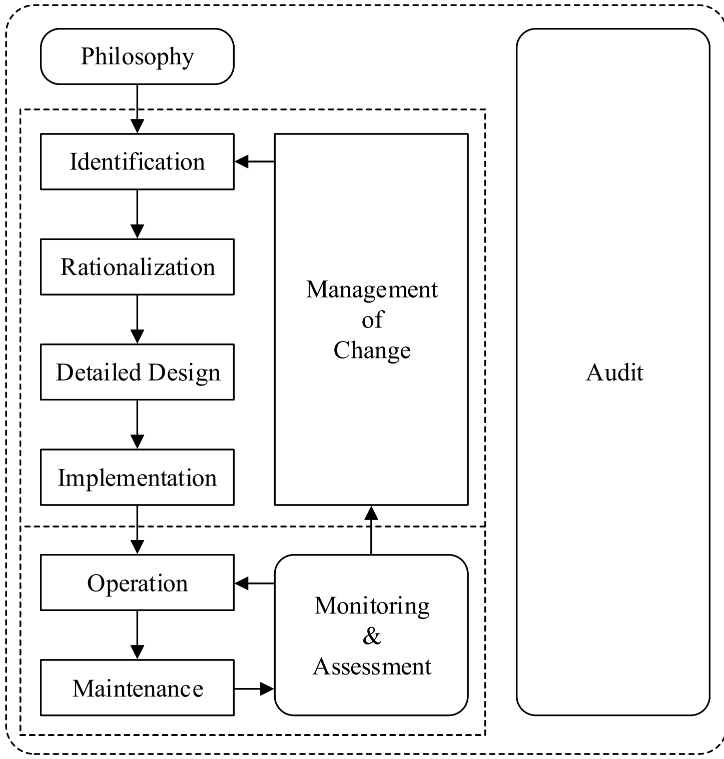

Figure 1.2: Alarm management lifecycle [49].

alarm configurations, and correlation of process variables [11, 96]. Accordingly, the typical types of nuisance alarms include chattering alarms, fleeting alarms, standing alarms, and correlated alarms [32, 49]. Chattering alarms are defined as alarms that are repeatedly triggered within a short period of time. According to the industrial standard ANSI/ISA-18.2 [49], an alarm becomes chattering, if it is triggered more than three times in a time interval of one minute. It is reported that chattering alarms commonly exist in many modern industrial facilities. For instance, the number of chattering alarms takes up 10– 60% of all alarm occurrences [75], and may even be more than 70% as mentioned in [38]. As another type of nuisance alarms, fleeting alarms are analogous to chattering alarms but do not necessarily repeat within a short time period. In addition, standing alarms refer to alarms that stay in active alarm states for a long time period, and thus cause distractions to operators as no new information indicating process abnormalities is presented by such alarms. As for correlated alarms, they are treated as nuisance alarms based on the concept that each abnormality should be indicated by only one alarm.

In addition, alarm floods could lead to severe reduction of alarm system functionalities, and even tally loose such functionalities. Two major factors are counted for alarm floods. First, modern industrial facilities are usually in large scales, involving many equipment connected with information flow and material flow paths. Second, with the advancement in alarm configurations, many process variables are monitored with different alarm signals. Consequently, when a fault occurs in the process, it is likely that a huge number of alarms would be triggered due to the propagation of abnormality along the information flow and material flow channels. Therefore, alarm floods are considered as the most difficult situations for alarm management. As mentioned in industrial standards [32, 49], an alarm flood is a situation that the number of annunciated alarms is more than what an operator can effectively manage. A more specific definition is [49]: An alarm flood is a situation that starts when the alarm rate is more than 10 alarms in 10 minutes and lasts until it drops below 5 alarms in 10 minutes. During alarm floods, operators are overwhelmed by a large number of alarms and may fail to respond to critical alarms; and thus the risks of having accidents in industrial facilities are significantly increased. According to a survey in [82], alarm floods were identified as the major culprits of many industrial accidents, such as the Three Mile Island accident, Bhopal gas tragedy, Texaco Refinery explosion, and Channel Tunnel fire. Such accidents ended up with disastrous outcomes and caused huge losses in economy, environment, reputation, and even human life.

Motivated by the above practical problems for alarm management, this thesis proposes several data-driven methods to reduce nuisance alarms and mitigate alarm floods based on industrial process data and alarm data.

1.2 Literature Survey

To improve the efficiency of alarm monitoring, signal processing and data mining technologies have been commonly exploited to detect nuisance alarms, restrain alarm overloading, and cope with alarm floods [34, 96]. This section presents a detailed literature survey on the recent development of such methods.

1.2.1 Methods to Reduce Nuisance Alarms

Aiming at reducing nuisance alarms and achieving better alarm monitoring, many effective methods have been proposed. Based on their objectives, these methods are classified into three major categories, including the rationalization of alarm systems, the implementation of basic reduction techniques, and the utilization of advanced alarm monitoring methods.

First, nuisance alarms are identified based on historical data, where chattering alarms were reported to take up a huge portion of nuisance alarms [38, 75]. To facilitate the identification of chattering alarms based on the calculation of alarm run length, chattering indexes were proposed in [56, 70]. In addition, the detection of chattering alarms was further extended to online cases with the advancement in distinguishing chattering alarms caused by process oscillations [93, 94]. Thereafter, alarm systems were rationalized by giving better design of alarm limits, which are critical parameters for industrial alarm systems. To help evaluate the performance of alarm systems, False Alarm Rate (FAR), Missed Alarm Rate (MAR), and Averaged Alarm Delay (AAD) were proposed as evaluation indexes [103, 105]. Specifically, the false alarm rate and missed alarm rate are indications of alarm system accuracy in detecting abnormalities, whereas averaged alarm delay measures the sensitivity, namely, the time difference between the occurrence of an abnormality and the instant when it is detected. To achieve improved alarm performance, many effective approaches were proposed for the design of alarm limits, such as evidence theory [104], Markov chain [112], multidimensional kernel density estimation [115], and alarm probability plot [112].

Second, many basic but effective techniques were adapted to reduce nuisance alarms in complex industrial facilities, and such techniques include delay timers, filters, deadbands [32, 46, 49]. Based on the alarm performance indexes, namely, FAR, MAR, and AAD, the design of filters in various forms was studied, such as rank order filters [86], median filters [85], and filters in generalized configurations [20]. Similarly, many strategies were developed to give optimal design of delay timers [116] and deadbands [2] to effectively minimize the occurrences of chattering alarms. For demonstration of effectiveness, delay timers and deadbands were applied to a nonlinear chemical process [87, 88]. Moreover, to help select suitable alarm reduction techniques, a decision index was proposed to determine the suitability of using deadbands [99]. It is worth mentioning that the above mentioned techniques to reduce chattering alarms are naturally applicable to mitigated fleeting alarms, which are essentially a special case of chattering alarms, namely, not necessarily with repetitions within a short time period.

Last, various advanced methods were exploited aiming at the discovery of correlated alarms, the suppression of alarms based on system dynamics and operation mode changes, and the analysis of historical data. For the discovery of correlated alarms and consequential alarms, similarity coefficients were proposed [109, 110]. To improve identification of correlated alarms and events, operator actions were incorporated [71]. Thereafter, the design of alarm limits was tackled with the incorporation of correlated alarms [36, 102], such that the alarm signals were generated based on mutual information from multiple process variables. Moreover, dynamic alarming, state-based alarming, and alarm shelving mechanisms were reported as effective approaches to cope with nuisance alarms caused by system dynamics and operation mode changes [50]. Dynamic alarm limits were exploited to mitigate alarm overloading during process transitions [118]. The design of delay-timers was generalized to cope with system dynamics [1, 53] and multiple operation modes [3, 84]. Alarm latches [4] and alarm shelving mechanisms [64] were adapted with considerations of system dynamics and operation mode changes. An alarm monitoring strategy based on hidden Markov models was proposed to handle the multimodality of process data and capture the restrictions for mode switchings [5]. Mode-based alarms were detected from historical Alarm & Event (A&E) logs by association rule mining [41]. Thereafter, mode-based alarming strategies were implemented for a hybrid process system, achieving effective alarm reduction [45]. Targeting at alarm suppression, standing alarms were suppressed by a state-based strategy [95] and false alarms were reduced by a dynamic Bayesian network [24]. In addition, to help evaluate alarm system performance and analyze historical operation data, many advanced tools were developed for distributed parallel alarm management [62], decision support with better data visualization [7, 39], and smart data analysis [42].

1.2.2 Techniques to Mitigate Alarm Floods

Addressing alarm floods is of great importance for industrial alarm systems. In the past decades, this topic has been drawing increasing attentions and a variety of methods have been proposed targeting at different directions, including the reduction of alarm rates to inhibit alarm floods, the comparison of alarm floods to facilitate analysis, the extraction of alarm flood patterns, the analysis of root causes, and the assistance of operators for enhanced situation awareness.

First, to inhibit alarm floods by reducing alarm rates, false alarms and missed alarms were minimized based on various approaches, such as the rationalization of alarm limits [63, 111], and the utilization of alarm reduction techniques, such as filters [101], deadbands [99], and delay timers [97]. Thereafter, the identification of alarm floods was studied in [98], where detection criteria were proposed to determine alarm floods for online applications and offline analysis.

Second, similarity analysis was conducted to compare historical alarm floods, which is a fundamental step towards the diagnosis and prediction of alarm floods. In [6], the similarity of alarm floods was compared based on historical alarm data. Sequence alignment was adapted to extract meaningful alarm sequential patterns as templates for fault diagnosis [19]. With the advancement to incorporate time stamps and allow gapped alignment, a modified Smith-Waterman algorithm was developed [21]. Based on improved local sequence alignment, the computation efficiency for similarity analysis was accelerated [43]. The alignment of alarm floods was also extended, such as alignment of multiple alarm flood sequences [61] and incremental calculation for online applications [59].

Third, data mining techniques were exploited to discover interesting alarm flood patterns, which are helpful for alarm suppression and decision support. For this purpose, alarm flood patterns were discovered based on itemset mining [40], sequential pattern mining [28, 72], association rules calculation [55, 89], and alarm data clustering [15, 100]. Plant hierarchy was integrated for pattern extraction [54]. Moreover, to determine alarms associated with certain root causes, correlated alarms [91] and mode-dependent alarms [41] were identified. Statistical comparisons of alarms and operator responses were conducted to identify potential solutions from historical operation strategies [73, 113]. Databases comprised of alarm flood patterns were prepared for online diagnosis of alarm floods using weighted dissimilarity index [18] and alarm coactivations [66].

Fourth, the analysis of root causes provides promising guidelines to improve alarm systems with discovery of potential solutions. For this purpose, causal relationships of process variables were captured using various causality inference methods, such as transfer entropy [69], direct causality [30], Granger causality [92], and transfer zero-entropy [31]. To improve detection accuracy and enhance applicability to practical situations, problems for the fusion of process data, alarm data, and plant connectivity were studied [71, 78, 108, 110]. For instance, pseudo-continuous time series were generated from binary alarm data to capture the causalities of alarm variables [44, 109]. Moreover, the localization of root causes was tackled with the identification of abnormality propagation paths [8, 9]. To determine alarms associated with process oscillations, methods were proposed based on the adjacency matrix [51] and comprehensive analysis of process data in temporal and spectral domains [29].

Last, operator assistance was achieved from different aspects, including the prediction, classification, diagnosis, and suppression of alarms/alarm floods. To achieve alarm prediction and then provide early warning, deep learning [14, 27], Bayesian estimators [107], alarm grouping strategies based on plant connectivity [78], and Shewhart control chart [47] were exploited. For the classification and diagnosis of alarm floods, incremental sequence alignment [60], alarm range normalization [67], similarity calculation [18], exponentially attenuated component analysis [80], alarm coactivations [65] and time series analysis [66] were adapted. Moreover, alarms were suppressed using a multi-temporal sequence mining-based algorithm [26]. To achieve operator decision support with increased situation awareness, many approaches were proposed, such as the design of improved alarm summary displays [13, 58, 77], utilization of integrated alarm monitoring frameworks [35], and fusion of historical data [106].

1.2.3 Applications to Industrial Facilities

In practice, many effective methods have been implemented in complex industrial facilities, leading to satisfactory results for alarm system improvement. More specifically, Bayesian filters were utilized to achieve effective alarm monitoring for electrical pumps in a thermal power plant [101]. The design of delay timers [87] and deadbands [88] were generalized with relaxed assumptions about process data distributions, so as to reduce nuisance alarms for a chemical reactor with a high degree of nonlinearity. Various alarm reduction techniques were applied to the alarm system in a thermal power plant over a long time period, where the obtained results indicated significant improvement in alarm system performance [97]. Dynamic alarm limits were designed to mitigate alarm floods during operation state transitions for a large-scale industrial process [118]. To reduce consequential alarms and standing alarms due to changes of operation modes, mode-based alarming strategies were deployed in an oil sands extraction plant [10], a power plant [96], and a chemical plant [38]. Comprehensive alarm information processing technology was adapted in a petrochemical plant [16]. For root cause analysis, an analytic model was developed for a power system [117].

1.3 Thesis Contributions

To improve process safety and achieve better alarm performance, this thesis proposes a variety of data-driven methods for alarm monitoring and alarm flood analysis, where both process data and alarm data are utilized. The major contributions in this thesis that distinguish it from other work are summarized as follows:

1. Proposed a new method for monitoring equipment start-up operations with applications to pumps in real industrial facilities. More specifically, an offline design framework is provided to detect the maximum unsuppression delay time and formulate dynamic alarm limits based on data associated with normal start-up operations; an online algorithm for alarm monitoring of equipment start-ups is designed based on the designed dynamic alarm limits and the calculation of an exact unsuppression delay time.

2. Proposed a systematic pattern matching method to compare alarm floods across different processes, such that alarm floods from different processes but associated with the same fault type are discovered and grouped, and the obtained results could be utilized to identify common root causes and give generalized solutions. To achieve this, a word processing approach is formulated to distill key words from textural alarm attributes and recon-

struct abstracted alarm descriptors, so as to generalize representations for alarms from different processes; a pattern matching approach is provided to compare alarm floods across different processes through three steps, including the unit-based extraction, the set-based pre-matching, and the sequence-based comparison.

3. Proposed a new method to extract compact alarm flood patterns from historical alarm flood sequences; the method is capable of avoiding the influences of order switchings caused by small time differences on pattern extraction, and also minimizing the redundancy of extracted alarm sequential patterns. More specifically, a closed alarm sequence mining approach is formulated to incorporate time stamps and tolerate alarm order switchings; a pattern distillation strategy is designed to merge similar alarm sequences and export more compact alarm sequential patterns, so as to cope with irrelevant alarms and different lengths of alarm flood sequences.

1.4 Thesis Outline

The remainder of the thesis is organized as follows.

In Chapter 2, a method to reduce nuisance alarms and achieve effective alarm monitoring for equipment start-ups is presented. Section 2.1 gives an overview of the research work in this chapter. Section 2.3 presents the offline design steps, including the determination of the data associated with normal equipment start-ups, the detection of the maximum delay time for alarm unsuppression, and the formulation of dynamic alarm limits. Section 2.4 presents the online monitoring methods, including the determination of alarm unsuppression and an algorithm for alarm monitoring based on designed dynamic alarm limits. An industrial case study is provided in Section 2.5 to illustrate the effectiveness of the method, followed by concluding remarks in Section 2.6.

In Chapter 3, a systematic pattern matching method is proposed to compare alarm floods across different processes, so as to facilitate the analysis of alarm floods across different processes by capturing common patterns and give general solutions. Section 3.1 gives an overview of the research work in this chapter. Section 3.2 formulates the problem of cross-process alarm flood pattern matching in detail. Section 3.3 proposes the method to generalize alarm representations from textural alarm attributes. Section 3.4 presents the pattern matching of alarm floods across different processes. An industrial case study is given in Section 3.5. Conclusions are provided in Section 3.6.

In Chapter 4, a method to extract alarm flood patterns from historical alarm flood sequences is proposed with improvement to incorporate time stamps, tolerate alarm order switchings, and distill compact results. Section 4.1 gives an overview of the research work in this chapter. Section 4.2 describes the problem of alarm flood pattern extraction. Section 4.3 proposes the detailed method for calculation. Section 4.4 gives an industrial case study. Section 4.5 concludes this chapter.

In Chapter 5, concluding remarks and some potential directions of future work are provided.

Chapter 2

Alarm Monitoring of Equipment Start-Up Operations*

2.1 Overview

In this chapter, a new method to monitor equipment start-up operations is proposed. The development of this method is motivated by the practical demand to prevent nuisance alarms during equipment start-up operations. More specifically, during equipment start-up operations, alarms configured for steady operating states fail to indicate true abnormalities and thus become nuisance. As a consequence, such nuisance alarms do not only distract plant operators, but may also cause equipment trips. In practice, alarm suppression can be used to eliminate standing alarms caused by equipment on-off switching. However, while the equipment is switched on, a question is when should a suppressed alarm be unsuppressed? Early unsuppression usually causes nuisance alarms during start-ups whereas late unsuppression may delay alarm monitoring in steady operating states. Moreover, the conventional constant limit based alarming mechanism is not able to indicate abnormalities during equipment start-ups, regardless of alarm suppression. A further related question is: how could equipment start-up operations be effectively monitored? To address the above practical problems, this study proposed a new method

*A version of this chapter has been published as: Boyuan Zhou, Wenkai Hu, and Tongwen Chen, A new method for alarm monitoring of equipment start-up operations with applications to pumps. *Industrial & Engineering Chemistry Research*, 58(26), 11251–11260, 2019.

for monitoring of equipment start-up operations with applications to pumps in real industrial facilities. The method includes two key stages, including 1) an offline design stage to detect the maximum unsuppression delay time and formulate dynamic alarm limits based on data associated with normal start-up operations, and 2) an online alarm monitoring of equipment start-ups based on the designed dynamic alarm limits and the calculation of an exact unsuppression delay time.

2.2 Alarm Monitoring for Start-Ups

2.2.1 Monitoring Requirement

In modern industrial facilities, alarm systems are deployed to monitor process operations. Alarms are presented to plant operators as audible and/or visible means to indicate abnormalities. An alarm signal is generated by comparing a process signal with its constant valued alarm limit, i.e., an alarm is annunciated whenever the value of its process variable is lower (or higher) than its low (or high) alarm limit. The signal form of a low limit alarm $a(t)$ is given by

$$a(t) = \begin{cases} 1, & \text{if } x(t) \le x_{tp}, \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$

where x_{tp} is the low limit of a process variable x.

However, the alarm generation mechanism in eqn. (2.1) works only for steady operating states, and may cause nuisance alarms or standing alarms due to switching on or off of equipment, such as pumps, motors, and valves. In practice, a state based alarm suppression strategy can be used to eliminate such nuisance alarms, i.e., an alarm is suppressed when the equipment is switched off, and unsuppressed when it is switched on. The alarm signal with the state based alarm suppression is given by

$$\bar{a}(t) = \begin{cases} 1, & \text{if } x(t) \le x_{tp} \;\&\; z(t) = 1, \\ 0, & \text{otherwise,} \end{cases} \tag{2.2}$$

where $z(t)$ the equipment state signal. $z(t) = 1 \,\&\, z(t-1) = 0$ indicates the equipment is switched on by an operator. By contrast, $z(t) = 0 \,\&\, z(t-1) = 1$ represents switching off.

Based on eqn. (2.2), the alarm is unsuppressed when the equipment is switched on. However, due to the initial low process value, the alarm is annunciated inevitably during the equipment start-up operation. The consequences are twofolds:

1. Such an alarm annunciated during the equipment start-up operation does not indicate any abnormality and thus distracts the plant operator.

2. Such an alarm may also cause an equipment trip, such as a pump trip, due to the extremely low process value.

An example of the electrical pump is given as follows to explain the occurrence of such a nuisance alarm.

Taking a Variable Frequency Drive (VFD) controlled pump as an example, the pressure difference $x(t)$ is up to the pump speed, which is manipulated by a VFD. To monitor the pump operation, a low limit alarm is configured for $x(t)$. Figure 2.1.(a) displays the measured pressure difference $x(t)$ (blue curve) with its alarm limit x_{tp} (red line). The pump was switched off and on at time instant t_a and t_0, respectively, as indicated by the state signal (yellow line). Figure 2.1.(b) displays the alarm signal generated by eqn. (2.1). It can be seen that standing alarms were present during pump off-states, namely, in the period between t_b and t_c, when $x(t)$ was below the low limit x_{tp}. Obviously, such a standing alarm would not be cleared until the pump was switched on. The solid line in Figure 2.1.(c) displays the alarm signal generated by eqn. (2.2). The alarm was suppressed during the pump off-state, leading to elimination of the standing alarm. However, there was still a short period from t_0 to t_c, when a nuisance alarm was present. It is obvious that this alarm did not indicate any abnormality, but was caused by the initial low value of $x(t)$ during the pump start-up operation.

The question is: how could such nuisance alarms during the pump start-up operation be prevented? An effective solution is to delay the alarm unsup-

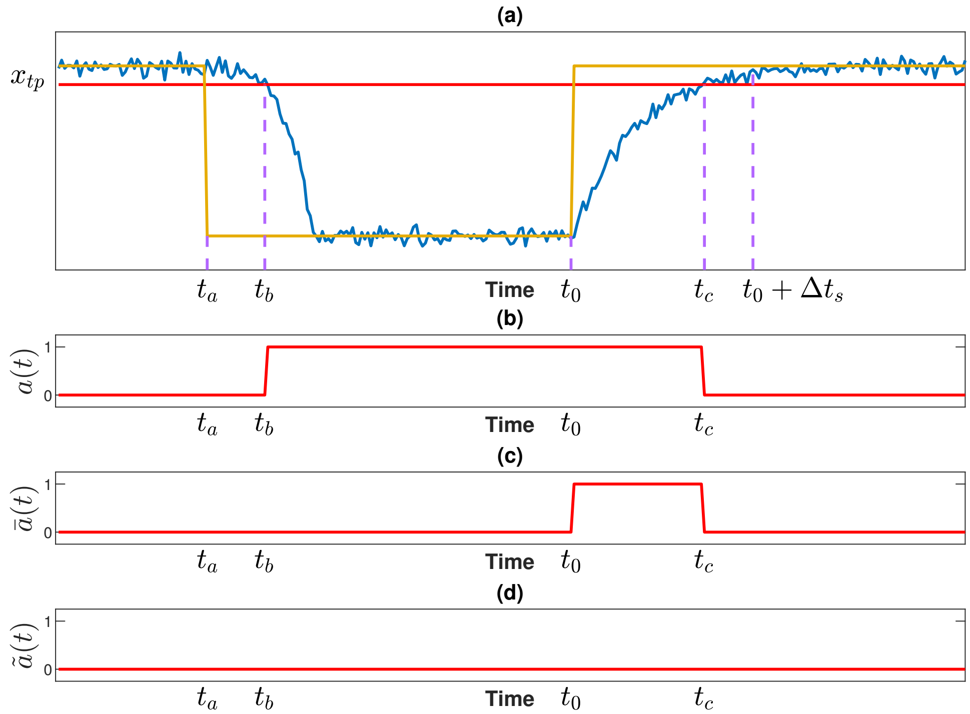

Figure 2.1: An example of alarms generated based on different alarming mechanisms for a pump pressure difference: (a) the process signal $x(t)$ (blue curve), state signal $z(t)$ (yellow line) and low alarm limit x_{tp} (red line), (b) the alarm signal $a(t)$ based on eqn. (2.1), (c) the alarm signal $\bar{a}(t)$ eqn. (2.2), as well as (d) the alarm signal $\tilde{a}(t)$ based on eqn. (2.3).

pression until the pump reaches a steady state. Applying a delay time to the alarm unsupression, the alarm signal is generated by

$$\tilde{a}(t) = \begin{cases} 1, & \text{if } x(t) \leq x_{tp} \text{ \& } z(t) = 1 \text{ \& } t \geq t_0 + \Delta t_s, \\ 0, & \text{otherwise,} \end{cases} \tag{2.3}$$

where t_0 is the time instant when the equipment is switched on, and Δt_s is the unsuppression delay time, which is formally defined as follows.

Definition 1. An unsuppression delay time is a time duration from the time instant when the equipment is switched on (the alarm is supposed to be unsuppressed right away as presented in eqn. (2.2)) to the time instant when the alarm is actually unsuppressed.

Then, the alarm configured for steady state will be kept suppressed until a delay time Δt_s is reached after t_0. Figure 2.1.(d) displays the alarm signal generated based on eqn. (2.3) with a proper delay time Δt_s applied. As a result, the nuisance alarm during the pump start-up operation was eliminated. Further, what is a proper delay time to unsuppress an alarm during equipment start-up operations? Due to different operating states or set points, the time for a process to reach a steady state from the initial start-up could be different. Thus, a fixed delay time Δt_s might not be applicable to all scenarios. To design the delay time Δt_s, the following two requirements should be considered:

1. The delay time Δt_s should be set within a proper range. A short delay time is not able to prevent nuisance alarms during equipment start-up operations, whereas a long one may lead to long detection delay of true abnormalities.

2. The delay time Δt_s should be accommodated to different operating states or set points during equipment start-up operations.

A further question is: now that the alarm configured for steady operating states is not applicable and thus suppressed during equipment start-ups, how could abnormal start-up operations be effectively monitored? A potential solution is to design an additional alarm working only during equipment start-up operations to indicate abnormal start-up situations. Eventually, the equipment start-up operation can be monitored using such a designed alarm until the delay time is reached, while the steady operation is still monitored by the original alarm with a constant alarm limit. As a result, the nuisance alarms are eliminated and the equipment start-up can be effectively monitored. Therefore, this work is going to address two problems: 1) to determine when to unsuppress an alarm during equipment start-up operations, and 2) to design a new alarm for effective monitoring of start-up operations.

2.2.2 Framework of the Method

This subsection presents a framework of the proposed method, which consists of two key stages, including an offline design stage and an online monitoring stage. In the offline design stage, there are four main steps:

1. Process datasets associated with equipment start-ups over different periods are collected. Each dataset is indexed by $s \in S, s = 1, 2, \cdots, N$, where S indicates the index set of the data collection and N is the number of collected datasets. Each individual dataset includes time series of m process signals $x_i(t)$ $(i = 1, 2, \cdots, m)$ related to the equipment operation.

2. Datasets associated with normal start-up operations are selected. A method based on the Local Outlier Probability (LoOP) algorithm and the Kernel Density Estimation (KDE) is presented to determine the index set $\tilde{S} \subseteq S$ of such normal datasets.

3. Given the normal datasets indexed by $s \in \tilde{S}$, the maximum delay time $\Delta\tilde{t}$ to unsuppress an alarm is designed based on the Principal Component Analysis (PCA).

4. The upper and lower bounds of the monitored process signal $x_i(t)$ are captured based on the normal data, and formulated as the dynamic alarm limits $\overline{f}_i(t)$ and $\underline{f}_i(t)$.

In the online monitoring stage, there are two main steps:

1. The alarm unsuppression is determined online. The exact delay time to unsuppress an alarm is calculated as $\Delta t_s = \min(\Delta\tilde{t}_s, \Delta\check{t}_s)$, where $\Delta\tilde{t}_s$ is the maximum delay time designed in the offline stage, and $\Delta\check{t}_s$ is the delay time calculated in an online manner based on a moving window PCA method.

2. A systematic online algorithm is developed to monitor the start-up operation based on the designed dynamic alarm limits until the delay time is reached, namely, $t = t_0 + \Delta t_s$, where t_0 is the switching on time instant of the equipment.

It should be noticed that the method is designed for single equipment, such as pumps, motors, fans, and valves, and may not work for multiple equipment assets or a unit. This restriction should be considered in real applications. Details of the offline design and the online monitoring methods are presented in the Offline Design section and the Online Monitoring for Equipment Start-Ups section, respectively.

2.3 Offline Design

This section presents the offline design methods, including the determination of the data associated with normal start-up operations, the detection of the maximum delay time for alarm unsuppression, and the formulation of dynamic alarm limits.

2.3.1 Determination of Normal Data

Process datasets associated with equipment start-ups over different periods are collected. Each dataset has m process variables $x_i, i = 1, 2, \ldots, m$, related to the equipment operation and contains enough samples to cover the whole start-up operation. The datasets are aligned to have the same switching on sample instant and trimmed to contain the same number of samples. Denote the numeric index of each dataset as s and the index set as $S = \{s | s = 1, 2, \cdots, N\}$, where N is the total number of datasets. Given a monitored process variable x_i, its corresponding alarm variable is denoted by a_i. Assuming that x_i has an increasing trend during the start-up operation, a_i is therefore a low limit alarm.

A practical observation of historical data is that the time series during abnormal start-up operations may have very different trends deviating away from the cluster of normal ones. As a result, such time series can be treated as abnormal data and discarded. It should be noticed that only normal equipment start-up data is needed to calculate unsuppression delay time and capture dynamic alarm limits. If abnormal data is used, the obtained dynamic alarm limits will contain the trajectories of abnormal start-ups. As a result, based on such limits, it may fail to detect abnormalities in start-up operations. The determination of the abnormal data is based on four steps:

1. Assign a score P_{is} for variable x_i in the sub-dataset s, to evaluate the probability of being abnormal. The Local Outlier Probability (LoOP) algorithm is utilized.

2. Estimate the Probability Density Function (PDF) $f_i(p)$ of the score P_{is} by Kernel Density Estimation (KDE). Given a significance level α_0, the index set Ω_i for x_i being taken as normal is calculated based on $f_i(p)$. Any sub-dataset s with x_i not in Ω_i is abnormal. The index dataset of abnormal start-ups is $S_i = \{s|s \notin \Omega_i\}$.

3. Repeat Steps 1) and 2) to obtain the index set of abnormal data for all m process variables as $S_1, S_2, \cdots, S_m$.

4. Obtain the index set $\tilde{S}$ of normal data by excluding the index set of all abnormal ones as $\tilde{S} = S \setminus \{S_1 \cup S_2 \cup \cdots \cup S_m\}$.

The detailed calculations are as follows.

For process variable x_i, combining all its time series from N datasets forms a data matrix $\mathbf{X}_i \in \mathbb{R}^{M \times N}$

$$\mathbf{X}_i = [X_{i1}, X_{i2}, \ldots, X_{iN}], \tag{2.4}$$

where X_{is} is the time series of x_i in sub-dataset s, and M is number of samples. Each time series X_{is} is denoted by

$$X_{is} = [x_{is}(t_0), x_{is}(t_1), \ldots, x_{is}(t_{M-1})]^T, \tag{2.5}$$

where $x_{is}(t)$ is a sample of x_i in the sub-dataset s at time instant t, and t_0 denotes the switching on time instant of the equipment.

The index set S_i that indicates abnormal start-up operations is obtained as follows. A score P_{is} for the time series X_{is} is calculated based on the Local

Outlier Probability (LoOP) algorithm [57], to evaluate the probability of X_{is} being an outlier time series associated with an abnormal start-up operation. Given the data matrix $\mathbf{X}_i \in \mathbb{R}^{M\times N}$ for x_i, a distance metric between X_{is} and its k-nearest neighbours is given by

$$\zeta(X_{is}, \mathcal{N}(X_{is})) = \sqrt{\frac{\sum_{X_{ik}\in\mathcal{N}(X_{is})} d^2(X_{is}, X_{ik})}{|\mathcal{N}(X_{is})|}}, \tag{2.6}$$

where $\mathcal{N}(X_{is})$ is a set containing the k-nearest neighbours of X_{is}; it can be obtained by ranking the distances $d(X_{is}, X_{ik})$ $(s \neq k)$ and choosing the X_{ik}'s corresponding to the k nearest ones. The Manhattan distance $d(X_{is}, X_{ik})$ is used, i.e.,

$$d(X_{is}, X_{ik}) = \sum_{t=t_0}^{t_{M-1}} |x_{is}(t) - x_{ik}(t)|, \tag{2.7}$$

where $|\cdot|$ denotes the absolute value.

A probabilistic set distance for X_{is} and $\mathcal{N}(X_{is})$ with a significance level λ is

$$D(\lambda, X_{is}, \mathcal{N}(X_{is})) = \lambda\zeta(X_{is}, \mathcal{N}(X_{is})). \tag{2.8}$$

The Probabilistic Local Outlier Factor (PLOF) for $X_{is} \in \mathbf{X}_i$ with respect to a significance level λ is

$$\psi_{is} = \frac{|\mathcal{N}(X_{is})| \cdot D(\lambda, X_{is}, \mathcal{N}(X_{is}))}{\sum_{X_{ik}\in\mathcal{N}(X_{is})}[D(\lambda, X_{ik}, \mathcal{N}(X_{ik}))]} - 1, \tag{2.9}$$

where $\mathcal{N}(X_{is})$ and $\mathcal{N}(X_{ik})$ are the sets of k-nearest neighbours of X_{is} and X_{ik}, respectively. The aggregate value for all time series of x_i is

$$\Psi_i = \lambda\sqrt{\frac{1}{N}\sum_{s=1}^{N}\psi_{is}^2}. \tag{2.10}$$

The Local Outlier Probability (LoOP) value is given by

$$P_{is} = \max\left(0, \mathrm{erf}(\frac{\psi_{is}}{\sqrt{2}\Psi_i})\right), \tag{2.11}$$

where $\mathrm{erf}(\cdot)$ is the Gaussian error function.

The Probability Density Function (PDF) $f_i(p)$ of P_{is} is estimated by the Kernel Density Estimation (KDE) [81], i.e.,

$$f_i(p) = \frac{1}{Nh} \sum_{s=1}^{N} \phi(\frac{p - P_{is}}{h}), \tag{2.12}$$

where $\phi(\cdot)$ is the kernel function and h is the kernel bandwidth. Here, the Gaussian kernel $\phi(x)$ is used, i.e.,

$$\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}. \tag{2.13}$$

An optimal choice of h is given by [81]

$$h = \left(\frac{4\hat{\sigma}^5}{3N} \right)^{\frac{1}{5}} \approx 1.06\hat{\sigma}N^{-\frac{1}{5}}, \tag{2.14}$$

where $\hat{\sigma}$ is the standard deviation of P_{is}'s.

A decision value is defined as

$$F_i(p|s) = \int_{-\infty}^{P_{is}} f_i(p)dp. \tag{2.15}$$

The times series for x_i in sub-dataset s is determined to be normal, if $F_i(p|s) \leq 1 - \alpha_0$. Therefore, the index set of the normal time series is obtained

$$\Omega_i = \{s | F_i(p|s) \leq (1 - \alpha_0)\}, \tag{2.16}$$

where α_0 is the significance level. Then, the index set of abnormal time series is

$$S_i = \{s | s \notin \Omega_i\}. \tag{2.17}$$

Repeat the calculations from eqn. (2.4) to eqn. (2.17) for all m process variables and get their index sets of abnormal time series as $S_1, S_2, \cdots, S_m$. Then, the index set $\tilde{S}$ for all time series associated with normal start-up operation is obtained by excluding all sets with abnormal ones, i.e,

$$\tilde{S} = S \setminus \{S_1 \cup S_2 \cup \cdots \cup S_m\}, \tag{2.18}$$

The datasets in $\tilde{S}$ can be treated as data associated with normal start-ups, and they will be used for the design of the maximum delay time $\Delta\tilde{t}$ and dynamic alarm limits. To ensure robustness, sufficient data associated with normal start-ups should be collected.

Remark 1. The above method to determine normal data has two major advantages: First, since the start-up operation is non-stationary, it is hard to describe it using a certain distribution. Thus, any distribution based outlier detection method is not applicable. By contrast, the LoOP is a local density based outlier detection method, which does not assume any data distribution [57] and thus is applicable in this case. Second, this method is completely data driven and does not require any prior knowledge. Thus, it is easy for implementation and can save time and resources compared to determining the normal data manually based on the experience from plant operators or process engineers.

2.3.2 Detection of the Maximum Unsuppression Delay Time

This subsection proposes a method to find the maximum delay time for alarm unsuppression. It is based on how much time is required for the operation to reach a steady state from the switching on time instant. The Principal Component Analysis (PCA) is applied. In each dataset s from the normal data collection $\tilde{S}$, a data matrix $\mathbf{X} \in \mathbb{R}^{M \times m}$ containing m process variables is

$$\mathbf{X} = [X_1, X_2, \ldots, X_m], \tag{2.19}$$

where $X_i = [x_i(t_0), x_i(t_1), \cdots, x_i(t_{M-1})]^T$ is the time series of $x_i, i = 1, 2, \cdots, m$. The matrix $\mathbf{X}$ is normalized as

$$\tilde{\mathbf{X}} = (\mathbf{X} - \mathbf{IM})\, \mathbf{\Xi}^{-1}, \tag{2.20}$$

where $\mathbf{M} = \mathrm{diag}(\mu)$ and $\mathbf{\Xi} = \mathrm{diag}(\sigma)$ are diagonal matrices containing the mean and the standard deviation of each column of $\mathbf{X}$, respectively, i.e., $\mu = [\mu_1, \mu_2, \ldots, \mu_m]$ and $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_m]$. The symbol $\mathbf{I}$ represents an all one matrix of size $M \times m$.

The matrix $\tilde{\mathbf{X}}$ is decomposed by the Principal Component Analysis (PCA) [52] as

$$\tilde{\mathbf{X}} = \mathbf{TP}^T + \mathbf{E}, \tag{2.21}$$

where $\mathbf{T} \in \mathbb{R}^{M\times l}$ is the score matrix, $\mathbf{P} \in \mathbb{R}^{m\times l}$ is the loading matrix, $\mathbf{E}$ is the residual matrix, and l is the number of principal components. The T^2 statistic is given by

$$T^2(t) = \tilde{X}(t)\mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^T\tilde{X}(t)^T, \tag{2.22}$$

where $\tilde{X}(t) \in \mathbb{R}^{1\times m}$ indicates one sample of m process variables at time instant t, and $\mathbf{\Lambda}$ is the diagonal matrix containing the l eigenvalues of the covariance matrix of $\tilde{\mathbf{X}}$. The statistical confidence limit for T^2 statistics is

$$J = \frac{l(M-1)}{M-l}F_{\alpha_1}(l, M-l), \tag{2.23}$$

where $F_{\alpha_1}(df_1, df_2)$ denotes the critical value of the F-distribution with two degrees of freedom df_1 and df_2 at the significance level α_1.

For each dataset $s \in \tilde{S}$, the delay time $\Delta\tilde{t}_s$ to unsuppress an alarm is essentially the period for the operation to reach its steady state from the switching on time instant t_0, and is formulated as

$$\Delta\tilde{t}_s = \min_{t>t_0, T^2(t)\leq J}(t - t_0). \tag{2.24}$$

Then, the maximum delay time to unsuppress an alarm during equipment start-up operations is

$$\Delta\tilde{t} = \max(\Delta\tilde{t}_s | s \in \tilde{S}). \tag{2.25}$$

This delay time $\Delta\tilde{t}$ can be used as a conservative value to determine alarm unsuppression. However, in view of some cases with faster start-ups, it is necessary to determine the alarm unsuppression in an online manner. Then, $[0, \Delta\tilde{t}\,]$ is the proper range of the delay time to unsuppress an alarm, and will be used in the online monitoring stage in the Online Monitoring for Equipment Start-Ups section to determine the exact time instant for alarm unsuppression.

Remark 2. Instead of detecting faults, the PCA method is used to identify the time instant when process variables reach steady states, based on the T^2 statistic. Process signals are stationary in both initial states (before start-up) and steady states, while they are non-stationary in start-up periods and contribute most of the variations. Such a feature makes PCA applicable to detect start-up periods. The conventional PCA works well for linear processes. However, some equipment start-up might be involved with strongly nonlinear processes. In such cases, the kernel PCA method [79] can be applied.

2.3.3 Formulation of Dynamic Alarm Limits

This subsection designs dynamic alarm limits for the monitored process variable x_i based on the normal data collection $\tilde{S}$. They are formulated based on the upper and lower bounds of time series in $s \in \tilde{S}$. The upper and lower bounds for x_i at time instant t are given by

$$\overline{x}_i(t) = \max(x_{is}(t) | s \in \tilde{S}, t_0 \leq t \leq t_0 + \Delta\tilde{t}), \tag{2.26}$$

$$\underline{x}_i(t) = \min(x_{is}(t) | s \in \tilde{S}, t_0 \leq t \leq t_0 + \Delta\tilde{t}), \tag{2.27}$$

where $x_{is}(t)$ is the time series of x_i at time instant t in the sub-dataset $s \in \tilde{S}$.

In view of that the sampling rate for online monitoring could be faster than that of the historical data, it is necessary to formulate the upper and lower bounds as models with respect to the time. Here, the cubic spline interpolation [25] is applied. Then, the upper bound $\overline{x}_i(t)$ and the lower bound $\underline{x}_i(t)$ are converted to piecewise cubic functions with respect to time. Given a time duration $t \in [t_1, t_2]$, the dynamic alarm limits are formulated as

$$\overline{f}_i(t|t_1, t_2) = \overline{B}_{i|t_1,t_2}^T T(t), \tag{2.28}$$

$$\underline{f}_i(t|t_1, t_2) = \underline{B}_{i|t_1,t_2}^T T(t), \tag{2.29}$$

where t_1 and t_2 indicate two consecutive sampling instants and $t_0 \leq t_1 < t_2 \leq t_0 + \Delta\tilde{t}$. $\overline{B}_{i|t_1,t_2}$ and $\underline{B}_{i|t_1,t_2}$ are the coefficient vectors of $\overline{f}_i(t|t_1, t_2)$ and $\underline{f}_i(t|t_1, t_2)$, respectively. $T(t) = [(t-t_1)^3, (t-t_1)^2, (t-t_1), 1]^T$. The models in Eq . (2.28) and eqn. (2.29) represent the boundaries of the time series for x_i during normal operations, and thus can be used as dynamic alarm limits to monitor the equipment start-up operation.

2.4 Online Monitoring for Equipment Start-Ups

This section presents the online monitoring method, including the determination of alarm unsuppression for real-time applications, and the algorithm of alarm monitoring based on the designed dynamic alarm limits.

2.4.1 Determination of Alarm Unsuppression

Due to the change of operating conditions or set points, the process may take different time to reach a steady state. Thus, a fixed delay time for alarm unsuppression would not be applicable for all cases. It is necessary to determine alarm unsuppression in an online manner. A moving window PCA method [74] is used to calculate the online delay time $\Delta\check{t}_s$, i.e., to determine when the alarm should be exactly unsuppressed. Such a delay time $\Delta\check{t}_s$ must be smaller than $\Delta\tilde{t}$, so as to avoid long detection delay of true abnormalities during steady operations.

Denote the data matrix consisting of m process variables from time instant t_0 to t as

$$\mathbf{X}(t) = \begin{bmatrix} x_1(t_0) & x_2(t_0) & \dots & x_m(t_0) \\ \vdots & \vdots & & \vdots \\ x_1(t) & x_2(t) & \dots & x_m(t) \end{bmatrix}, \tag{2.30}$$

where t_0 indicates the time instant when the equipment is switched on and t is the current time instant. The moving window PCA gets the T^2 statistics based on the current matrix $\mathbf{X}(t)$ and the PCA model is trained from the matrix $\mathbf{X}(t)$ over a time window of w samples at L samples ago, i.e.,

$$\mathbf{X}_w = \mathbf{X}(t - L - w + 1 : t - L), \tag{2.31}$$

where $\mathbf{X}(t_a : t_b)$ denotes the matrix containing the samples from time instant t_a to t_b ($t_0 \leq t_a \leq t_b$) for all m process variables. The matrix $\mathbf{X}_w$ is normalized as

$$\tilde{\mathbf{X}}_w = (\mathbf{X}_w - \mathbf{I}_w \mathbf{M}_w)\, \mathbf{\Xi}_w^{-1}, \tag{2.32}$$

where $\mathbf{M}_w = \text{diag}(\mu_w) \in \mathbb{R}^{m\times m}$ and $\mathbf{\Xi}_w = \text{diag}(\sigma_w) \in \mathbb{R}^{m\times m}$ are diagonal matrices containing the mean $\mu_w = [\mu_{w1}, \mu_{w2}, \dots, \mu_{wm}]$ and the standard deviation $\sigma_w = [\sigma_{w1}, \sigma_{w2}, \dots, \sigma_{wm}]$ of each column of $\mathbf{X}_w$, respectively. The symbol $\mathbf{I}_w$ represents an all one matrix of size $w \times m$.

The rest calculations are analogous to the convectional PCA in the Offline Design section. The matrix $\tilde{\mathbf{X}}_w$ is decomposed to be

$$\tilde{\mathbf{X}}_w = \mathbf{T}_w \mathbf{P}_w^T + \mathbf{E}_w. \tag{2.33}$$

The T_w^2 statistics at time instant t is

$$T_w^2(t) = \tilde{X}(t) \mathbf{P}_w \mathbf{\Lambda}_w^{-1} \mathbf{P}_w^T \tilde{X}(t)^T, \tag{2.34}$$

where $\tilde{X}(t) \in \mathbb{R}^{1\times m}$ indicates one sample of m process variables at time instant t, and $\mathbf{\Lambda}_w$ is the diagonal matrix containing the l_w eigenvalues of the covariance matrix of $\tilde{\mathbf{X}}_w$. The statistical confidence limit for the online calculated T_w^2 statistic is

$$J_w = \frac{l_w(w-1)}{w-l_w} F_{\alpha_2}(l_w, w-l_w). \tag{2.35}$$

The operation is said to reach its steady state if $T_w^2(t) \leq J_w$. The period for the operation to reach its steady state from the switching on time instant t_0 is

$$\Delta \check{t}_s = \min_{t>t_0, T_w^2(t) \leq J_w} (t - t_0). \tag{2.36}$$

The exact delay time for alarm unsuppression is eventually taken as

$$\Delta t_s = \min(\Delta \tilde{t}_s, \Delta \check{t}_s). \tag{2.37}$$

The alarm is not unsuppressed until Δt_s is reached after t_0. Accordingly, it guarantees the delay time for alarm unsuppression as short as possible to avoid long detection delay of true abnormalities during steady operating states.

Remark 3. The delay time calculated in the offline stage is called the maximum unsupressed delay time $\Delta \tilde{t}_s$, which is essentially the longest time for the process variable to reach its steady state in normal start-ups. It can be used as a conservative option of unsuppression delay time. Due to different operating conditions or set points, the start-up operation may take different time to reach its steady state. To cope with this, the unsupressed delay time $\Delta\check{t}_s$ is calculated online. Then, the exact time delay Δt_s of a new start-up is determined by eqn. (2.37). This can avoid long detection delay of true abnormalities in steady states.

2.4.2 Alarm Monitoring Based on the Dynamic Alarm Limits

To monitor the start-up operations based on the designed dynamic alarm limits and the calculation of unsuppression delay time, an online alarm monitoring algorithm is developed in this subsection. A new alarm $\tilde{a}_i^*$ is designed for the monitored process variable x_i; this alarm works only during the period from the switching on time instant t_0 to the delayed unsuppression time instant $t_0 + \Delta t_s$. Meanwhile, the original alarm $\tilde{a}_i$ configured for steady operating states is suppressed during this period. After $t_0 + \Delta t_s$, the alarm $\tilde{a}_i$ is unsuppressed to work normally in the steady state, while the designed alarm $\tilde{a}_i^*$ is then suppressed at the same time. The signal form of the designed new alarm $\tilde{a}_i^*$ is given by

$$\tilde{a}_i^*(t) = \begin{cases} 0, & \text{if } \underline{f}_i(t) \leq x_i(t) \leq \overline{f}_i(t), \\ 1, & \text{otherwise.} \end{cases} \tag{2.38}$$

where $t_0 \leq t \leq t_0 + \Delta t_s$. Whenever there is an abnormality during the start-up operation, this new alarm will be annunciated to notify the operator.

The alarm monitoring for equipment start-up operations is formulated as Algorithm 2. The inputs include the maximum delay time $\Delta\tilde{t}$ for alarm unsuppression, the dynamic alarm limits, and the online updated data matrix $\mathbf{X}(t)$. The outputs are the alarm signals $\tilde{a}_i^*(t)$ and $\tilde{a}_i(t)$. The variable Θ is an indicator for the switching of alarm monitoring strategies and initially set to 0. The moving window PCA model is recursively calculated to obtain the delay time $\Delta\check{t}_s$ in lines $8 - 16$. The equipment start-up operation is monitored by $\tilde{a}_i^*(t)$ in lines $21 - 25$. The exact delay time Δt_s to unsuppress $\tilde{a}_i$ is

Algorithm 1 Online alarm monitoring for equipment start-up operations.

Input Arguments: $\Delta\tilde{t}$, $\overline{f}_i(t)$, $\underline{f}_i(t)$, $\mathbf{X}(t)$.
Output Arguments: $\tilde{a}_i^*(t)$, $\tilde{a}_i(t)$.
Set $\Theta = 0$;
Set $\Delta\check{t}_s = \infty$;
while $z(t) = 1$ **do**
    if $\Theta = 0$ **then**
        $\tilde{a}_i(t) = 0$;
        $\mathbf{X}_w = \mathbf{X}(t - L - w + 1 : t - L)$;
        $\tilde{\mathbf{X}}_w = (\mathbf{X}_w - \mathbf{1}\mathbf{M}_w)\,\mathbf{\Xi}_w^{-1}$;
        $\tilde{\mathbf{X}}_w = \mathbf{T}_w\mathbf{P}_w^T + \mathbf{E}_w$;
        $\tilde{X}(t) = (X(t) - \mu_w)\mathbf{\Xi}_w^{-1}$;
        $T_w^2(t) = \tilde{X}(t)\mathbf{P}_w\Lambda_w^{-1}\mathbf{P}_w^T\tilde{X}(t)^T$;
        $J_w = \frac{l_w(w-1)}{w-l_w}F_{\alpha_2}(l_w, w - l_w)$;
        if $T_w^2(t) \le J_w$ **then**
            $\Delta\check{t}_s = \min_{t>t_0, T_w^2(t)\le J_w}(t - t_0)$;
        end if
        $\Delta t_s = \min(\Delta\tilde{t}_s, \Delta\check{t}_s)$;
        if $t \ge t_0 + \Delta t_s$ **then**
            Set $\Theta = 1$;
        end if
        if $\underline{f}_i(t) \le x_i(t) \le \overline{f}_i(t)$ **then**
            $\tilde{a}_i^*(t) = 0$;
        else
            $\tilde{a}_i^*(t) = 1$;
        end if
        Update $\mathbf{X}(t)$;
    else
        $\tilde{a}_i^*(t) = 0$;
        if $x_i(t) \le x_{tp}$ **then**
            $\tilde{a}_i(t) = 1$;
        else
            $\tilde{a}_i(t) = 0$;
        end if
    end if
end while

determined in line 17. It should be noted that $\Delta\check{t}_s$ is initially set to ∞, such that the exact delay time Δt_s is no more than $\Delta\tilde{t}_s$. When the delay time is reached, i.e., $t \geq t_0 + \Delta t_s$, the alarm monitoring is switched from $\tilde{a}_i^*(t)$ to $\tilde{a}_i(t)$ by setting $\Theta = 1$ as in lines $18-20$. Thereafter, the alarm $\tilde{a}_i^*$ is suppressed and the calculation of the moving window PCA model stops, while the alarm $\tilde{a}_i$ is unsuppressed to continue monitoring the equipment operation in its steady state as presented in lines $28 - 33$.

The obtained dynamic alarm limits in eqn. (2.28) and (2.29) are used to detect abnormalities in equipment start-ups. Without such dynamic limits, it is impossible to indicate true abnormalities in start-ups, since constant based alarms only work for steady states and become useless in the start-up operation. The dynamic thresholds are captured offline and used online. The major calculation is to obtain a dynamic threshold from a large amount of historical data and this is done in the offline design stage. Thus, the computational complexity is not a problem. As for the online monitoring stage, there is not much calculation involved since the calculation is just to compare the current process value with the obtained dynamic limits.

2.5 Industrial Case Study

This section presents an industrial case study involving real process data to demonstrate the effectiveness of the proposed method. The studied equipment is a booster pump, which transfers liquids from an upstream process to a downstream one and increases the pressure to meet supply requirements. Based on the historical data, the offline design results are presented first, followed by two cases of online monitoring. Comparisons of different alarm monitoring strategies are given through more online monitoring scenarios.

Offline Design

At first, $N = 30$ datasets associated with pump start-up operations over different periods were collected. The sampling period was 1 sec. These

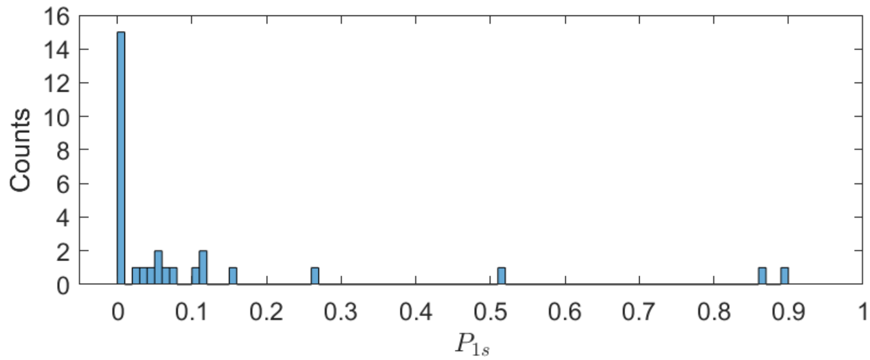

Figure 2.2: Histogram of scores P_{1s}'s for the pressure difference variable x_1.

datasets were aligned to have the same switching on time instant at $t_0 = 61$ sec, and trimmed to contain the same number of samples. The variables related to the pump operation include the pressure difference, pump speed and pump motor current denoted by x_1, x_2 and x_3, respectively. The pressure difference x_1 indicates the difference between the discharge pressure and the suction pressure. It is the monitored process variable with a low limit alarm configured. In the offline stage, the maximum alarm unsuppression delay and the dynamic alarm limits for x_1 during pump start-up operations are determined.

The index set $\tilde{S}$ of datasets associated with abnormal pump start-up operations are obtained using the method in the Determination of Normal Data subsection. Taking the pressure difference x_1 for example, the scores P_{1s}'s are calculated from eqn. (2.11), and the histogram of P_{1s}'s is shown in Figure 2.2. The PDF $f_1(p)$ is estimated using eqn. (2.12), and the index set S_1 of abnormal start-ups of process variable x_1 is calculated from eqn. (2.17). The determination of abnormal data is shown in Figure 2.3, where the bold dashed curves are abnormal time series of x_1, namely $s \in S_1$. Analogously, the index sets S_2 and S_3 for process variables x_2 and x_3 are obtained. Therefore, the index set for normal pump start-ups is $\tilde{S} = S \setminus \{S_1 \cup S_2 \cup S_3\}$. The column $\in \tilde{S}$ of Table 2.1 tells whether the dataset s is associated with normal (1) or abnormal (0) pump start-up operations.

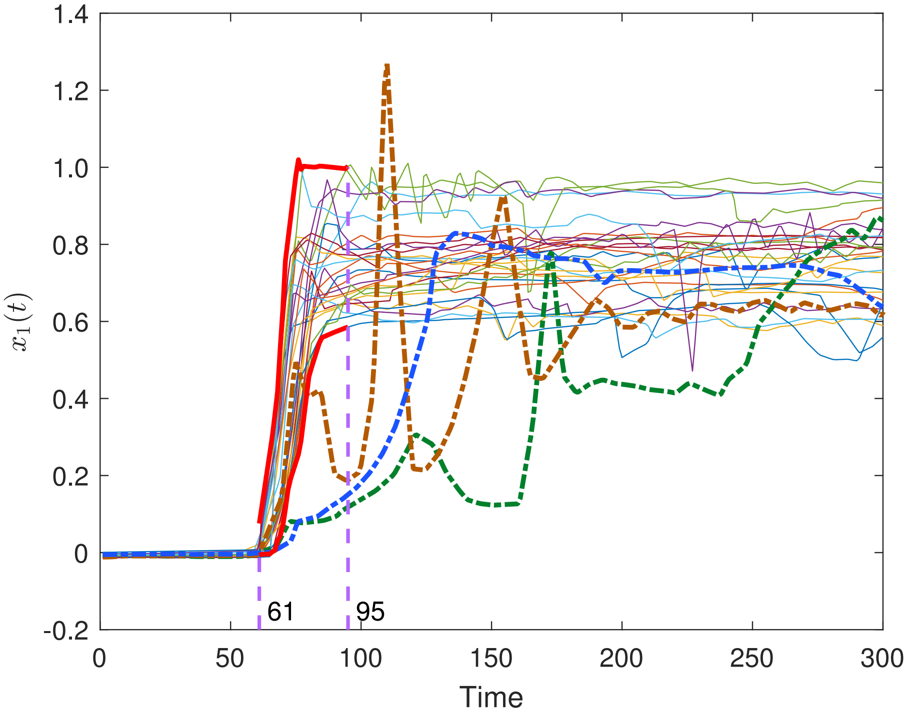

Figure 2.3: The historical time series of $x_1(t)$. The normal and abnormal ones are indicated by solid thin curves and bold dashed curves. The designed dynamic alarm limits are captured and represented by two bold red curves.

Based on $\tilde{S}$, the maximum alarm unsuppression delay $\Delta\tilde{t}$ and the dynamic alarm limits are designed following the procedures in the Detection of the Maximum Unsuppression Delay Time subsection and the Formulation of Dynamic Alarm Limits subsection, respectively. The time duration to reach steady states from the initial start-ups are shown in the column $\Delta\tilde{t}_s$ of Table 2.1. The maximum delay time $\Delta\tilde{t}$ for alarm unsuppression is finally obtained as $\Delta\tilde{t} = 34$ sec. The designed dynamic alarm limits of the pressure difference x_1 are captured and shown as two bold red curves in Figure 2.3. Based on $\Delta\tilde{t}$ and the dynamic alarm limits, the pump operation is monitored in an online manner following Algorithm 2. Two cases including both normal and abnormal pump start-up operations are given as follows.

Table 2.1: Determination of normal data and detection of time to reach steady states from initial start-ups.

Index s	$\in \tilde{S}$	$\Delta\tilde{t}_s$	Index s	$\in \tilde{S}$	$\Delta\tilde{t}_s$
1	1	21 sec	16	1	31 sec
2	1	34 sec	17	1	17 sec
3	1	32 sec	18	1	11 sec
4	1	33 sec	19	1	26 sec
5	0	/	20	1	17 sec
6	1	11 sec	21	0	/
7	1	13 sec	22	1	17 sec
8	1	27 sec	23	1	11 sec
9	1	12 sec	24	1	26 sec
10	1	20 sec	25	1	17 sec
11	1	26 sec	26	1	30 sec
12	1	11 sec	27	0	/
13	1	27 sec	28	1	12 sec
14	1	26 sec	29	1	16 sec
15	1	15 sec	30	1	26 sec

Online Monitoring of a Normal Pump Start-Up Case

The data associated with a normal pump start-up operation was collected to validate the proposed online monitoring method. The sampling period was 1 sec. The pressure difference signal $x_1(t)$ and its alarm signals are shown in Figure 2.4. It can be found that the online calculated delay time $\Delta\check{t}$ was shorter than the maximum delay time $\Delta\tilde{t}$. Thus, the original alarm $\tilde{a}_1$ was kept suppressed until $\Delta t_s = \min(\Delta\tilde{t}_s, \Delta\check{t}_s) = 28$ sec later. Meanwhile, the designed new alarm $\tilde{a}_1^*$ was used to monitor the start-up operation by comparing $x_1(t)$ with the dynamic alarm limits (two red curves in Figure 2.4(a)) captured from normal data in the offline stage. The pressure difference did not exceed the dynamic limits, and thus there was no alarm annunciated during the start-up operation from $t_0 = 61$ sec to $t_0 + \Delta t_s = 79$ sec. Starting from $t_0 + \Delta t_s = 79$ sec, the alarm $\tilde{a}_1^*$ was suppressed and the original alarm $\tilde{a}_1$ was unsuppressed. The alarm monitoring strategy was switched to $\tilde{a}_1$ to continue monitoring the pump operation in its steady state. It can be seen that the nuisance alarm was successfully prevented in this case. As a comparison, the alarm signal

$\bar{a}(t)$ based on eqn. (2.2) is given in Figure 2.4(d); it can be found that there was a nuisance alarm from 61 sec to 75 sec, due to the low initial value of the pressure difference signal $x_1(t)$. Therefore, if the alarm signal $\bar{a}(t)$ is used to monitor the booster pump, operators would be distracted by such nuisance alarms, whenever it is switched on.

Online Monitoring of an Abnormal Pump Start-Up Case

The data associated with an abnormal pump start-up operation was collected. Figure 2.5 presents signals of the pressure difference $x_1(t)$, the designed new alarm $\tilde{a}_1^*$ (eqn. (2.38)), the original alarm $\tilde{a}_1$ (eqn. (2.3)), and the conventional alarm $\bar{a}_1$ (eqn. (2.2)) that has no unsuppression delay time. It can be found that the pump operation did not reach its steady state at the maximum delay time $\Delta\tilde{t} = 34$ sec from the switching on time instant $t_0 = 61$ sec. It was also detected that the pressure difference $x_1(t)$ dropped below the lower bound at the 70th time instant. The designed alarm $\tilde{a}_1^*$ was annunciated to indicate this abnormal start-up operation. Starting from $t_0 + \Delta t_s = 95$ sec, the alarm $\tilde{a}_1^*$ was suppressed and the original alarm $\tilde{a}_1$ was unsuppressed. Since the pressure difference was still lower than the constant alarm limit x_{tp}, the original alarm $\tilde{a}_1$ was annunciated to indicate the abnormality. It can be seen that an abnormal start-up operation was successfully reported in this case. However, the conventional alarm $\bar{a}_1$ was annunciated immediately after the pump was switched on at time instant $t_0 = 61$ sec. Between 61 sec and 70 sec, $\bar{a}_1$ was a nuisance alarm, because the pressure difference $x_1(t)$ was still within its normal range.

As shown in Figures 2.4 and 2.5, no matter the start-up was normal or abnormal, nuisance alarms were always triggered based on the conventional alarming mechanism $\bar{a}_1$. However, the proposed method can effectively remove such nuisance alarms, and indicate abnormalities in start-ups based on $\tilde{a}_1^*$ and $\tilde{a}_1$.

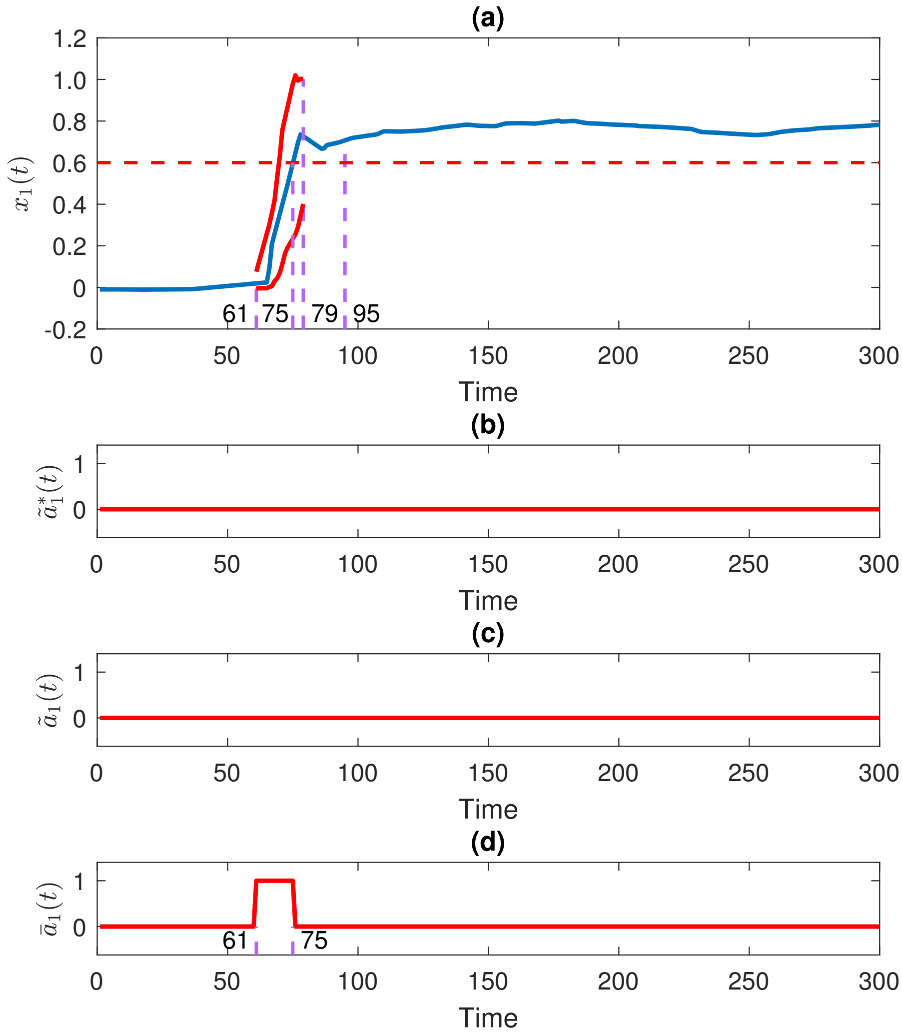

Figure 2.4: Example of a normal pump start-up operation. (a) The pressure difference signal $x_1(t)$ (blue curve), dynamic alarm limits (solid red curve), and original constant alarm limit x_{tp} (dashed red line). (b) The alarm signal $\tilde{a}_1^*(t)$ based on eqn. (2.38) for start-up operation. (c) The alarm signal $\tilde{a}_1(t)$ based on eqn. (2.3) for steady operation. (d) The alarm signal $\bar{a}(t)$ based on eqn. (2.2).

Comparison of Different Alarm Monitoring Mechanisms

The data of a longer operation period for the booster pump, was used to test and compare the three alarm monitoring methods, including the conventional method based on eqn. (2.1), the method with an alarm suppression strategy based on eqn. (2.2), and the proposed method based on Algorithm 2.

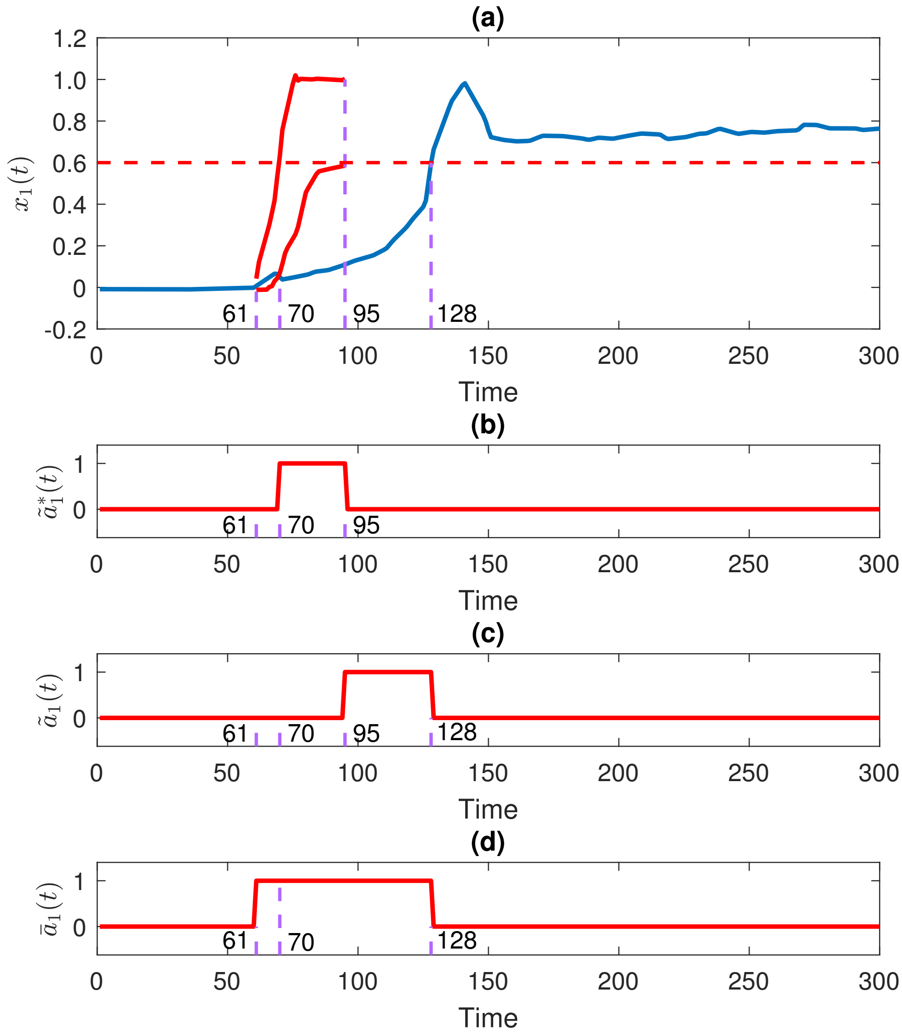

Figure 2.5: Example of an abnormal pump start-up operation.

The data included 8 normal pump start-up operations. Table 2.2 presents the alarm count and alarm duration for each method. The conventional method gave 20 alarm occurrences and long alarm durations (more than 4787 sec in total), due to the changes of pump operating states and the occurrences of real faults. Using the method with an alarm suppression strategy, all standing alarms were removed. As a result, the alarm count and alarm duration were reduced to 12 and 406 sec, respectively. Among the 12 alarm occurrences, 8 were caused by initial low values during the pump start-ups. Thus, using the proposed method based on Algorithm 2, all these 8 false alarms were removed,

Table 2.2: Performance comparison of different alarm monitoring methods.

Method	Alarm Count	Alarm Duration (sec)
Conventional method based on eqn. (2.1)	20	> 4787
Method with suppression based on eqn. (2.2)	12	406
Proposed method based on Algorithm 2	4	270

leading to a further reduction of the alarm count and alarm duration. The remaining 4 alarm occurrences were associated with 4 real faults during the pump operation. Thus, the comparison results demonstrate that the proposed method is capable of preventing nuisance alarms during equipment start-up operations.

The methods were also compared on two abnormal start-up operations. Using the proposed method based on Algorithm 2, the process signal was detected to exceed the dynamic alarm limits in a few seconds after the switching on of the booster pump. Thus, the proposed method successfully detected the abnormalities during the two start-up operations. By contrast, neither the conventional method nor the method with an alarm suppression strategy was able to indicate such abnormalities, since the alarms were active from the beginning (switching on time instant) of the pump start-up operations. Therefore, the comparison results demonstrate that the proposed method is capable of achieving effective alarm monitoring for equipment start-up operations.

The proposed method involves three parameters, namely, α_0 in eqn. (2.16), α_1 in eqn. (4.2), and α_2 in eqn. (2.35) which represent significance levels to determine abnormal data and to determine the T^2 statistic threshold. The influences of the two parameters are very limited to the final results. Thus, there is no need to tune the parameters.

2.6 Summary

The study in this chapter is motivated by a practical problem that nuisance alarms appear commonly in equipment start-up operations. To solve this problem, a new method for the alarm monitoring of equipment start-up operations is proposed. The method includes an offline design stage and an online monitoring stage. The offline stage detects how much time in maximum an equipment operation needs to reach its steady state from the initial switching on time instant. This time period is used as the maximum delay time for alarm unsuppression. Meanwhile, dynamic alarm limits are formulated based on the data associated with normal start-up operations. In the online stage, these two outcomes are used for alarm monitoring of the equipment start-up operations. First, the unsuppression of the original alarm is determined using a moving window PCA method, i.e., the original alarm will not be unsuppressed until a delay time is reached and this delay time must be no more than the maximum one detected in the offline stage. Meanwhile, a systematic algorithm is proposed to achieve effective alarm monitoring based on the dynamic alarm limits and online calculation of the unsuppression delay time. Results from an industrial case study demonstrate that the proposed method is capable of preventing nuisance alarms as well as achieving effective alarm monitoring for equipment start-up operations.

Chapter 3

Generalized Pattern Matching of Alarm Flood Sequences*

3.1 Overview

Addressing alarm floods in a large-scale industrial facility is not easy, owing to disparate fault types in many different locations and a variety of consequential alarm flood sequences; therefore, analyzing alarm floods individually would be a large undertaking. If alarm floods from different processes but associated with the same fault type are discovered and grouped, the obtained results could be utilized to identify common root causes and give generalized solutions. To our best knowledge, the above problem is open and unsolved. Motivated by such a practical problem, a systematic pattern matching method is proposed in this chapter to compare alarm floods across different processes. There are two major contributions:

1. A word processing approach is proposed to distill key words from textural alarm attributes and reconstruct abstracted alarm descriptors, so as to generalize representations for alarms from different processes;

*A version of this chapter has been published as: Boyuan Zhou, Wenkai Hu, Kevin Brown, and Tongwen Chen, Generalized pattern matching of industrial alarm flood sequences via word processing and sequence alignment. *IEEE Transactions on Industrial Electronics*, early access. A short version has been published as Boyuan Zhou, Wenkai Hu, and Tongwen Chen, Cross-process alarm flood similarity analysis based on abstracted alarm descriptors. *28th IEEE International Symposium on Industrial Electronics (ISIE)*, 1749–1754, Vancouver, Canada, 2019.

2. A pattern matching approach is proposed to compare alarm floods across different processes through three steps, including the unit-based extraction, the set-based pre-matching, and the sequence-based comparison.

3.2 Framework of the Method

Alarm floods are the most difficult situations for industrial alarm management. Pattern analysis is reorganized as one of the potential ways that can effectively address alarm floods as investigated in the literature, including similarity comparisons [18, 19, 21, 43, 59, 61, 66] and data mining techniques [15, 40, 54, 72, 89, 100]. Further, the obtained results can be used to facilitate the root cause analysis of similar alarm floods and to predict incoming alarm floods for online applications. However, this analysis becomes sophisticated and time-consuming when the process is very complex and there exist many types of faults leading to all kinds of alarm floods.

Modern industrial facilities, such as petroleum refineries, power stations, chemical plants, and pipeline systems, are usually in very large scales and comprised of numerous processes or units, which may contain the same types of equipment or have similar functionalities. For instance, a thermal power station may contain multiple coal storages to supply fuels or have several power units to generate electricity; an oil & gas pipeline system is made up of a series of pump stations, which could be built with identical architectures. In such circumstances, similar processes or units are susceptible to the same type of faults. Consequently, generated alarm floods may have analogous series of consequential alarms, which are of the same types but assigned with different tag names. If such similar alarm floods across different processes could be discovered and grouped, the obtained results would significantly increase the efficiency in alarm flood analysis by giving generalized solutions.

However, the existing methods in [15, 18, 19, 21, 40, 43, 54, 59, 61, 66, 72, 89, 100] are not applicable for the above problem, because they all compare alarm floods based on tag names, which give unique representations of alarms.

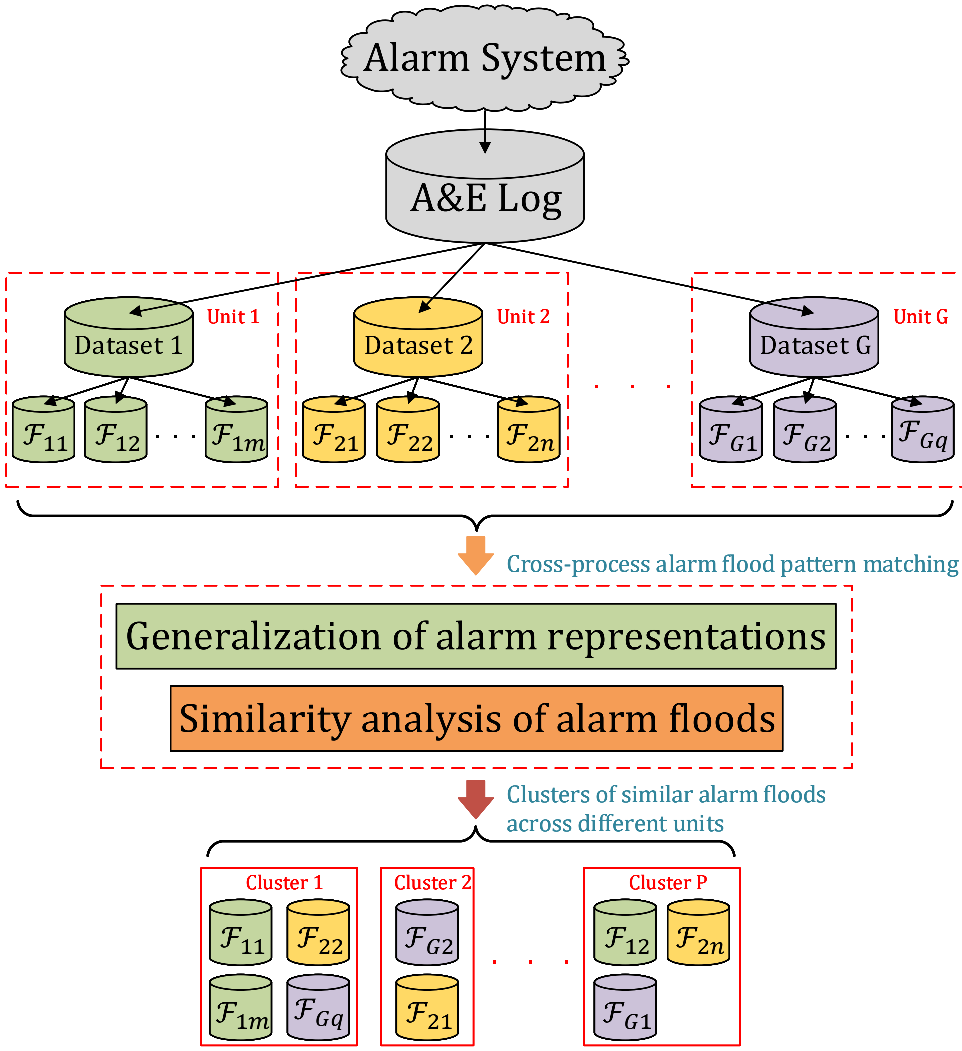

Figure 3.1: Framework for the cross-process alarm flood pattern matching.

In other words, alarms in different processes or units have different tag names, and thus make them not directly comparable. Consequently, alarm floods from different processes or units are typically treated as different situations, even if they are associated with the same type of fault.

Motivated by such a practical problem, a systematic method for cross-process alarm flood pattern matching is proposed. The framework of the proposed method is shown in Fig. 3.1. First, alarm floods in each process unit are extracted from the historical Alarm & Event log. Second, cross-process alarm flood pattern matching is conducted in two stages, including the generalization of alarm representations and the similarity analysis of alarm floods. Last, the clusters of similar alarm floods are obtained. Details of the proposed method are presented in Sections 3.3 and 3.4 with respect to the two stages for cross-process alarm flood pattern matching. In this work, the usage of the term "pattern matching" follows the study in [21] to represent sequence alignment based similarity analysis between alarm floods, but the meaning is also extended to cover the first step, namely, the generalization of alarm representations.

3.3 Generalization of Alarm Representations

This section presents the generalization of alarm representations, which includes three major steps, namely, the distillation of key words, the determination and removal of stop words, and the reconstruction of abstracted alarm descriptors.

3.3.1 Alarm Descriptions

The analysis of alarm floods is based on the structured textual data, known as Alarm & Event (A&E) logs. According to [38], there are two categories of alarm information recorded in A&E logs, including alarm configurations and alarm events. Alarm configurations are referred to as alarm descriptive attributes, such as tag names, process units, and detailed descriptions. Denote the finite set of alarms by $\mathcal{A}$. An alarm $a \in \mathcal{A}$ is described by a few descriptive attributes as

$$a = (\epsilon_a, \zeta_a, \chi_a), \tag{3.1}$$

where ϵ_a represents the tag name, namely, a unique label of an alarm a; ζ_a indicates the process unit, namely, the location where the alarm is configured; χ_a denotes the detailed description, which provides detailed information about a, such as the tag name, process name, equipment name, alarm type, process type, and process measurement.

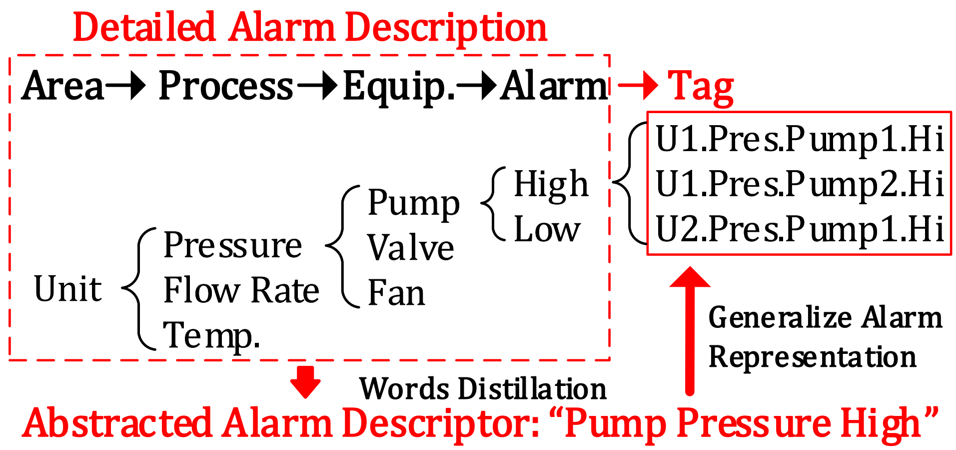

Figure 3.2: Reconstruction of an abstracted alarm descriptor based on frequent key words distilled from detailed alarm descriptions.

Alarm events are referred to as real-time events recorded to indicate transitions of alarm states, such as occurrences or clearances of alarms, and their corresponding time stamps. An event record for $a \in \mathcal{A}$ at time instant t is given by

$$E = (a, t, s), \tag{3.2}$$

where $t \in \mathcal{T}$ denotes the time stamp and $\mathcal{T}$ represents the time period of the studied data; $s \in \mathcal{S}$ indicates the transition of alarm states; $\mathcal{S}$ is a set representing alarm state transitions, e.g., $\mathcal{S} = \{0, 1\}$ with 1 and 0 standing for alarm occurrence and clearance, respectively.

An A&E log is essentially a sequence of chronologically ordered alarm events. In an A&E log, each alarm is uniquely identified by its tag name. Therefore, distinct alarms in different processes have different tag names, even if they are associated with the same type of fault. To compare alarm floods across different processes, it is necessary to overcome the restrictions caused by the difference in tag names. In this section, a systematic method is proposed to generalize alarm representations by reconstructing abstracted alarm descriptors based on key words distilled from detailed alarm descriptions. The reconstruction of abstracted alarm descriptor is illustrated by an example in

Fig. 3.2: the three alarms have different tag names, but indicate the same type of abnormality, namely, high pump discharge pressure. Based on key words distilled from alarm descriptions, the abstracted alarm descriptor "Pump Pressure High" is reconstructed to represented these alarms identically.

To generalize alarm representations from detailed alarm descriptions in historical A&E log, a word preprocessing method consisting of three major steps is proposed in the following subsections. The proposed method to generalize alarm representations is based on [17, 23, 68, 114], but it is not a simple combination of these techniques. The main contributions are: 1) A two-step word conversion principle is proposed by adapting the bag-of-words model in [23] based on actual problems on textual records in historical Alarm & Event logs. 2) The criteria for the determination and removal of stop words are specifically designed for the studied problem, and are critical in the generalization of alarm representations, so as to reduce differences caused by useless words.

3.3.2 Distillation of Key Words

At first, key words are distilled from detailed alarm descriptions. For alarm $a \in \mathcal{A}$, its detailed description χ_a is converted into a bag-of-words model [23] and denoted as $\tilde{\chi}_a = \{\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_{|\tilde{\chi}_a|}\}$, which is essentially a set of words contained in χ_a. Here, $\tilde{x}_j$ represents the jth word in $\tilde{\chi}_a$, $j = 1, 2, \cdots, |\tilde{\chi}_a|$, and $|\tilde{\chi}_a|$ is the number of words. A two-step word conversion principle is proposed by adapting the bag-of-words model in [23] based on actual problems on textual records in historical Alarm & Event logs as follows:

1. *Extract words without considering the orders*: This step purifies detailed descriptions by removing the orders of words and discarding meaningless connection symbols, such as space, comma, and special characters (e.g., - and /). It should be noticed that repeated words in χ_a are also excluded.

2. *Unify words with different spellings but the same meaning*: It is common to see that words with different spellings may have the same meaning. For example, the plural form of a noun could be different from its singular

form. The abbreviation or a portion of a word is used instead of its full spelling. Unifying such words can be based on alarm management manuals and basic lexicons.

Thereafter, the set containing bag-of-words models for all alarms in $\mathcal{A}$ is obtained as

$$\tilde{\mathbb{X}} = \{\tilde{\chi}_{a_1}, \tilde{\chi}_{a_2}, \cdots, \tilde{\chi}_{a_{|\mathcal{A}|}}\}, \tag{3.3}$$

where $\tilde{\chi}_{a_i}$ denotes the bag-of-words model for alarm $a_i \in \mathcal{A}$, $i = 1, 2, \cdots, |\mathcal{A}|$. Then, the complete set of words from all bag-of-words models is collected as $\Psi = \bigcup_{i=1}^{|\mathcal{A}|} \tilde{\chi}_{a_i}$.

3.3.3 Determination and Removal of Stop Words

In the extracted bag-of-words models, there are a variety of useless words (or known as stop words in [17]) that should be removed. In this subsection, three criteria are proposed to determine and exclude stop words.

Criterion 1 by the Unit-Based Occurrence Frequency

A unit-specific word is determined as a stop word based on how frequently it appears in bag-of-words models. Denote the set of all process units by $\mathcal{Z} = \{\zeta^k | k = 1, 2, \cdots, |\mathcal{Z}|\}$, where ζ^k denotes the kth process unit and $|\mathcal{Z}|$ is the number of process units. The set comprised by the bag-of-words models of alarms from unit $\zeta^k \in \mathcal{Z}$ is

$$\tilde{\mathbb{X}}(\zeta^k) = \{\tilde{\chi}_a | \tilde{\chi}_a \in \tilde{\mathbb{X}}, \zeta_a = \zeta^k, a \in \mathcal{A}\}, \tag{3.4}$$

where $\tilde{\chi}_a$ and ζ_a are the bag-of-words model and the process unit of alarm a, respectively. Then, the set of words from $\tilde{\mathbb{X}}(\zeta^k)$ is obtained as $\Psi(\zeta^k) = \bigcup_{\tilde{\chi}_a \in \tilde{\mathbb{X}}(\zeta^k)} \tilde{\chi}_a$.

Denote x as a unique word in Ψ and β_k as the presence of x in the kth unit. Then, $\beta_k = 1$, if $x \in \Psi(\zeta^k)$; otherwise, $\beta_k = 0$. The unit-based occurrence frequency of $x \in \Psi$ is calculated as $\Omega_u(x) = \sum_{k=1}^{|\mathcal{Z}|} \beta_k$, which indicates the number of process units having x in their bag-of-words models. With the unit-based occurrence frequency, the set of stop words is determined as

$$\xi_1 = \{x | x \in \Psi, \Omega_u(x) \leq \Omega_{u,th}\}, \tag{3.5}$$

where the user-defined threshold $\Omega_{u,th}$ is set as 1 by default, such that all unit-specific words are captured as stop words.

Criterion 2 by the Tag-Based Occurrence Frequency

The remained key words to describe an alarm should be neither too general, such as prepositions and conjunctions, nor too specific, such as the tag name of a device. The former is usually frequent while the latter is relatively rare. Both of them should be taken as stop words, which can be determined by the tag-based occurrence frequency. Denote γ_i as the presence of one word x in $\tilde{\chi}_{a_i}$ for $a_i \in \mathcal{A}$, where $i = 1, 2, \cdots, |\mathcal{A}|$. Then, $\gamma_i = 1$, if $x \in \tilde{\chi}_{a_i}$; otherwise, $\gamma_i = 0$. Thus, the tag-based occurrence frequency of $x \in \Psi$ is calculated as $\Omega_a(x) = \sum_{i=1}^{|\mathcal{A}|} \gamma_i$, which denotes the number of alarms having x in their bag-of-words models. With the tag-based occurrence frequency, the set of stop words is determined as

$$\xi_2 = \{x | x \in \Psi, \Omega_a(x) \geq \overline{\Omega}_a \text{ or } \Omega_a(x) \leq \underline{\Omega}_a\}, \tag{3.6}$$

where the two thresholds $\overline{\Omega}_a$ and $\underline{\Omega}_a$ are user-defined upper and lower bounds of the tag-based occurrence frequency, and they can be set as $|\mathcal{A}|$ and 1 by default, respectively, where $|\mathcal{A}|$ represents the number of unique alarms. For improvement, the two thresholds can also be determined by the Zipf curve [68]; or industrial document and process knowledge can be incorporated to ensure that no key words to distinguish alarms are determined as stop words. The reason for setting the two thresholds is that the key words to describe an alarm should be neither too general nor too specific, where the former is usually very frequent while the latter is relatively rare.

Criterion 3 by the Word Length

In alarm descriptions, there may exist short words that are of little value to distinguish alarms. Therefore, more stop words can be determined by word length and obtained as

$$\xi_3 = \{x | x \in \Psi, L(x) \leq L_{th}\}, \tag{3.7}$$

where $L(x)$ denotes the number of letters in x; L_{th} is a user-defined threshold for word length, which can be set as 2 by default, so as to remove short words, such as “a”, “is”, and “as”, which are useless in distinguishing alarms.

Eventually, all bag-of-words models in $\tilde{\mathbb{X}}$ are purified by removing stop words determined in ξ_1, ξ_2, and ξ_3 based on Criteria 1, 2, and 3, respectively. The set of purified bag-of-words models is

$$\bar{\mathbb{X}} = \{\bar{\chi}_{a_1}, \bar{\chi}_{a_2}, \cdots, \bar{\chi}_{a_{|\mathcal{A}|}}\}, \tag{3.8}$$

where $\bar{\chi}_{a_i} = \tilde{\chi}_{a_i} \setminus \{\xi_1 \cup \xi_2 \cup \xi_3\}$, $i = 1, 2, \cdots, |\mathcal{A}|$. The operator $\setminus$ denotes the exclusion of elements from a set.

Remark 4. For applications in different industrial systems, the settings for the generalization of alarm representations should be changed accordingly. More specifically, it should be decided at first based on process knowledge what data attributes to be selected from the historical Alarm & Event log, since different systems may have different data formats. In some systems, selecting one attribute providing the full alarm descriptions is enough; whereas in other systems multiple attributes, such as alarm types, process types, and tag names, might be needed if there is no such attribute containing the full descriptions.

3.3.4 Reconstruction of Abstracted Alarm Descriptors

In this subsection, abstracted alarm descriptors are reconstructed from the purified set $\bar{\mathbb{X}}$ through a closed itemset mining algorithm [114]. Following the concept in [114], *Abstracted Alarm Descriptors* are mathematically defined as follows.

Definition 2. An abstracted alarm descriptor $\mathcal{E}$ is a set of key words satisfying the following conditions:

1. $\mathcal{E} = \{\bar{x}_p | \bar{x}_p \in \bar{\Psi}, p = 1, 2, \cdots, |\mathcal{E}|\}$ with its support $\sigma(\mathcal{E}) \geq \sigma_{th}$;
2. There exists no $\mathcal{E}' = \{\bar{x}_q | \bar{x}_q \in \bar{\Psi}, q = 1, 2, \cdots, |\mathcal{E}'|\}$, such that $\mathcal{E} \subset \mathcal{E}'$ and $\sigma(\mathcal{E}) = \sigma(\mathcal{E}')$.

Here, set $\mathcal{I}(\mathcal{E}) = \{a | a \in \mathcal{A}, \mathcal{E} \subseteq \bar{\chi}_a\}$ is comprised of all alarms represented by $\mathcal{E}$. The support $\sigma(\mathcal{E}) = |\mathcal{I}(\mathcal{E})|$ is the size of $\mathcal{I}(\mathcal{E})$. The minimum support σ_{th} is used to guarantee that all abstracted alarm descriptors are frequent and $\bar{\Psi} = \Psi \setminus \{\xi_1 \cup \xi_2 \cup \xi_3\}$ is the complete set of words with stop words removed. $\square$

Given the set of alarms and their corresponding bag-of-words models as $\Theta = \{(a, \bar{\chi}_a) | a \in \mathcal{A}, \bar{\chi}_a \in \bar{\mathbb{X}}\}$. The closed itemset mining algorithm [114] is applied to Θ to obtain a finite set containing abstracted alarm descriptors, which are given together with their corresponding alarms as

$$\mathbb{C} = \{\mathcal{E}^p \times \mathcal{I}(\mathcal{E}^p) | p = 1, 2, \cdots, |\mathbb{C}|\}, \tag{3.9}$$

where the operator $\times$ denotes the pair of the pth abstracted alarm descriptor $\mathcal{E}^p$ and its corresponding set $\mathcal{I}(\mathcal{E}^p)$ of alarms that can be generally represented by $\mathcal{E}^p$. Obviously, all alarms should have their abstracted alarm descriptors, so that the union of all $\mathcal{I}(\mathcal{E}^p)$'s gives the complete set of alarms, i.e, $\mathcal{A} = \cup_{p=1,2,\cdots,|\mathbb{C}|} \mathcal{I}(\mathcal{E}^p)$. Besides, each alarm can only be represented by one abstracted alarm descriptor; and thus the sets $\mathcal{I}(\mathcal{E}^p)$ and $\mathcal{I}(\mathcal{E}^q)$ are disjoint, i.e., $\mathcal{I}(\mathcal{E}^p) \cap \mathcal{I}(\mathcal{E}^q) = \emptyset$ if $p \neq q$, where $\emptyset$ denotes the empty set.

For an alarm $a \in \mathcal{A}$, its abstracted alarm descriptor is $\mathcal{E}_a = \{\mathcal{E} | a \in \mathcal{I}(\mathcal{E})\}$. Then, the tuple of alarm a in eqn. (3.1) is augmented by the abstracted alarm descriptor $\mathcal{E}_a$ as $a^* = (\epsilon_a, \zeta_a, \chi_a, \mathcal{E}_a)$, and its corresponding record E in eqn. (3.2) is replaced by $E^* = (a^*, t, s)$. Finally, alarm representations are generalized, such that alarms indicating the same type of abnormalities are represented identically. Thereafter, similarity analysis is conducted to discover similar alarm floods across different processes based on such abstracted alarm descriptors.

For the generalization of alarm representations, other alternative approaches may be exploited. For example, tag names could be utilized to quickly generalize alarm descriptions, if the tag structures are clearly known such that alarms are distinguishable based on the prefix, postfix, and certain letters in tag names. Besides, other natural language processing techniques, such as grammar induction, terminology extraction, and part-of-speech tagging [48], may also be applied to obtain abstracted alarm descriptors, and thus deserve further studies.

3.4 Cross-Process Similarity Analysis of Alarm Floods

To capture similar alarm floods across different processes, a systematic similarity analysis method is proposed in this section; three main steps are involved, including the unit-based sequence extraction, the set-based pre-matching, and the sequence-based comparison.

3.4.1 Unit-Based Sequence Extraction

Based on an A&E log, alarm flood sequences are extracted. To avoid false identification of alarm floods caused by chattering alarms, delay timers should be applied first [97]. Moreover, to compare alarm floods across different processes, the extraction of alarm flood sequences must consider the process unit information. In different systems, the grouping of alarms based on sub-units or sub-processes could be different; therefore, prior process knowledge should be incorporated to decide the sub-units or sub-processes in the step of unit-based alarm flood extraction. For instance, to compare alarm floods among several coal mills in a power plant, a unit is a coal mill system; while a unit becomes a pump station when the comparison is conducted for pipeline systems.

For a process unit $\zeta^k \in \mathcal{Z}$, its A&E log is given by

$$\mathbb{D}(\zeta^k) = \langle E_1^{\zeta^k}, E_2^{\zeta^k}, \cdots, E_{|\mathbb{D}(\zeta^k)|}^{\zeta^k} \rangle, \tag{3.10}$$

where $\langle \cdot \rangle$ denotes a sequence and $E_i^{\zeta^k}$ is the ith event record in $\mathbb{D}(\zeta^k)$, $i = 1, 2, \cdots, |\mathbb{D}(\zeta^k)|$. Here, $\mathbb{D}(\zeta^k)$ is utilized to extract alarm floods following the definition in ANSI/ISA-18.2 [49]. Thereafter, the set of alarm floods from all process units is

$$\mathbb{F} = \bigcup_{k=1}^{|\mathcal{Z}|} \mathbb{F}(\zeta^k), \tag{3.11}$$

where $\mathbb{F}(\zeta^k)$ gives all alarm floods from process unit ζ^k.

Later, similarity analysis for alarm floods across different processes is conducted through the set-based pre-matching and the sequence-based comparison.

3.4.2 Set-Based Pre-Matching

Set-based pre-matching is a critical step to efficiently exclude dissimilar floods. Alarm floods are pre-matched without considering the orders of alarms prior to the sequence-based comparison. The latter yields to more accurate similarity analysis but has higher computational complexity.

An alarm flood is a sequence of alarms and denoted by

$$\mathcal{F} = \langle a_1, a_2, \cdots, a_{|\mathcal{F}|} \rangle, \tag{3.12}$$

where a_i is the ith alarm in $\mathcal{F} \in \mathbb{F}$, $i = 1, 2, \cdots, |\mathcal{F}|$. Given a pair of alarm floods $\mathcal{F}_i$, $\mathcal{F}_j \in \mathbb{F}$, their set-based similarity score is calculated based on the Szymkiewicz-Simpson coefficient [22] as

$$\Phi_0 = \frac{|\Gamma_{\mathcal{F}_i} \cap \Gamma_{\mathcal{F}_j}|}{\min(|\Gamma_{\mathcal{F}_i}|, |\Gamma_{\mathcal{F}_j}|)}, \tag{3.13}$$

where $\Gamma_{\mathcal{F}} = \{\mathcal{E}_a | a \in \mathcal{F}\}$ denotes the set of abstracted alarm descriptors for all alarms in $\mathcal{F}$. The operator $\cap$ stands for set intersection and $\min(\cdot)$ gives the minimum of input values.

The computation proceeds to the sequence-based comparison between alarm floods $\mathcal{F}_i$ and $\mathcal{F}_j$, if and only if they have enough abstracted alarm descriptors in common, i.e., $\Phi_0 \geq \Phi_{0,th}$, where $\Phi_{0,th}$ is a user defined threshold to exclude dissimilar alarm flood pairs. Generally, selecting a larger value of $\Phi_{0,th}$ would lead to the exclusion of more alarm floods from the sequence-based comparison, and vice versa. By default, set $\Phi_{0,th} = 0$, such that the alarm floods without common abstracted alarm descriptors are excluded [43].

3.4.3 Sequence-Based Comparison

In this subsection, alarm floods passing the set-based pre-matching are compared based on sequences of alarms. A sequence alignment method, namely, the Smith-Waterman algorithm [83], is exploited and adapted to achieve comparisons between alarm floods across different processes. To make this algorithm more compatible with the alignment of alarm flood sequences, time stamps of alarm events are incorporated [21]. The algorithm in [21] is further adapted by revising the scoring function based on abstracted alarm descriptors, so as to achieve cross-process alarm flood pattern matching.

Given a flood sequence $\mathcal{F}$, its time weighting matrix is

$$\mathbf{W}_{\mathcal{F}} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_{|\mathcal{F}|}], \tag{3.14}$$

where $\mathbf{w}_i = [w_i^1, w_i^2, \cdots, w_i^{|\mathbb{C}|}]$ denotes the time weighting vector for $a_i \in \mathcal{F}$; the pth element of $\mathbf{w}_i$ is $w_i^p = f(d_i^p)$, $p = 1, 2, \cdots, |\mathbb{C}|$; $f(\cdot)$ is the weighting function; d_i^p represents the time interval between a_i and the alarm with abstracted alarm descriptor $\mathcal{E}^p$ in $\mathcal{F}$. The time interval d_i^p is obtained as

$$d_i^p = \begin{cases} \min\limits_{a_j \in \mathcal{F}} \{|t_i - t_j| : \mathcal{E}_{a_j} = \mathcal{E}^p, \mathcal{E}^p \in \mathbb{C}\}, \\ \quad \text{if the set is not empty}, \\ \infty, \text{otherwise}, \end{cases} \tag{3.15}$$

where t_i and $\mathcal{E}_{a_i}$ (t_j and $\mathcal{E}_{a_j}$) are the time stamp and abstracted alarm descriptor of alarm a_i (a_j) in $\mathcal{F}$, respectively.

Given alarm floods $\mathcal{F}_i$, $\mathcal{F}_j \in \mathbb{F}$, consisting of abstracted alarm descriptors, the sequence-based similarity score Φ and the score matrix $\mathbf{H}$ are calculated

Algorithm 2 $[\Phi_s, \mathcal{L}] = \text{Fcn\_Sqc}(\mathcal{F}_i, \mathcal{F}_j, \mu, \delta)$: Sequence-Based Alarm Flood Comparison

Input Arguments: $\mathcal{F}_i$, $\mathcal{F}_j$, μ, δ
Output Arguments: Φ_s, $\mathcal{L}$
Calculate $\Phi_{\mathcal{F}_i\mathcal{F}_j}$ and $\mathbf{H}_{\mathcal{F}_i\mathcal{F}_j}$ based on $\mathcal{F}_i, \mathcal{F}_j, \mu, \delta$
Calculate $\Phi_{\mathcal{F}_j\mathcal{F}_i}$ and $\mathbf{H}_{\mathcal{F}_j\mathcal{F}_i}$ based on $\mathcal{F}_j, \mathcal{F}_i, \mu, \delta$
$\Phi_s = \max(\Phi_{\mathcal{F}_i\mathcal{F}_j}, \Phi_{\mathcal{F}_j\mathcal{F}_i}) / \min(|\mathcal{F}_i|, |\mathcal{F}_j|)$
if $\Phi_{\mathcal{F}_i\mathcal{F}_j} \geq \Phi_{\mathcal{F}_j\mathcal{F}_i}$ **then**
    $\mathbf{H} = \mathbf{H}_{\mathcal{F}_i\mathcal{F}_j}$
else
    $\mathbf{H} = \mathbf{H}_{\mathcal{F}_j\mathcal{F}_i}$
end if
Initialize $(p, q) = \arg\max \mathbf{H}(p, q)$, $k = 1$
while $\mathbf{H}(p, q) \neq 0$ **do**
    Record $\mathcal{L}(k) = (p, q)$
    $H_0 = \max(\mathbf{H}(p-1, q), \mathbf{H}(p-1, q-1), \mathbf{H}(p, q-1))$
    $$(p, q) = \begin{cases} (p-1, q), \text{if } H_0 = \mathbf{H}(p-1, q) \\ (p, q-1), \text{if } H_0 = \mathbf{H}(p, q-1) \\ (p-1, q-1), \text{otherwise} \end{cases}$$
    $k = k + 1$
end while

using the modified Smith-Waterman algorithm [21]. First, the time weighting matrices $\mathbf{W}_{\mathcal{F}_i}$ and $\mathbf{W}_{\mathcal{F}_j}$ for $\mathcal{F}_i$ and $\mathcal{F}_j$ are computed by eqn. (3.14), using different weighting functions $f_1(\cdot)$ and $f_2(\cdot)$. Here, $f_1(x) = e^{-x/2\eta^2}$, where η is the scaling parameter; $f_2(x) = 1$, iff $x = 0$; otherwise, $f_2(x) = 0$. Second, $\mathbf{H}$ is initialized as an all-zero matrix with $|\mathcal{F}_i| + 1$ rows and $|\mathcal{F}_j| + 1$ columns. Third, $\mathbf{H}$ is recursively updated as $\mathbf{H}(m+1, n+1) = \max(H_1, H_2, H_3, 0)$, where $m = 1, \cdots, |\mathcal{F}_i|$, $n = 1, \cdots, |\mathcal{F}_j|$, $H_1 = \mathbf{H}(m, n+1) + \delta$, $H_2 = \mathbf{H}(m, n) + \phi(a_m^i, a_n^j)$, and $H_3 = \mathbf{H}(m+1, n) + \delta$. The match score between a_m^i and a_n^j is $\phi(a_m^i, a_n^j) = \max\{\mathbf{W}_{\mathcal{F}_i}(m) \circ \mathbf{W}_{\mathcal{F}_j}(n)\}(1 - \mu) + \mu$, where $\circ$ indicates the element-wise product of two vectors and $\mathbf{W}_{\mathcal{F}_i}(m)$ ($\mathbf{W}_{\mathcal{F}_j}(n)$) represents the corresponding column in the time weighting matrix for the mth (nth) alarm $a_m^i \in \mathcal{F}_i$ ($a_n^j \in \mathcal{F}_j$). The parameters μ and δ are mismatch penalty and gap penalty, respectively. By default, select $-1 < \mu < 2\delta < 0$ to prefer gapped alignment instead of mismatch. Last, the final similarity score is the maximum element of $\mathbf{H}$, i.e., $\Phi = \max(\mathbf{H})$.

Due to the selection of weighting functions $f_1(\cdot)$ and $f_2(\cdot)$, the obtained alarm flood similarity score Φ is asymmetric. Therefore, to deal with such a problem, the comparison of alarm floods is conducted in two ways as in lines 3-4 of Algorithm 2, which eventually gives the optimal normalized similarity score as $\Phi_s \in [0, 1]$ in line 5, and its corresponding score matrix $\mathbf{H}$ in lines 6-10. Thereafter, the optimal alignment $\mathcal{L}$ of $\mathcal{F}_i$ and $\mathcal{F}_j$ is determined in lines 11-17 through the recursive search of the index pair (p, q) that yields to the optimal alignment in each updating step.

It should be noted that even though the idea is based on the methods in [21, 43], the proposed method in this section is different from [21, 43] in three aspects: the incorporation of process unit information for alarm flood extraction, the set-based pre-matching by Szymkiewicz-Simpson coefficients with abstracted alarm descriptors, and the generalized sequence alignment to capture similar alarm floods across different processes.

3.4.4 Implementation and Discussions

For implementation, the complete procedures of cross-process alarm flood pattern matching are summarized in Algorithm 3. The input is the A&E log $\mathbb{D}$, and the outputs are the similarity matrix $\mathbf{S}$ and the set of alignments $\mathbf{L}$. There are two main calculation stages, including the generalization of alarm representations (lines 3-6) and the similarity analysis of alarm floods (lines 7-15).

As for the computational complexity, the proposed method for generalized alarm flood pattern matching is conducted in an offline manner; its computational complexity is mainly comprised of two parts, namely, C_1 for the generalization of alarm representations and C_2 for the comparison of alarm floods. More specifically, $C_1 = \mathcal{O}\left(|\mathcal{A}| \max_{a \in \mathcal{A}} |\tilde{\chi}_a|\right)$ and $C_2 = \mathcal{O}\left(|\mathbb{F}|^2 MN |\mathbb{C}|\right)$, where $\mathcal{O}(\cdot)$ is the big-O notation of computational complexity, $|\mathcal{A}|$ denotes the number of alarms, $|\tilde{\chi}_a|$ is the number of key words in the bag-of-words

Algorithm 3 $[\mathbf{S}, \mathbf{L}]$ = Fcn_Main($\mathbb{D}$): Cross-Process Alarm Flood Pattern Matching

Input Argument: $\mathbb{D}$
Output Arguments: $\mathbf{S}$, $\mathbf{L}$
Obtain the set of bag-of-words models $\tilde{\mathbb{X}}$ by eqn. (3.3)
Determine sets of stop words as ξ_1, ξ_2, and ξ_3 by eqns. (3.5), (3.6), and (3.7), respectively
Purify the set of bag-of-words models $\bar{\mathbb{X}}$ by eqn. (3.8)
Reconstruct abstracted alarm descriptors $\mathbb{C}$ by eqn. (3.9)
Collect alarm floods into set $\mathbb{F}$ by eqn. (4.1)
for $\forall \mathcal{F}_i, \mathcal{F}_j \in \mathbb{F}$ **do**
    Calculate set-based similarity score Φ_0 by eqn. (3.13)
    if $\Phi_0 \geq \Phi_{0,th}$ **then**
        $[\Phi_s, \mathcal{L}] = \text{Fcn\_Sqc}(\mathcal{F}_i, \mathcal{F}_j, \mu, \delta)$
        $\mathbf{S}(i,j) = \mathbf{S}(j,i) = \Phi_s$
        $\mathbf{L}(i,j) = \mathbf{L}(j,i) = \{\mathcal{L}\}$
    end if
end for

model for alarm $a \in \mathcal{A}$, $|\mathbb{F}|$ indicates the number of alarm flood sequences, M and N represent the lengths of two alarm flood sequences for comparison, and $|\mathbb{C}|$ is the number of abstracted alarm descriptors. It can be seen that the computational complexity is mainly related to the number of alarms, the number of key words, and the number of alarm floods. The computational complexity C_2 for the second part, namely, the comparison of alarm floods, is almost the same as that of the method in [21]. Since the proposed method is offline, there is no strict requirement for computational efficiency.

Remark 5. The applicability of the proposed method is based on two conditions: 1) The studied large-scale industrial facility must be comprised of similar sub-systems or sub-processes that resemble each other in functionalities or architectures, e.g., a thermal power plant with multiple coal mills or a pipeline system with many pump stations, such that these sub-systems or sub-processes have the same types of faults accompanied by certain alarm floods. 2) The analyzed historical alarm and event data must contain integrated textual information for each alarm rather than unexplainable code names or abbreviations such that key words can be extracted to construct abstracted alarm descriptors.

3.5 Industrial Case Study

The proposed method was tested in a real large-scale industrial facility. Details of the application results are presented in the following subsections.

3.5.1 Comparison of Pattern Matching Results

The studied industrial facility is comprised of quite a few similar process units with the same architecture and analogous functionalities. Alarm floods were found to appear commonly in these units, compromising the efficiency of the alarm system. The proposed method was applied to compare alarm floods, so as to figure out a generalized solution for similar alarm floods caused by the same fault but from different process units. Delay timers were applied to reduce chattering alarms, so as to avoid false identification of alarm floods. Then, alarm floods were extracted and collected into a set $\mathbb{F}$, where 5, 7, and 6 alarm flood sequences were found from three different units, respectively. The numerical indices of alarm floods in Unit 1 are $1, 2, \cdots, 5$; for Unit 2, the indices are $6, 7, \cdots, 12$; for Unit 3, the indices are $13, 14, \cdots, 18$. In total, 18 alarm flood sequences were extracted from the three units.

The obtained results using both the proposed method and the method in [21] were compared. The similarity color map in Fig. 3.3 displays the pattern matching results using the method in [21], where alarms are represented by exact tag names. Each number on the vertical and horizon axes represents the numerical index of an alarm flood, and the color gradation in each cell indicates the similarity score between an alarm flood on the horizontal axis and the other one on the vertical axis. Here, similarity scores for alarm floods from three different units are highlighted by solid red squares. Clusters of similar floods are detected and highlighted by dashed red rectangles.

Then, the proposed method was applied to the same A&E log. First,

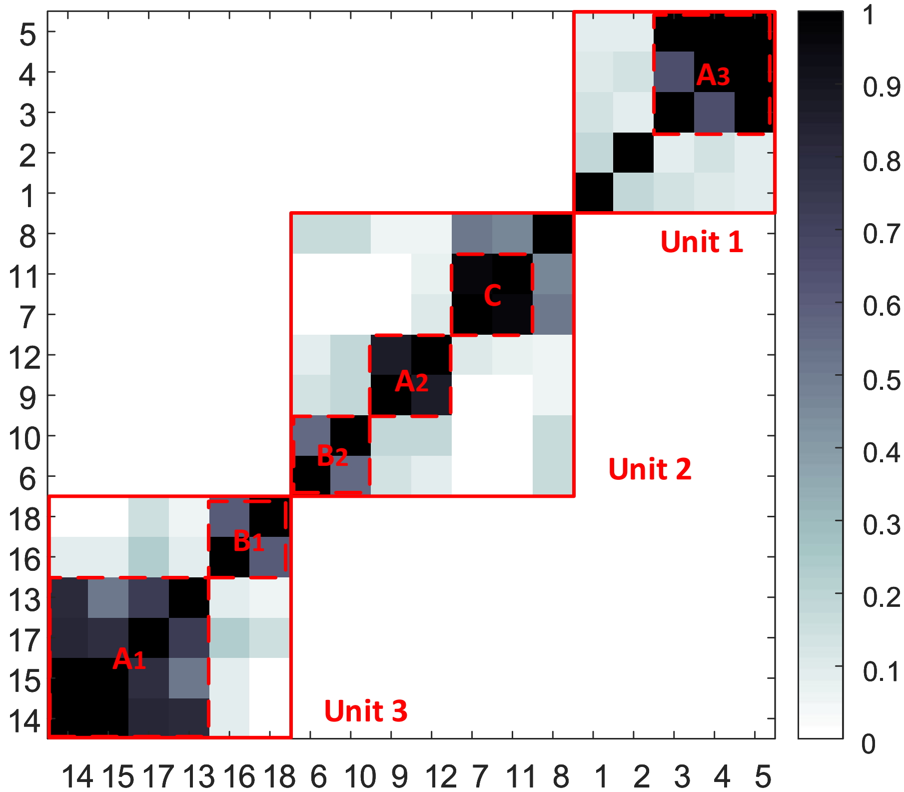

Figure 3.3: Alarm flood similarity color map based on exact alarm tags.

abstracted alarm descriptors were reconstructed to generalize alarm representations, following the steps in lines 3-6 of Algorithm 3. There were 388 unique alarms from the A&E log, and 155 key words were distilled from their detailed descriptions, such as "valve", "motor", and "level". In total, 100 unique abstracted alarm descriptors were obtained from the purified key words. Then, the similarity scores of alarm floods were recursively calculated based on abstracted alarm descriptors through lines 7-15 of Algorithm 3. The obtained results of similarity scores between alarm floods are shown in Fig. 3.4. To make it easier to compare, these alarm floods are sorted in the same order as that in Fig. 3.3.

Comparing Figs. 3.3 and 3.4, we have that the obtained similarity scores within each unit were almost identical by both methods. However, the sim-

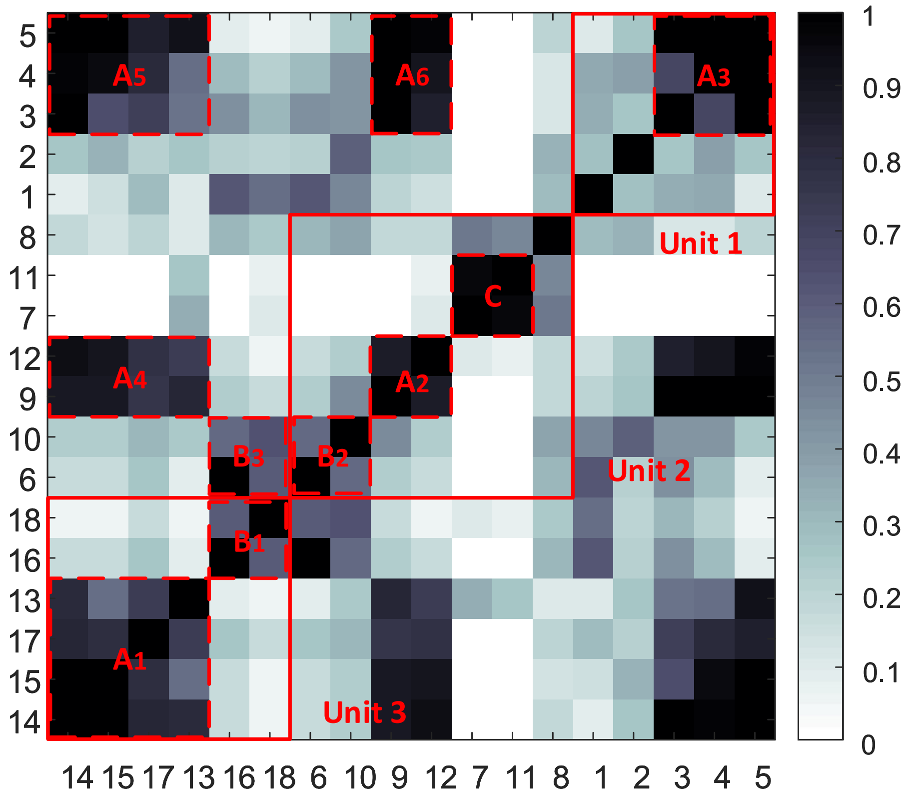

Figure 3.4: Alarm flood similarity color map based on abstract alarm descriptors.

ilarity scores between alarm floods across different units were significantly different using the two methods. More specifically, the proposed method discovered several clusters of similar alarm floods across different units, while the method in [21] did not find any of them. Eventually, three clusters are formed and highlighted by dashed red rectangles: cluster A is comprised of areas A1, A2, $\cdots$, and A6 associated with three different units; cluster B consists of areas B1, B2, and B3 associated with units 2 and 3; cluster C only contains alarm floods from unit 2.

Alarm Flood Sequence #5 (from Unit 1)		
Time	Tag	Abstracted Alarm Descriptor
21:18:20	U1.Tag.1	discharge valve travel open
21:21:57	U1.Tag.2	bypass valve auto
21:21:57	U1.Tag.58	bypass valve local
21:22:20	U1.Tag.62	suction valve open
21:22:25	U1.Tag.63	suction valve travel close
21:22:32	U1.Tag.64	suction valve close
21:22:42	U1.Tag.68	discharge valve open
21:22:55	U1.Tag.69	discharge valve close
21:23:10	U1.Tag.70	discharge valve open
21:23:15	U1.Tag.71	discharge valve travel close
21:23:25	U1.Tag.76	discharge valve close
21:23:27	U1.Tag.77	suction valve travel open
21:23:32	U1.Tag.78	suction valve open
21:23:40	U1.Tag.82	discharge valve travel close
21:23:45	U1.Tag.83	suction valve close

Alarm Flood Sequence #15 (from Unit 3)		
Time	Tag	Abstracted Alarm Descriptor
15:59:05	U3.Tag.1	bypass valve local
15:59:05	U3.Tag.2	bypass valve auto
15:59:19	U3.Tag.108	suction valve travel open
15:59:24	U3.Tag.109	suction valve open
15:59:29	U3.Tag.110	suction valve travel close
15:59:35	U3.Tag.114	suction valve close
15:59:39	U3.Tag.115	discharge valve travel open
15:59:44	U3.Tag.116	discharge valve open
15:59:49	U3.Tag.117	discharge valve travel close
15:59:54	U3.Tag.121	discharge valve close
16:00:09	U3.Tag.122	discharge valve open
16:00:19	U3.Tag.123	discharge valve close
16:00:24	U3.Tag.124	suction valve travel open
16:00:29	U3.Tag.127	suction valve open
16:00:39	U3.Tag.128	suction valve close
16:02:38	U3.Tag.129	discharge valve travel close

Figure 3.5: Sequence alignment of alarm floods #5 and #15 from cluster A. The two alarm floods were from two different units (Units 1 and 3), but associated with the same root cause, namely, the unusual valve operation mode.

Alarm Flood Sequence #6 (from Unit 2)		
Time	Tag	Abstracted Alarm Descriptor
2:29:08	U2.Tag.41	unit lockout
2:29:08	U2.Tag.42	unit power fail
2:29:08	U2.Tag.45	unit power fail
2:29:08	U2.Tag.46	unit power fail
2:29:08	U2.Tag.50	suction valve status unknown
2:29:08	U2.Tag.55	suction valve local
2:29:08	U2.Tag.61	suction valve status unknown
2:29:08	U2.Tag.66	suction valve local
2:29:08	U2.Tag.67	discharge valve status unknown
2:29:08	U2.Tag.72	discharge valve local
2:29:08	U2.Tag.74	discharge valve status unknown
2:29:08	U2.Tag.80	discharge valve local
2:29:38	U2.Tag.91	power monitor communication fail

Alarm Flood Sequence #16 (from Unit 3)		
Time	Tag	Abstracted Alarm Descriptor
10:21:05	U3.Tag.16	unit lockout
10:21:05	U3.Tag.17	unit power fail
10:21:41	U3.Tag.30	unit lockout
10:21:41	U3.Tag.46	unit power fail
10:22:27	U3.Tag.49	unit lockout
10:22:27	U3.Tag.57	unit power fail
10:23:20	U3.Tag.62	unit trouble
10:23:20	U3.Tag.72	unit lockout
10:23:20	U3.Tag.76	suction valve status unkown
10:23:20	U3.Tag.111	suction valve local
10:23:20	U3.Tag.112	suction valve status unkown
10:23:20	U3.Tag.118	suction valve local
10:23:20	U3.Tag.119	discharge valve status unkown
10:23:20	U3.Tag.125	discharge valve local
10:23:20	U3.Tag.126	discharge valve status unkown
10:23:20	U3.Tag.130	discharge valve local

Figure 3.6: Sequence alignment of alarm floods #6 and #16 from cluster B. The two alarm floods were from two different units (Units 2 and 3), but associated with the same root cause, namely, the power failure.

3.5.2 Clusters of Alarm Flood Sequences

The three clusters of alarm floods are presented with more detailed explanations as follows.

Alarm Flood Sequence #7 (from Unit 2)

Time	Tag	Abstracted Alarm Descriptor
15:20:22	U2.Tag.48	block valve travel open
15:20:28	U2.Tag.51	sump pump run
15:20:38	U2.Tag.56	block valve open
15:20:38	U2.Tag.62	discharge valve travel trap
15:20:38	U2.Tag.68	suction valve travel trap
15:20:42	U2.Tag.69	discharge valve trap
15:20:46	U2.Tag.71	suction valve trap
15:20:46	U2.Tag.76	valve trap command unknown
15:20:46	U2.Tag.83	discharge valve travel trap
15:20:52	U2.Tag.89	discharge valve trap

Alarm Flood Sequence #11 (from Unit 2)

Time	Tag	Abstracted Alarm Descriptor
6:38:52	U2.Tag.48	block valve travel open
6:38:52	U2.Tag.51	sump pump run
6:38:52	U2.Tag.56	block valve open
6:38:52	U2.Tag.62	discharge valve travel trap
6:38:56	U2.Tag.69	discharge valve trap
6:38:56	U2.Tag.71	suction valve trap
6:38:56	U2.Tag.76	valve trap command unknown
6:39:01	U2.Tag.83	discharge valve travel trap
6:39:01	U2.Tag.85	discharge valve travel trap
6:39:11	U2.Tag.89	discharge valve trap

Figure 3.7: Sequence alignment of alarm floods #7 and #11 from cluster C. The two alarm floods were from the same unit (Unit 2), and associated with the same root cause, namely, the valve trap.

Cluster A

Alarm floods in this cluster were from three different units. Fig. 3.5 presents an example of the sequence alignment of two alarm floods (#5 and #15) across units 1 and 3. Matched alarms between the two alarm flood sequences are connected by red lines. It can be seen that the matched alarms are identical in abstracted alarm descriptors but different in tag names. Therefore, only the proposed method discovered such similar alarm floods across different process units, while the method in [21] based on tag names failed. From the abstracted alarm descriptors, all these alarms were associated with valves and appeared almost in the same sequential order. In fact, the alarm floods in cluster A were all associated with the same root cause, namely, the unusual valve operation mode.

Cluster B

Alarm floods in this cluster were from units 2 and 3. Fig. 3.6 presents an example of the sequence alignment of two alarm floods (#6 and #16) across two different units. Matched alarms are found to appear in the same sequential order. This pair of similar alarm floods was also detected by the proposed method, while the similarity between them was zero using the method in [21]. In fact, the alarm floods in cluster B were all associated with the same root cause, namely, the power failure.

Cluster C

Alarm floods in this cluster were all from unit 2. Fig. 3.7 presents an example of the sequence alignment of two alarm floods (#7 and #11). Since the alarm floods were from the same process unit, the matched alarms have the same tag names and abstracted alarm descriptors. Thus, both the proposed method and the method in [21] succeeded in discovering such pair of similar alarm floods. In fact, the alarm floods in this cluster were all associated with the same root cause, namely, the valve trap in unit 2.

In summary, the results from clusters A and B show that the proposed method can discover similar alarm floods across different processes without the restriction of common tag names, whereas the method in [21] based on exact alarm tag names is not able to identify these similar alarm floods. The results from cluster C show that the proposed method leads to almost the same conclusions of sequence alignments as those detected using the method in [21]. Similar alarm floods across different units were discovered using the proposed method, where such alarm floods were associated with the same type of abnormalities. The obtained results could lead to generalized solutions to address such similar alarm floods together.

However, it is possible that strong similarities of alarm floods may arise due to the reduction of the informative contents in alarm descriptors, if such constructed alarm descriptors are too general, e.g., different process types or alarm types are ignored. To cope with the problem, the proposed generalization method of alarm representations makes sure that some key words are reserved. However, as shown in the industrial case study, some similarities for alarms within the same unit also increased. Since the proposed method is specifically designed for cross-process analysis of alarm floods, such increased similarities within the same unit are not desirable. To avoid such problems for similarity analysis within one process or unit, the method in reference [21] rather than the proposed method should be used.

3.6 Summary

To capture similar alarm floods across different processes, a systematic pattern matching method is proposed in this chapter. The method consists of two major steps, including the generalization of alarm representations and the similarity analysis of alarm floods. The effectiveness of the proposed method is demonstrated by a case study using alarm data from a real industrial facility. As shown in the case study, the generalization of alarm representations by abstracted alarm descriptors is critical for cross-process alarm flood pattern matching. The discovered clusters of similar floods can help to find common root causes, which lead to generalized solutions to address alarm floods from different processes all at once. As a result, the efficiency of alarm flood analysis can be significantly improved.

Chapter 4

Pattern Extraction from Industrial Alarm Flood Sequences*

4.1 Overview

As a fundamental step for alarm flood analysis, pattern extraction can effectively distill useful information from historical alarm & event data, and use the results for many purposes, such as configuration of alarm suppression modules, root cause analysis, and decision support for real-time alarm monitoring. However, due to complicated combinations of alarms across different alarm floods, the extraction of alarm patterns is not easy. Methods based on similarity calculations [19, 21, 43, 61] are capable of identifying similar alarm floods, but not straightforward in pattern extraction. For methods based on data mining [28, 40, 72, 89], alarm patterns are extracted directly from alarm floods without comparing them. However, these methods still have some shortcomings, such as lack of consideration for orders and time stamps [40] and incapability of handling order switchings caused by small time differences [28, 72, 89]. As a result, either alarm orders are not considered or critical alarms could be missing in the extracted patterns, which may

*A version of this chapter has been submitted for publication as: Boyuan Zhou, Wenkai Hu, and Tongwen Chen, Pattern extraction from industrial alarm flood sequences by a modified CloFAST algorithm. *IEEE Transactions on Industrial Informatics*.

impair the usability of extracted alarm patterns. Motivated by the above discussion, this chapter proposes a new method to extract alarm flood patterns from historical alarm flood sequences. The main contributions are twofold:

1. A closed alarm sequence mining approach is proposed based on the CloFAST method in [33] with improvement to incorporate time stamps and tolerate alarm order switchings.

2. A pattern distillation strategy is designed to merge similar alarm sequences and export more compact alarm sequential patterns, so as to cope with irrelevant alarms and different lengths of alarm flood sequences.

The proposed method is capable of avoiding the influences of order switchings caused by small time differences to pattern extraction, and also minimizing the redundancy of extracted alarm sequential patterns.

4.2 Framework of Pattern Extraction

In the industrial standard ANSI/ISA-18.2 [49], a detailed definition of an alarm flood is: "The start of an alarm flood is indicated by the first regular 10 minutes interval with an alarm rate that exceeds 10 alarms per 10 minutes. The end of an alarm flood is indicated by the first regular 10 minutes interval with an alarm rate of less than 5 alarms per 10 minutes." Accordingly, by calculating alarm rate and comparing it with the two thresholds, alarm floods can be identified from historical Alarm & Event (A&E) logs, which are essentially textural datasets comprised by a series of chronologically ordered alarm events. Thereafter, the alarm flood sequences are collected into a set as

$$\mathbb{F} = \{\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_{|\mathbb{F}|}\}, \tag{4.1}$$

where $\mathcal{F}_i = \langle E_1, E_2, \cdots, E_{|\mathcal{F}_i|}\rangle$ is the ith alarm flood, $i = 1, 2, \cdots, |\mathbb{F}|$. The operator $\langle\cdot\rangle$ indicates a sequence, and $|\cdot|$ stands for the size of a sequence, set, or list. Denote an alarm event in $\mathcal{F}_i$ as $E = (a, t)$, where $a \in \mathcal{A}$ is the tag name of a unique alarm, and time stamp $t \in \mathcal{T}$ indicates when this alarm occurs. Here, $\mathcal{A}$ is the finite set of all configured alarms in an alarm system and $\mathcal{T}$ denotes the studied time period. Thereafter, alarm sequential patterns are extracted from $\mathbb{F}$, so as to distill useful information that can help with alarm system improvement, such as root cause analysis, alarm suppression, and decision support for real-time alarm monitoring.

For this objective, a systematic alarm flood pattern extraction method is proposed based on a closed sequential pattern mining method called the CloFAST algorithm [33] with improvement in three major aspects, including the incorporation of time stamps, the tolerance of alarm order switchings, and the distillation of compact alarm sequential patterns. For better understanding, some relevant concepts are defined as follows.

Alarm Itemset

An alarm itemset is a set of alarms that have identical time stamps in each of the analyzed alarm floods. Denote an alarm itemset as $\mathcal{I} = \{a_1, a_2, \cdots, a_{|\mathcal{I}|}\}$, where $a_i \in \mathcal{A}$ is the ith alarm in $\mathcal{I}$, $i = 1, 2, \cdots, |\mathcal{I}|$. The support of $\mathcal{I}$ is defined as the number of alarm floods containing $\mathcal{I}$, i.e., $\sigma(\mathcal{I}) = |\{\mathcal{F} | \forall a \in \mathcal{I}, E = (a, t) \in \mathcal{F}\}|$. It should be noticed that, for each alarm flood $\mathcal{F} \in \mathbb{F}$, it is always satisfied that $t_i = t_j$, $\forall a_i, a_j \in \mathcal{I}$, where t_i (t_j) denotes the time stamp of a_i (a_j) in $\mathcal{F}$. As a result, the basic components of alarm floods are captured as alarm itemsets. To ensure the compactness of such basic components, it is desired that alarm itemsets are closed. Following the concept in [33], a closed alarm itemset is defined as follows.

Definition 3. Alarm itemset $\mathcal{I}$ is a closed itemset, iff

1. $\mathcal{I}$ is frequent, i.e., $\sigma(\mathcal{I}) \geq \bar{\sigma}_{\mathcal{I}}$, and
2. $\mathcal{I}$ is closed, i.e., there exists no super itemset $\mathcal{I}'$, such that $\mathcal{I} \subset \mathcal{I}'$ and $\sigma(\mathcal{I}') = \sigma(\mathcal{I})$.

Here, $\bar{\sigma}_{\mathcal{I}}$ is a user-defined threshold called minimum itemset support. $\square$

Alarm Sequence

An alarm sequence is comprised by a collection of closed alarm itemsets, which are sequentially ordered based on time stamps. Denote an alarm sequence as $\mathcal{S} = \langle \mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_{|\mathcal{S}|} \rangle$, where $\forall \mathcal{I}_i, \mathcal{I}_j \in \mathcal{S}$, $t_i \prec t_j$, if $i \leq j$. Here, t_i (t_j) is the time stamp of $\mathcal{I}_i$ ($\mathcal{I}_j$). The operator $\prec$ denotes a relaxed comparison of time stamps, where $t_i \prec t_j$ holds, iff $t_i \leq t_j - \Delta t$ and Δt is small nonnegative value. Therefore, as inspired by [33], but with further modifications to tolerate order switchings of alarms, a closed alarm sequence is defined as follows.

Definition 4. Alarm sequence $\mathcal{S}$ is a closed sequence, iff

1. $\mathcal{S}$ is frequent, i.e., $\sigma(\mathcal{S}) \geq \bar{\sigma}_{\mathcal{S}}$, and

2. $\mathcal{S}$ is closed, i.e., there exists no super sequence $\mathcal{S}'$, such that $\mathcal{S} \sqsubset \mathcal{S}'$ and $\sigma(\mathcal{S}') = \sigma(\mathcal{S})$.

Here, $\bar{\sigma}_{\mathcal{S}}$ is a user-defined threshold called minimum sequence support. By default, set $\bar{\sigma}_{\mathcal{S}} = \bar{\sigma}_{\mathcal{I}}$. The support of $\mathcal{S}$ is denoted as $\sigma(\mathcal{S})$, which indicates the occurrence frequency of $\mathcal{S}$ across the analyzed alarm floods. The operator $\sqsubset$ denotes that $\mathcal{S}$ is the subsequence of $\mathcal{S}'$, i.e., $\mathcal{S}'$ captures all alarm itemsets in $\mathcal{S}$ with identical sequential orders. More specifically, for $\forall \mathcal{I}_i, \mathcal{I}_j \in \mathcal{S}$, there exist $\mathcal{I}'_i, \mathcal{I}'_j \in \mathcal{S}'$, such that $\mathcal{I}_i = \mathcal{I}'_i$ and $\mathcal{I}_j = \mathcal{I}'_j$, while both $t_i \prec t_j$ and $t'_i \prec t'_j$ hold. $\square$

Accordingly, the problem in this study is then formulated as: given the dataset $\mathbb{F}$ of alarm flood sequences, the objective is to extract alarm sequential patterns of interest. The framework of the proposed method for the extraction of alarm flood patterns is summarized in Fig. 4.1, where the calculation is conducted in four major stages, including 1) the set-based pre-matching to facilitate calculation by excluding irrelevant alarm flood sequences, 2) the determination of closed alarm itemsets to give basic components of alarm floods by incorporating time stamps, 3) the discovery of closed alarm sequences

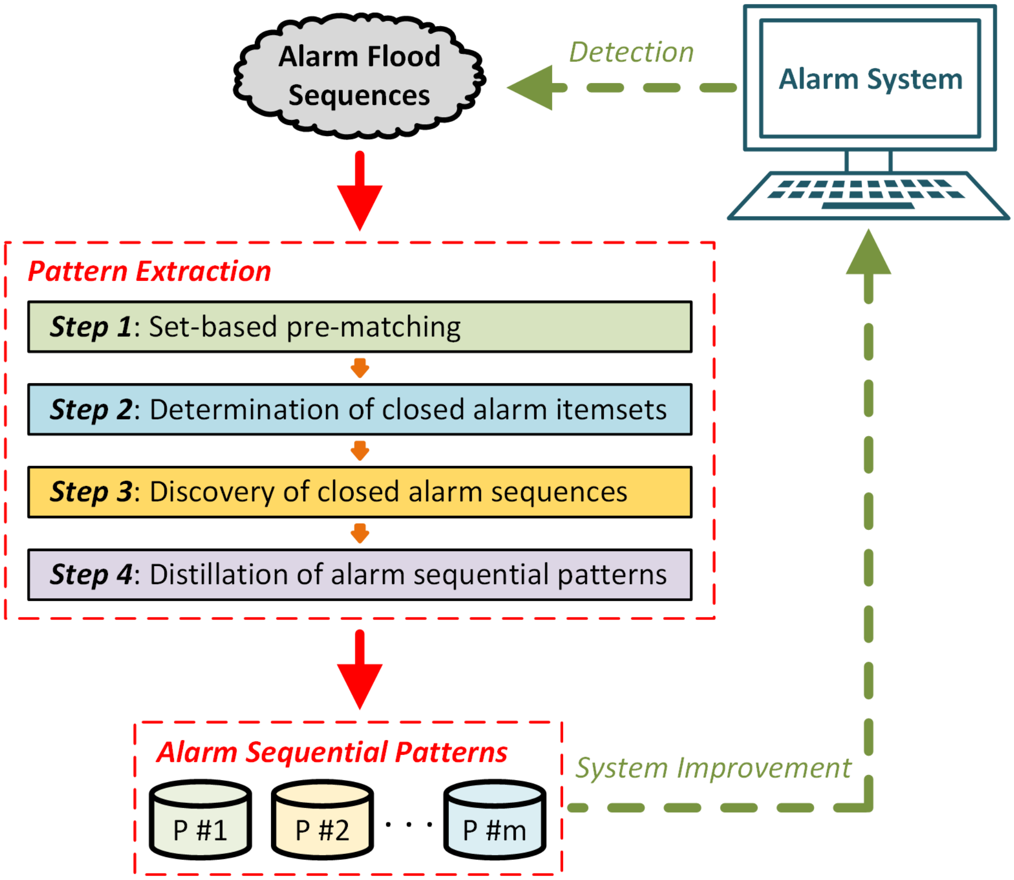

Figure 4.1: Framework of the proposed method to extract alarm sequential patterns.

to identify typical representations of alarm floods with the tolerance of alarm order switchings, and 4) the distillation of alarm sequential patterns to reduce the impact of irrelevant alarms and different lengths of alarm flood sequences. Detailed calculations and procedures regarding the four stages are presented in the next section.

4.3 Methodology

This section proposes a systematic method for the extraction of alarm sequential patterns. Four major steps are involved, including the set-based pre-matching, the determination of closed alarm itemsets, the discovery of closed alarm sequences, and the distillation of alarm sequential patterns.

4.3.1 Set-Based Pre-Matching

As the first step, a set-based pre-matching mechanism is exploited to separate alarm flood sequences into groups for pattern extraction and also exclude the irrelevant alarm floods. Given historical alarm flood sequences in $\mathbb{F}$, a binary matrix J is calculated with each element given by

$$J_{i,j} = \begin{cases} 1, & \text{if } |\Gamma(\mathcal{F}_i) \cap \Gamma(\mathcal{F}_j)| \geq L_{th}, \\ 0, & \text{otherwise}, \end{cases} \tag{4.2}$$

where $\Gamma(\mathcal{F}) = \{a | E = (a, t) \ \in \mathcal{F}\}$ gives the set of alarms contained in $\mathcal{F} \in \mathbb{F}$, and $J_{i,j}$ represents the element in the ith row and jth column of the matrix. $J_{i,j} = 1$, if $\mathcal{F}_i$ and $\mathcal{F}_j$ have enough common alarms; otherwise, $J_{i,j} = 0$. Here, L_{th} is a user-defined threshold called minimum pattern length. By default, a rule of thumb is to set the threshold $L_{th} = 5$, since the industrial standard ANSI/ISA-18.2 [49] defines the end of an alarm flood as when the alarm rate drops below 5 alarms over a period of 10 minutes.

Thereafter, by clustering the alarm floods with $J_{i,j} = 1$, alarm floods are separated into different collections; in each collection $\mathbb{F}_k = \{\mathcal{F}_1^k, \mathcal{F}_2^k, \cdots, \mathcal{F}_{|\mathbb{F}_k|}^k\}$, any two alarm floods $\mathcal{F}_i^k$ and $\mathcal{F}_j^k$ must have at least L_{th} alarms in common (i.e., $J_{i,j} = 1$). As a result, irrelevant sequences are excluded from pattern extraction for $\mathbb{F}_k$, such that unnecessary calculations can be avoided. Eventually, the set of alarm floods is divided into K collections, i.e., $\mathbb{F} = \bigcup_{k=1}^{K} \mathbb{F}_k$, where pattern extraction is recursively conducted for each collection of alarm floods. It is worth mentioning that the set-based pre-matching is robust to irrelevant alarms and different lengths of alarm flood sequences because the calculation only measures the number of common alarms. In addition, set-based pre-matching can help with the selection of proper values of $\bar{\sigma}_{\mathcal{I}}$ and $\bar{\sigma}_{\mathcal{S}}$ based on the number of alarm floods in each collection. Next, closed alarm itemsets are determined from each collection $\mathbb{F}_k$ of alarm floods by incorporating time stamps.

4.3.2 Determination of Closed Alarm Itemsets

For the determination of closed alarm itemsets, a list data structure (called itemset time-list) is utilized to represent the time stamps of alarms in an itemset. Specifically, given an alarm itemset $\mathcal{I}$, its itemset time-list is given by

$$\Psi(\mathcal{I}) = [\psi_1, \psi_2, \cdots, \psi_{|\mathbb{F}|}], \tag{4.3}$$

where $\psi_i = \{t_1, t_2, \cdots, t_{|\psi_i|}\}$ is a set composed of all time stamps corresponding to alarm occurrences of $\mathcal{I}$ in the ith alarm flood $\mathcal{F}_i \in \mathbb{F}$, which contains all alarms in $\mathcal{I}$; otherwise $\psi_i = \emptyset$, and $\emptyset$ denotes an empty set. Then, the support value of $\mathcal{I}$, namely, the number of alarm flood sequences containing $\mathcal{I}$, is calculated as the number of non-empty sets in $\Psi(\mathcal{I})$, i.e.,

$$\sigma(\mathcal{I}) = |\{\psi | \psi \neq \emptyset \text{ and } \psi \in \Psi(\mathcal{I})\}|. \tag{4.4}$$

For initialization, the values of $\mathcal{I}$ and $\Psi(\mathcal{I})$ are assigned based on alarm tag names and time stamps, respectively. Each alarm itemset is initialized as a set comprised by a single alarm, i.e., $\mathcal{I} = \{a\}$, $a \in \mathcal{A}$, and then $\psi_i \in \Psi(\mathcal{I})$ is

$$\psi_i = \{t | E = (a, t) \in \mathcal{F}_i, \mathcal{F}_i \in \mathbb{F}\}. \tag{4.5}$$

It is noteworthy that $\psi_i = \emptyset$, if $a \in \mathcal{A}$ is not in the ith alarm flood $\mathcal{F}_i$, $i = 1, 2, \cdots, |\mathbb{F}|$.

Thereafter, itemset extension is recursively performed to determine closed alarm itemset. Given two alarm itemsets $\mathcal{I}$ and $\mathcal{I}'$, an extended alarm itemset is obtained as

$$\tilde{\mathcal{I}} = \mathcal{I} \cup \mathcal{I}', \tag{4.6}$$

where $\cup$ stands for the set union. The itemset time-list of $\tilde{\mathcal{I}}$ is represented by $\Psi(\tilde{\mathcal{I}}) = [\tilde{\psi}_1, \tilde{\psi}_2, \cdots, \tilde{\psi}_{|\mathbb{F}|}]$ with each element obtained by

$$\tilde{\psi}_i = \psi_i \cap \psi_i', \tag{4.7}$$

where $\cap$ denotes set intersection, and ψ_i and ψ_i' are the ith sets in $\Psi(\mathcal{I})$ and $\Psi(\mathcal{I}')$, respectively, $i = 1, 2, \cdots, |\mathbb{F}|$. Then, it is determined that if $\tilde{\mathcal{I}}$ gives a

Algorithm 4 $\tilde{\mathcal{H}}_{\mathcal{I}} = g(\mathcal{H}_{\mathcal{I}}, \tilde{\mathcal{I}}, \bar{\sigma}_{\mathcal{I}})$: Update discovered closed alarm itemsets

Input Arguments: $\mathcal{H}_{\mathcal{I}}$, $\tilde{\mathcal{I}}$
Output Argument: $\tilde{\mathcal{H}}_{\mathcal{I}}$
if $\sigma(\tilde{\mathcal{I}}) \geq \bar{\sigma}_{\mathcal{I}}$ **then**
    if $\exists(\sigma \times \mathcal{N}_\sigma) \in \mathcal{H}_{\mathcal{I}}$, s.t. $\sigma = \sigma(\tilde{\mathcal{I}})$ **then**
        for $\mathcal{I} \in \mathcal{N}_\sigma$ **do**
            if $\tilde{\mathcal{I}} \subseteq \mathcal{I}$ and $\Psi(\tilde{\mathcal{I}}) = \Psi(\mathcal{I})$ **then**
                $\tilde{\mathcal{H}}_{\mathcal{I}} = \mathcal{H}_{\mathcal{I}}$, **return** ▷ Early termination
            else if $\mathcal{I} \subset \tilde{\mathcal{I}}$ and $\Psi(\mathcal{I}) = \Psi(\tilde{\mathcal{I}})$ **then**
                $\mathcal{H}_{\mathcal{I}} = \mathcal{H}_{\mathcal{I}} \ominus \mathcal{I}$
            end if
        end for
    end if
    $\tilde{\mathcal{H}}_{\mathcal{I}} = \mathcal{H}_{\mathcal{I}} \oplus \tilde{\mathcal{I}}$
else
    $\tilde{\mathcal{H}}_{\mathcal{I}} = \mathcal{H}_{\mathcal{I}}$
end if

closed alarm itemset based on a hash table, which is comprised by a collection of key-value pairs as $\mathcal{H}_{\mathcal{I}} = (\sigma \times \mathcal{N}_\sigma)$, where the key is σ and the value is $\mathcal{N}_\sigma$ that contains discovered closed alarm itemsets having the same support σ. The procedures to update hash table are summarized by Algorithm 4, which is inspired by [33] but further revised with set operations to facilitate the calculation. In Algorithm 4, $\mathcal{H}_{\mathcal{I}}$ ($\tilde{\mathcal{H}}_{\mathcal{I}}$) represents the existing (updated) hash table, which stores the discovered closed alarm itemsets. The operator $\oplus$ ($\ominus$) means to update hash table by adding (removing) alarm itemset into (from) $\mathcal{N}_\sigma$. The condition $\Psi(\tilde{\mathcal{I}}) = \Psi(\mathcal{I})$ holds, iff $\tilde{\psi}_i = \psi_i$, $\forall i = 1, 2, \cdots, |\mathbb{F}|$.

Itemset extension is recursively performed until no more closed alarm itemsets could be discovered. Then, the obtained results are collected into a set as $\mathbb{C} = \{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_{|\mathbb{C}|}\}$, where $\mathcal{I}_i$ is the ith closed alarm itemset, $i = 1, 2, \cdots, |\mathbb{C}|$. To implement such recursive calculation for itemset extension, the depth-first strategy in [33] is adapted in this work. As a result, given $\mathbb{F}$ and $\bar{\sigma}_{\mathcal{I}}$, closed alarm itemsets are determined and collected into a set by

$$\mathbb{C} = f_{\mathcal{I}}(\mathbb{F}, \bar{\sigma}_{\mathcal{I}}), \tag{4.8}$$

where $f_{\mathcal{I}}(\cdot)$ denotes the function to determine closed alarm itemsets. Further, closed alarm sequences are discovered by sequentially assembling the obtained closed alarm itemsets based on their time stamps.

4.3.3 Discovery of Closed Alarm Sequences

An alarm sequence is comprised by a collection of closed alarm itemsets that are sequentially ordered by time stamps. Thus, a sequence time-list is used. Given an alarm sequence $\mathcal{S} = \langle \mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_{|\mathcal{S}|} \rangle$, its sequence time-list is given by

$$\Omega(\mathcal{S}) = [\omega_1, \omega_2, \cdots, \omega_{|\mathbb{F}|}], \tag{4.9}$$

where $\omega_i = \langle t_1, t_2, \cdots, t_{|\mathcal{S}|} \rangle$ is a sequence of time stamps corresponding to the closed alarm itemsets of $\mathcal{S}$ in the ith alarm flood, $i = 1, 2, \cdots, |\mathbb{F}|$. For instance, given that the jth closed alarm itemset $\mathcal{I}_j \in \mathcal{S}$, $j = 1, 2, \cdots, |\mathcal{S}|$, exists in alarm flood $\mathcal{F}_i \in \mathbb{F}$ at time instant $\tilde{t}$, the time stamp is then recorded in ω_i with $t_j = \tilde{t}$. For initialization, the values of $\mathcal{S}$ and $\Omega(\mathcal{S})$ are determined based on $\mathcal{I}$ and $\Psi(\mathcal{I})$, respectively. Each alarm sequence is initialized with a single closed alarm itemset, i.e., $\mathcal{S} = \langle \mathcal{I} \rangle$, $\mathcal{I} \in \mathbb{C}$. Then, $\omega_i \in \Omega(\mathcal{S})$ is obtained as

$$\omega_i = \begin{cases} \min(\psi_i), \text{ if } \psi_i \neq \emptyset, \psi_i \in \Psi(\mathcal{I}), \\ \infty, \text{ otherwise}, \end{cases} \tag{4.10}$$

where ∞ denotes an empty element in sequence time-list. The support of $\mathcal{S}$ is calculated as

$$\sigma(\mathcal{S}) = |\{\omega | \omega \neq \infty \text{ and } \omega \in \Omega(\mathcal{S})\}|, \tag{4.11}$$

which indicates the occurrence frequency of $\mathcal{S}$, namely, how many alarm floods contain $\mathcal{S}$.

Thereafter, closed alarm sequences are discovered by sequence extension, which is to recursively attach a closed alarm itemset to the end of an alarm sequence. Given an alarm sequence $\mathcal{S} = \langle \mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_{|\mathcal{S}|} \rangle$ and a closed alarm itemset $\mathcal{I} \in \mathbb{C}$, an extended alarm sequence $\tilde{S}$ is obtained by

$$\tilde{\mathcal{S}} = \mathcal{S} \sqcup \mathcal{I} = \langle \mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_{|\mathcal{S}|}, \mathcal{I} \rangle, \tag{4.12}$$

where the operator $\sqcup$ indicates merging an element at the end of a sequence. The sequence time-list of $\tilde{\mathcal{S}}$ is given by $\Omega(\tilde{\mathcal{S}}) = [\tilde{\omega}_1, \tilde{\omega}_2, \cdots, \tilde{\omega}_{|\mathbb{F}|}]$. To achieve relaxed comparison of time stamps such that order switchings of alarms could be tolerated in sequence extension, $\tilde{\omega}_i$ is calculated by

$$\tilde{\omega}_i = \begin{cases} \omega_i \sqcup \min(\theta), \text{ if } \theta \neq \emptyset, \\ \infty, \text{ otherwise}, \end{cases} \tag{4.13}$$

where $\theta = \{t | t \geq \max(\omega_i) - \Delta t, t \in \psi_i, \psi_i \in \Psi(\mathcal{I})\}$ gives the set of time stamps associated with $\mathcal{I}$ for sequence extension. Depending on θ, the value of $\tilde{\omega}_i$ is assigned with two options: if $\theta \neq \emptyset$, i.e., the sequence extension is feasible, $\tilde{\omega}_i = \omega_i \sqcup \min(\theta) = \langle t_1, t_2, \cdots, t_{|\mathcal{S}|}, \min(\theta) \rangle$; otherwise, $\tilde{\omega}_i = \infty$, which indicates that $\tilde{\mathcal{S}}$ does not exist in alarm flood $\mathcal{F}_i \in \mathbb{F}$, $i = 1, 2, \cdots, |\mathbb{F}|$. The values of ψ_i and ω_i are determined using the ith elements in $\Psi(\mathcal{I})$ and $\Omega(\mathcal{S})$, respectively. Here, Δt is a user-defined threshold to tolerate alarm order switchings and it should be a small nonnegative value. A smaller Δt leads to reduced tolerance of alarm order switchings across different alarm floods; given $\Delta t = 0$, alarm sequential patterns are extracted with exact alarm occurrence orders. By contrast, a larger Δt increases such tolerance but also enlarges the search space for pattern extraction.

After sequence extension, it is decided that if $\tilde{\mathcal{S}}$ gives a closed alarm sequence based on a hash table, which is comprised by a collection of key-value pairs for support values and the corresponding closed alarm sequences. The detailed procedures are summarized in Algorithm 5, where $\mathcal{H}_{\mathcal{S}}$ ($\tilde{\mathcal{H}}_{\mathcal{S}}$) denotes the existing (updated) hash table. Moreover, to update the hash table, the sequential orders of alarms should be further considered, and thus the condition $\Omega(\tilde{\mathcal{S}}) \sqsubseteq \Omega(\mathcal{S})$ holds, iff $\tilde{\omega}_i \sqsubseteq \omega_i$, $\forall i = 1, 2, \cdots, |\mathbb{F}|$.

Sequence extension is recursively performed until no more closed alarm sequences could be discovered. Then, the obtained results are put into a set as $\mathbb{S} = \{\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_{|\mathbb{S}|}\}$, where $\mathcal{S}_i$ is the ith closed alarm sequence, $i = 1, 2, \cdots, |\mathbb{S}|$. Such recursive calculation for the discovery of closed alarm sequences is implemented by the depth-first strategy. Therefore, given $\mathbb{C}$ and

Algorithm 5 $\tilde{\mathcal{H}}_{\mathcal{S}} = h(\mathcal{H}_{\mathcal{S}}, \tilde{\mathcal{S}}, \bar{\sigma}_{\mathcal{S}})$: Update discovered closed alarm sequences

Input Arguments: $\mathcal{H}_{\mathcal{S}}$, $\tilde{\mathcal{S}}$
Output Argument: $\tilde{\mathcal{H}}_{\mathcal{S}}$
if $\sigma(\tilde{\mathcal{S}}) \geq \bar{\sigma}_{\mathcal{S}}$ **then**
  if $\exists(\sigma \times \mathcal{N}_{\sigma}) \in \mathcal{H}_{\mathcal{S}}$, s.t. $\sigma = \sigma(\tilde{\mathcal{S}})$ **then**
    for $\mathcal{S} \in \mathcal{N}_{\sigma}$ **do**
      if $\tilde{\mathcal{S}} \sqsubseteq \mathcal{S}$ and $\Omega(\tilde{\mathcal{S}}) \sqsubseteq \Omega(\mathcal{S})$ **then**
        $\tilde{\mathcal{H}}_{\mathcal{S}} = \mathcal{H}_{\mathcal{S}}$, **return** ▷ Early termination
      else if $\mathcal{S} \sqsubset \tilde{\mathcal{S}}$ and $\Omega(\mathcal{S}) \sqsubset \Omega(\tilde{\mathcal{S}})$ **then**
        $\mathcal{H}_{\mathcal{S}} = \mathcal{H}_{\mathcal{S}} \ominus \mathcal{S}$
      end if
    end for
  end if
  $\tilde{\mathcal{H}}_{\mathcal{S}} = \mathcal{H}_{\mathcal{S}} \oplus \tilde{\mathcal{S}}$
else
  $\tilde{\mathcal{H}}_{\mathcal{S}} = \mathcal{H}_{\mathcal{S}}$
end if

$\bar{\sigma}_{\mathcal{S}}$, closed alarm sequences are discovered and collected into a set by

$$\mathbb{S} = f_{\mathcal{S}}(\mathbb{C}, \bar{\sigma}_{\mathcal{S}}), \tag{4.14}$$

where $f_{\mathcal{S}}(\cdot)$ is the function to discover closed alarm sequences. It is noteworthy that in the obtained results, there may exist some closed alarm sequences that resemble each other with most alarms the same and only a few alarms distinct, leading to redundancy of extracted patterns. Thus, it is necessary to merge such similar closed alarm sequences to obtain more compact results. This is achieved by a pattern distillation step in the next subsection.

4.3.4 Distillation of Alarm Sequential Patterns

The closed alarm sequences collected in $\mathbb{S}$ should be further distilled, so as to reduce the redundancy of the obtained results and export compact alarm sequential patterns. The distillation is conducted by the following three major steps.

First, the groups of closed alarm sequences for pattern distillation are identified based on the proportion of shared common alarms. For this purpose, a binary matrix Ξ is calculated with each element given by

$$\Xi_{i,j} = \begin{cases} 1, & \text{if } \rho(\mathcal{S}_i, \mathcal{S}_j) \geq \bar{\rho}, \\ 0, & \text{otherwise}, \end{cases} \tag{4.15}$$

where $\bar{\rho}$ denotes a user-defined threshold for pattern distillation and $\Xi_{i,j}$ stands for the element in the ith row and jth column of the matrix. Function $\rho(\cdot)$ measures the proportion of identical alarms between $\mathcal{S}_i$ and $\mathcal{S}_j$ in $\mathbb{S}$, where the compared alarms are taken as identical, iff they have the same tag names and equal time stamps. Therefore, $\rho(\mathcal{S}_i, \mathcal{S}_j)$ is calculated by

$$\rho(\mathcal{S}_i, \mathcal{S}_j) = \frac{|\mathcal{X}(\mathcal{S}_i, \mathcal{S}_j)\}|}{\min(|\Phi(\mathcal{S}_i)|, |\Phi(\mathcal{S}_j)|)}, \tag{4.16}$$

where $\min(\cdot)$ gives the minimum of input values. $\Phi(\mathcal{S}) = \{a|\tilde{E} = (a, \Gamma(a)) \in \mathcal{S}\}$ stands for a set captures all alarms in $\mathcal{S} \in \mathbb{S}$ and $\Gamma(a) = [\gamma_1, \gamma_2, \cdots, \gamma_{|\mathbb{F}|}]$ denotes the time-list of alarm a. The value of $\gamma_i \in \Gamma(a)$ is determined based on $\omega_i \in \Omega(\mathcal{S})$, i.e., γ_i is assigned with the time stamp of $\mathcal{I} \in \mathcal{S}$ that is found from ω_i, where $a \in \mathcal{I}$, $i = 1, 2, \cdots, |\mathbb{F}|$. $\mathcal{X}(\mathcal{S}_i, \mathcal{S}_j) = \{\tilde{E}^p = \tilde{E}^q | \tilde{E}^p = (a^p, \Gamma(a^p)) \in \mathcal{S}_i, \tilde{E}^q = (a^q, \Gamma(a^q)) \in \mathcal{S}_j\}$ identifies the identical alarms between $\mathcal{S}_i$ and $\mathcal{S}_j$. It should be noticed that the condition $\tilde{E}^p = \tilde{E}^q$ holds, iff $a^p = a^q$ and $\gamma_i^p = \gamma_i^q$, $\forall i = 1, 2, \cdots, |\mathbb{F}|$, where $\gamma_i^p \in \Gamma(a^p)$, $\gamma_i^q \in \Gamma(a^q)$.

Second, by clustering closed alarm sequences with $\Xi_{i,j} = 1$, the collections of closed alarm sequences for pattern distillation are identified. Such collections are put into a set as

$$\mathbb{Z} = \{\mathcal{Z}_1, \mathcal{Z}_2, \cdots, \mathcal{Z}_{|\mathbb{Z}|}\}, \tag{4.17}$$

where $\mathcal{Z}_n = \{\mathcal{S}_1^n, \mathcal{S}_2^n, \cdots \mathcal{S}_{|\mathcal{Z}_n|}^n\}$ is the nth collection of closed alarm sequences, $n = 1, 2, \cdots, |\mathbb{Z}|$. For any two closed alarm sequences $\mathcal{S}_i^n$ and $\mathcal{S}_j^n$ in $\mathcal{Z}_n$ they are similar in the captured alarms, i.e., $\Xi_{i,j} = 1$.

Last, a compact alarm sequential pattern is distilled from $\mathcal{Z}_n$ by combining the corresponding closed alarm sequences as $\mathcal{P}_n = \mathcal{S}_1^n \uplus \mathcal{S}_2^n \uplus \cdots \mathcal{S}_{|\mathcal{Z}_n|}^n$, where $\uplus$ indicates the distillation of compact alarm sequential pattern, i.e., the captured alarms are sequentially ordered based on their average time stamps.

Here, the compact alarm sequential pattern is given by

$$\mathcal{P}_n = \langle \bar{E}_1, \bar{E}_2, \cdots, \bar{E}_{|\mathcal{P}_n|} \rangle, \tag{4.18}$$

where $\bar{E}_j = (a, \bar{t})$ stands for the jth captured alarm event in $\mathcal{P}_n$, $j = 1, 2, \cdots, |\mathcal{P}_n|$, and the tuple gives tag name a and average time $\bar{t}$. More specifically, the average time stamp is calculated by

$$\bar{t} = f_m(\Upsilon(a)), \tag{4.19}$$

where $f_m(\cdot)$ calculates the mean of the inputs and $\Upsilon(a) = \{t|\mathcal{E} = (a, t, r(\mathcal{F})) \in \mathbb{E}(a)\}$ gives a set of proper time stamps to determine $\bar{t}$. Here, $\mathbb{E}(a) = \{\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_{|\mathbb{E}|}\}$ represents the set of historical occurrences for alarm a, where $\mathcal{E} = (a, t, r(\mathcal{F}))$ is a tuple comprised by tag name a, time stamp t, and alarm flood index $r(\mathcal{F})$ (namely, a numerical label to uniquely identify an alarm flood). The value of $\mathcal{E}$ is assigned based on $\Gamma(a)$ using the associated time stamps and alarm floods. It is satisfied that $\forall \mathcal{E}_i, \mathcal{E}_j \in \mathbb{E}(a)$, they have identical tag names as a and similar sequential orders, i.e., $\{t|t_i - \Delta t \leq t \leq t_i + \Delta t\} \cap \{t|t_j - \Delta t \leq t \leq t_j + \Delta t\} \neq \emptyset$, where t_i (t_j) is the time stamp of $\mathcal{E}_i$ ($\mathcal{E}_j$). In addition, to ensure that $\mathcal{E}$ is correctly determined, alarm time stamps should be properly shifted based on an anchor point, which could be selected as the time stamp of an alarm that occurs across the closed alarm sequences to be distilled. In practice, such anchor point is not hard to find as the associated closed alarm sequences involve many identical alarms.

Eventually, all obtained compact alarm sequential patterns are collected into a set as

$$\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_{|\mathbb{Z}|}\}, \tag{4.20}$$

where pattern $\mathcal{P}_n$ is distilled from $\mathcal{Z}_n$, $n = 1, 2, \cdots, |\mathbb{Z}|$. It is noteworthy that the distillation is based on closed alarm sequences rather than original alarm floods, such that the impact due to irrelevant alarms and different lengths of alarm flood sequences are avoided.

Algorithm 6 $\mathbb{P} = f(\mathbb{F}, \lambda)$: Alarm Flood Pattern Extraction

1: Input Arguments: $\mathbb{F}$, λ
2: Output Argument: $\bar{\mathbb{P}}$
3: Divide $\mathbb{F}$ into K collections of alarm floods as $\mathbb{F}_k$, $k = 1, 2, \cdots, K$, by set-based pre-matching
4: **for** $\mathbb{F}_k \subset \mathbb{F}$ **do**
5: Calculate $\mathbb{C}_k = f_{\mathcal{I}}(\mathbb{F}_k, \bar{\sigma}_{\mathcal{I}})$ by eqn. (4.8) to determine closed alarm itemsets
6: Calculate $\mathbb{S}_k = f_{\mathcal{S}}(\mathbb{C}_k, \bar{\sigma}_{\mathcal{S}})$ by eqn. (4.14) to discover closed alarm sequences
7: **for** $\mathcal{S}_i, \mathcal{S}_j \in \mathbb{S}_k$ **do**
8: Calculate $\Xi_{i,j}$ by eqn. (4.15)
9: **end for**
10: Determine $\mathbb{Z} = \{\mathcal{Z}_1, \mathcal{Z}_2, \cdots, \mathcal{Z}_{|\mathbb{Z}|}\}$ by eqn. (4.17)
11: Calculate $\mathcal{P}_n$ by eqn. (4.18), $\forall \mathcal{Z}_n \in \mathbb{Z}$
12: Get $\mathbb{P}_k = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_{|\mathbb{Z}|}\}$ by eqn. (4.20)
13: **end for**
14: Obtain $\bar{\mathbb{P}} = \bigcup_{k=1}^{K} \mathbb{P}_k$ by eqn. (4.21)

4.3.5 Implementation Procedures

To improve calculation efficiency, the extraction of alarm sequential patterns is conducted based on the alarm flood collections determined by set-based pre-matching. Given the kth collection $\mathbb{F}_k \subset \mathbb{F}$, $k = 1, 2, \cdots, K$, compact alarm sequential patterns are extracted, and then collected into a set $\mathbb{P}_k$ by eqn. (4.20). Eventually, after recursively performing such calculations for pattern extraction, the set collecting all discovered alarm sequential patterns is obtained as

$$\bar{\mathbb{P}} = \bigcup_{k=1}^{K} \mathbb{P}_k. \tag{4.21}$$

Detailed procedures of the proposed method for alarm flood pattern extraction are summarized in Algorithm 6, where the inputs are alarm floods in dataset $\mathbb{F}$ and the vector $\lambda = [\bar{\sigma}_{\mathcal{I}}, \bar{\sigma}_{\mathcal{S}}, \Delta t, \bar{\rho}, L_{th}]$ contains all predefined parameters; the output is $\bar{\mathbb{P}}$ giving the set of extracted compact alarm sequential patterns. Specifically, the calculation is performed as follows:

1. Set-based pre-matching is conducted in line 3 to separate alarm flood

sequences into multiple collections.

2. Closed alarm itemsets are determined in line 5 by recursively calling Algorithm 1.

3. Closed alarm sequences are discovered in line 6 by recursively calling Algorithm 2.

4. Compact alarm sequential patterns are distilled for each collection of alarm floods in lines 7-12.

As a result, all extracted compact alarm sequential patterns are collected into $\bar{\mathbb{P}}$.

4.4 Industrial Case Study

In this section, an industrial case study is presented to demonstrate the effectiveness of the proposed method based on real A&E data.

4.4.1 Overall Results

The A&E data was collected from a large-scale industrial process, over a long period of one year and eight months. First, alarm floods were identified from the data based on the definition in industrial standard ANSI/ISA-18.2 [49]. Off-delay timers were applied to reduce chattering alarms prior to alarm flood extraction. In total, 178 alarm floods were extracted, involving 933 alarm tags. To distinguish these alarm floods, unique numerical labels were assigned as $1, 2, \cdots, 178$. Then, 32 collections of alarm floods were obtained by set-based pre-matching. Next, alarm flood pattern extractions were recursively performed for each collection of alarm floods. As a result, 34 compact alarm sequential patterns were obtained.

It is worth mentioning that the search space for alarm flood pattern extraction was effectively reduced by the set-based pre-matching, such that the maximum number of alarms to be examined for the discovery of an alarm flood pattern was reduced to 36. Without set-based pre-matching, all the 933 alarms would have to be examined together, causing many redundant alarm combinations. Therefore, the proposed method is capable of extracting alarm sequential patterns based on very low support values (e.g., pairwise alignment of alarm flood sequences, i.e., $\bar{\sigma}_{\mathcal{I}} = 2$ and $\bar{\sigma}_{\mathcal{S}} = 2$).

A summary of the discovered closed alarm itemsets and sequences is shown in Fig. 4.2, where these alarm itemsets and sequences are intermediate results for alarm flood pattern extraction. In this figure, the numerical indexes on the horizontal axis stand for different collections of alarm floods, while the vertical axis presents the numbers of closed alarm itemsets, closed alarm sequences, and compact alarm sequential patterns. Based on the determination of closed alarm itemsets, complicated alarm combinations were reduced in the latter calculation stage for the discovery of closed alarm sequences. For example, in alarm flood collections #16, #27, and #30, alarm sequential patterns were effectively extracted by getting closed alarm itemsets, as all alarms in the corresponding flood sequences were triggered simultaneously. As a result, the calculations to discover closed alarm sequences were totally avoided. Eventually, by pattern distillation to significantly reduce pattern redundancy, there were only 34 compact alarm sequential patterns, whereas without pattern distillation, there were 109 closed alarm sequences, among which many of them were similar. Thus, the pattern distillation step is necessary to avoid redundant results and output meaningful compact alarm sequential patterns.

4.4.2 Extracted Alarm Flood Patterns

For further demonstration, three alarm sequential patterns extracted using the proposed method are presented as follows.

Alarm Flood Pattern #1

This pattern was extracted from four alarm floods, namely, #117, #137, #161, and #173. Based on expert evaluation with process knowledge, this pattern was caused by the control power failure. The details associated with

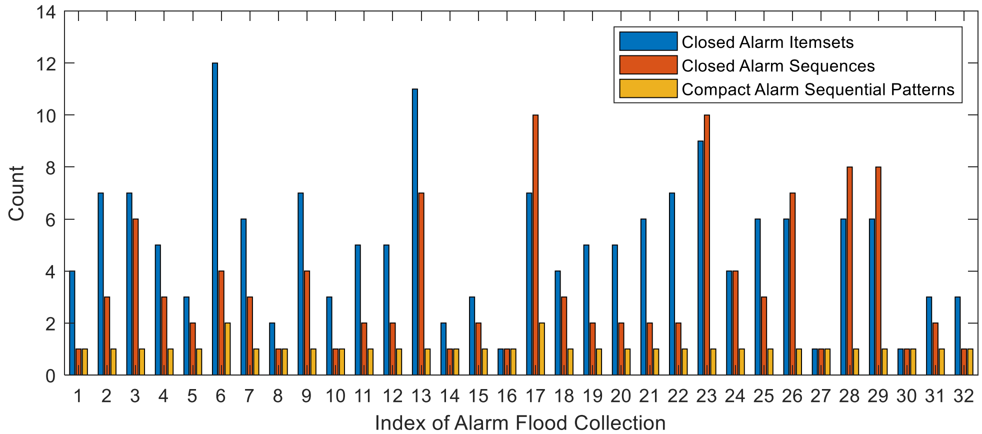

Figure 4.2: The numbers of closed alarm itemsets, closed alarm sequences, and compact alarm sequential patterns obtained from the 32 collections of alarm flood sequences.

Subsequences in Original Alarm Floods

Alarm Flood #117	
Time	Alarm Tag
23:24:01	Tag57.ACTIVE
23:24:01	Tag55.ACTIVE
23:24:01	Tag63.ACTIVE
23:24:01	Tag62.ACTIVE
23:24:03	Tag59.ACTIVE
23:24:03	Tag60.ACTIVE
23:24:03	Tag61.ACTIVE
23:24:03	Tag56.ACTIVE
23:24:26	Tag55.CLEARED
23:24:26	Tag63.CLEARED
23:24:26	Tag62.CLEARED
23:24:26	Tag53.ACTIVE
23:24:38	Tag56.CLEARED
23:33:32	Tag55.ACTIVE
23:33:32	Tag56.ACTIVE

Alarm Flood #137	
Time	Alarm Tag
7:20:30	Tag59.ACTIVE
7:20:30	Tag60.ACTIVE
7:20:30	Tag61.ACTIVE
7:20:30	Tag55.ACTIVE
7:20:30	Tag56.ACTIVE
7:20:30	Tag63.ACTIVE
7:20:30	Tag62.ACTIVE
7:20:41	Tag54.ACTIVE
7:20:53	Tag55.CLEARED
7:20:53	Tag63.CLEARED
7:20:53	Tag62.CLEARED
7:20:53	Tag53.ACTIVE

Alarm Flood #161	
Time	Alarm Tag
3:46:18	Tag59.ACTIVE
3:46:18	Tag60.ACTIVE
3:46:18	Tag61.ACTIVE
3:46:18	Tag57.ACTIVE
3:46:18	Tag55.ACTIVE
3:46:18	Tag56.ACTIVE
3:46:18	Tag63.ACTIVE
3:46:18	Tag62.ACTIVE
3:46:27	Tag54.ACTIVE
3:46:45	Tag55.CLEARED
3:46:45	Tag63.CLEARED
3:46:45	Tag62.CLEARED
3:46:45	Tag53.ACTIVE
3:46:52	Tag56.CLEARED

Alarm Flood #173	
Time	Alarm Tag
12:50:12	Tag59.ACTIVE
12:50:12	Tag60.ACTIVE
12:50:16	Tag61.ACTIVE
12:50:16	Tag57.ACTIVE
12:50:16	Tag55.ACTIVE
12:50:16	Tag56.ACTIVE
12:50:16	Tag63.ACTIVE
12:50:16	Tag62.ACTIVE
12:50:38	Tag55.CLEARED
12:50:38	Tag63.CLEARED
12:50:38	Tag62.CLEARED
12:50:38	Tag53.ACTIVE

Pattern Extraction

Extracted Alarm Sequential Pattern

Alarm Tag	Time (sec)	Alarm Description
Tag59.ACTIVE	0.0	Unit 1 Control Power Fail
Tag60.ACTIVE	0.0	Unit 2 Control Power Fail
Tag55.ACTIVE	0.5	Unit Valves in Local
Tag62.ACTIVE	0.5	Discharge Valve in Local
Tag63.ACTIVE	0.5	Suction Valve in Local
Tag57.ACTIVE	0.8	Unit Trouble
Tag56.ACTIVE	1.0	Pressure Control Valve Fail
Tag61.ACTIVE	1.0	Unit 3 Control Power Fail
Tag53.ACTIVE	24.8	Generator On
Tag55.CLEARED	24.8	Unit Valves in Local
Tag62.CLEARED	24.8	Discharge Valve in Local
Tag63.CLEARED	24.8	Suction Valve in Local

Figure 4.3: Alarm flood pattern #1 was extracted from alarm floods #117, #137, #161, and #173. This pattern was caused by the power failure.

this pattern are shown in Fig. 4.3, where the subsequences in original alarm floods are provided in the dashed red rectangle and the captured alarms by this pattern are highlighted in blue. There are three columns describing the extracted alarm sequential pattern, including the alarm tag, average time stamp, and alarm description. For better presentation, the average time stamps were shifted such that alarms occurred at the beginning of a sequential pattern have their average time stamps as 0's. To reduce the impact of irrelevant alarms, it is required that the captured alarms in a pattern should have occurred at least in three alarm flood sequences.

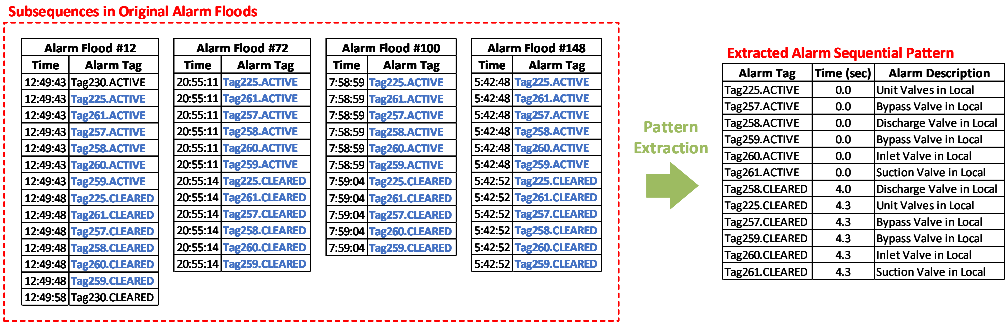

Figure 4.4: Alarm flood pattern #2 was extracted from alarm floods #12, #72, #100, and #148. This pattern was caused by the unusual valve operation mode.

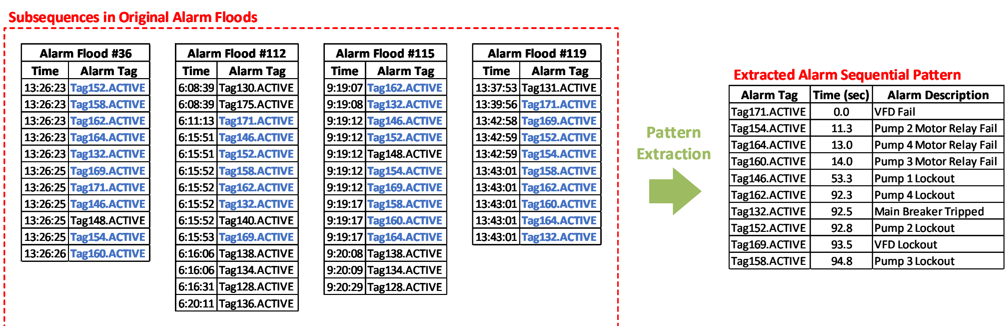

Figure 4.5: Alarm flood pattern #3 was extracted from alarm floods #36, #112, #115, and #119. This pattern was caused by the pump lockout.

In this pattern, the three alarms “Tag59.ACTIVE”, “Tag60.ACTIVE”, and “Tag61.ACTIVE” were directly associated with the control power failure, while the remaining ones were consequential alarms of this abnormality. More specifically, once the control power was lost, the alarms for “active” status of valves (e.g., “Tag55.ACTIVE”) were triggered. About 25 seconds later, a generator was automatically turned on as the backup electrical power source, and thus the alarms for “cleared” status of valves (e.g., “Tag55.CLEARED”) were annunciated, indicating that the process was recovering from this abnormality. Therefore, this alarm flood pattern is meaningful as it effectively reveals the associated abnormality.

Alarm Flood Pattern #2

This pattern was extracted from four alarm floods, namely, #12, #72, #100, and #148. The details of this pattern are shown in Fig. 4.4. Based on expert evaluation with process knowledge, this pattern was caused by the unusual valve operation mode. In this pattern, the captured alarms, namely, "Tag225.ACTIVE", "Tag257.ACTIVE", "Tag258.ACTIVE", "Tag259.ACTIVE", "Tag260.ACTIVE", and "Tag261.ACTIVE", were triggered simultaneously because their associated valves were changed to unusual operation modes. About 4 seconds later, such valves were switched back to the normal modes, and thus triggered the remaining alarms in this pattern. Therefore, this pattern is meaningful because all alarms associated with the abnormality are captured.

Alarm Flood Pattern #3

This pattern was extracted from four alarm floods, namely, #36, #112, #115, and #119. The details associated with this pattern are shown in Fig. 4.5. Based on expert evaluation with process knowledge, this pattern was caused by the pump lockout. As the pumps were controlled by a shared Variable Frequency Drive (VFD), the lockout abnormality happened in one pump could quickly propagate, leading to more lockout of pumps in the same process unit. Therefore, this pattern is meaningful; based on the result, a dynamic alarm suppression module could be configured to suppress the associated alarm floods.

4.5 Summary

To facilitate the analysis of alarm floods by discovering alarm sequential patterns, this chapter proposes a systematic alarm flood pattern extraction method, which is comprised by four major calculation stages, namely, the set-based pre-matching, the determination of closed alarm itemsets, the discovery of closed alarm sequences, and the distillation of compact alarm sequential patterns. To demonstrate the effectiveness of the proposed method, case studies were presented based on alarm data from a complex industrial facility. As shown in the obtained results, compact alarm sequential patterns were effectively extracted with the incorporation of time stamps and tolerance of alarm order switchings. The extracted results were validated to be meaningful based on expert evaluation with process knowledge.

Chapter 5

Conclusions and Future Work

In this chapter, remarks are provided to conclude this thesis, and then some potential research directions are pointed out for future work.

5.1 Conclusions

This thesis proposes a variety of data-driven methods for alarm monitoring and alarm flood pattern extraction, aiming at improving alarm system performance of complex industrial facilities. The outcomes of the studies in this thesis are summarized as follows:

1. A new method for monitoring equipment start-up operations is developed. The method is comprised of an offline design stage and an online monitoring stage. In the offline stage, the maximum delay time for alarm unsuppression is determined and dynamic alarm limits are formulated based on the data associated with normal start-up operations. In the online stage, a systematic algorithm is proposed to achieve effective alarm monitoring based on the dynamic alarm limits and online calculation of the unsuppression delay time. The proposed method is capable of preventing nuisance alarms as well as achieving effective alarm monitoring for equipment start-up operations.

2. A systematic pattern matching method is proposed to compare alarm floods across different processes. The method consists of two major

steps, including the generalization of alarm representations based on the distillation of key words from textural alarm attributes and reconstruction of abstracted alarm descriptors, and the similarity analysis to compare alarm floods across different processes. Based on this method, the efficiency of alarm flood analysis can be significantly improved as clusters of similar floods are discovered to help find common root causes, which lead to generalized solutions to address alarm floods from different processes all at once.

3. A new method to extract compact alarm flood patterns from historical alarm flood data is proposed with the incorporation of time stamps, tolerance of alarm order switchings, and a distillation strategy. The proposed method is capable of effectively extracting compact alarm sequential patterns while avoiding the influences of order switchings caused by small time differences on pattern extraction, and also minimizing the redundancy of extracted alarm sequential patterns.

The effectiveness and applicability of the proposed methods are validated by case studies using alarm data from complex industrial facilities.

5.2 Future Work

For the improvement of alarm system performance, the thesis proposed effective methods to address nuisance alarms and alarm floods, which are two major practical problems for alarm management. However, alarm management is sophisticated and time-consuming, as it is usually associated with large-scale industrial facilities. Consequently, various problems and unexpected challenges may arise in the lifecycle of alarm management. For instance, the design and configuration of alarm systems become critical for new industrial facilities, whereas maintenance and improvement are more emphasized for existing alarm systems. In addition, with the aging of hardware, some equipment should be updated or replaced, introducing another layer of complexity for alarm management, especially during the transitions for hardware replacement. To meet the demands in the complete lifecycle of alarm management, the following promising directions deserve efforts for future work.

Alarm Monitoring for System with Complex Dynamics

The basic but critical functionality of alarm systems is to achieve effective process minoring for complex industrial facilities. However, the operation of industrial processes involves many dynamics, such as start-up/shut-down, change of operation mode, and manufacture of products in different types. Such dynamics are usually complicated and could even vary case by case. In such situations, alarms configured based on steady state could fail, causing many nuisance alarms and even alarm floods. Chapter 2 in this thesis only studies the problem of alarm monitoring on the level of single equipment, namely, pump start-up operations. In practice, there exist many other kinds of state transitions with increased complexities, such as setpoint changes and feed property changes. Such situations may also present similar alarm monitoring problems deserving further studies. Therefore, for effective alarm monitoring to cope with system dynamics, the future work includes various extensions of the proposed method to more general and complicated cases. In addition, as a plant could be operated under different modes, the causal relationships for operator actions and alarms may change accordingly, leading to the opportunities to design some advanced logic-based alarm suppression rules for each operation mode. Moreover, for the discovery of optimal operation strategies to mitigate various alarm situations, work flow models could be established based on historical alarm data and the corresponding operator responses.

Extension and Generalization of Alarm Flood Analysis

Data-driven methods are proposed in Chapters 3 and 4 to generalize and facilitate the extraction of alarm flood patterns and such studies could be further generalized with extended applicability to different alarm management scenarios. The analysis of alarm floods is sophisticated and time-consuming, and one major reason is that the obtained alarm flood patterns have to be analyzed manually based on expert evaluations with process knowledge to identify the root causes and then design corresponding solutions. To help with such analysis with improved efficiency and accuracy, data-driven methods for the validation and labeling of obtained alarm flood patterns with root causes could be studied. For instance, to identify potential solutions to handle alarm floods, operator responses could be incorporated for alarm flood pattern extraction as a means of validation based on historical operations. Moreover, studies could be conducted to obtain databases with labeled alarm floods, leading to opportunities for advanced online applications, such as prediction, classification, and diagnosis of alarm floods.

Improvement for Implementation

For alarm management in practice, there always exist some limitations in resources, such as computational capacity, size of datasets, and incomplete historical data. To ensure the analysis of historical data is correct and the implementation of alarm improvement techniques gives satisfactory results, the following practical problems could be studied. To cope with large databases and online calculations, methods to improve computational efficiency are desired and deserve further research. Some potential directions include efficient computation algorithms and resources, such as parallel algorithms, cluster computing, and cloud computing. In many existing studies based on historical data, the procedures for data preprocessing are not well documented. However, as those methods heavily rely on the quality of historical data, it is critical to use reasonable data preprocessing methods. For instance, to reduce the data size for efficient storage, the historical process data could be compressed, leading to undesired distortion of data properties (e.g., data distribution). As a result, a data recovery strategy should be applied first before the process data is used for alarm design. The management of alarm systems is a sophisticated task with a long-lasting lifecycle. Therefore, it is desired that better interactive alarm management tools could be developed, such that alarm system configurations could be improved in a more user-friendly way and with increased efficiency.

Bibliography

[1] Naseeb Ahmed Adnan, Yue Cheng, Iman Izadi, and Tongwen Chen. Study of generalized delay-timers in alarm configuration. *Journal of Process Control*, 23(3):382–395, 2013.

[2] Naseeb Ahmed Adnan, Iman Izadi, and Tongwen Chen. On expected detection delays for alarm systems with deadbands and delay-timers. *Journal of Process Control*, 21(9):1318–1331, 2011.

[3] Muhammad Shahzad Afzal and Tongwen Chen. Analysis and design of multimode delay-timers. *Chemical Engineering Research and Design*, 120:179–193, 2017.

[4] Muhammad Shahzad Afzal, Tongwen Chen, Ali Bandehkhoda, and Iman Izadi. Analysis and design of time-deadbands for univariate alarm systems. *Control Engineering Practice*, 71:96–107, 2018.

[5] Muhammad Shahzad Afzal, Wen Tan, and Tongwen Chen. Process monitoring for multimodal processes with mode-reachability constraints. *IEEE Transactions on Industrial Electronics*, 64(5):4325–4335, 2017.

[6] Kabir Ahmed, Iman Izadi, Tongwen Chen, David Joe, and Tim Burton. Similarity analysis of industrial alarm flood data. *IEEE Transactions on Automation Science and Engineering*, 10(2):452–457, 2013.

[7] Ahmad W Al-Dabbagh, Wenkai Hu, Shiqi Lai, Tongwen Chen, and Sirish L Shah. Toward the advancement of decision support tools for industrial facilities: addressing operation metrics, visualization plots, and

alarm floods. *IEEE Transactions on Automation Science and Engineering*, 15(4):1883–1896, 2018.

[8] Margret Bauer, John W Cox, Michelle H Caveness, James J Downs, and Nina F Thornhill. Nearest neighbors methods for root cause analysis of plantwide disturbances. *Industrial & Engineering Chemistry Research*, 46(18):5977–5984, 2007.

[9] Margret Bauer and Nina F Thornhill. A practical method for identifying the propagation path of plant-wide disturbances. *Journal of Process Control*, 18(7-8):707–719, 2008.

[10] Suvomoy Bhaumik, John MacGowan, and Vimal Doraj. Mode based alarm solutions at Syncrude Canada. *IFAC PapersOnLine*, 48(8):653–656, 2015.

[11] Matthew Bransby and James Jenkinson. *The Management of Alarm Systems*. HSE Books Suffolk, UK, 1998.

[12] Neil Brown. Alarm management/ the EEMUA guidelines in practice. *Measurement and Control*, 36(4):114–119, 2003.

[13] Peter T Bullemer, Mischa Tolsma, Dal Vernon C Reising, and Jason C Laberge. Towards improving operator alarm flood responses: alternative alarm presentation techniques. *Abnormal Situation Management Consortium*, 2011.

[14] Shuang Cai, Ahmet Palazoglu, Laibin Zhang, and Jinqiu Hu. Process alarm prediction using deep learning and word embedding methods. *ISA Transactions*, 85:274–283, 2019.

[15] Shuang Cai, Laibin Zhang, Ahmet Palazoglu, and Jinqiu Hu. Clustering analysis of process alarms using word embedding. *Journal of Process Control*, 83:11–19, 2019.

[16] Zhansheng Cai, Laibin Zhang, Jinqiu Hu, Yan Yi, and Yu Wang. Comprehensive alarm information processing technology with application in petrochemical plant. *Journal of Loss Prevention in the Process Industries*, 38:101–113, 2015.

[17] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, San Francisco, CA USA, 2002.

[18] Sylvie Charbonnier, Nabil Bouchair, and Philippe Gayet. A weighted dissimilarity index to isolate faults during alarm floods. *Control Engineering Practice*, 45:110–122, 2015.

[19] Sylvie Charbonnier, Nabil Bouchair, and Philippe Gayet. Fault template extraction to assist operators during industrial alarm floods. *Engineering Applications of Artificial Intelligence*, 50:32–44, 2016.

[20] Yue Cheng, Iman Izadi, and Tongwen Chen. Optimal alarm signal processing: filter design and performance analysis. *IEEE Transactions on Automation Science and Engineering*, 10(2):446–451, 2013.

[21] Yue Cheng, Iman Izadi, and Tongwen Chen. Pattern matching of alarm flood sequences by a modified Smith-Waterman algorithm. *Chemical Engineering Research and Design*, 91(6):1085–1094, 2013.

[22] Seung Seok Choi, Sung Hyuk Cha, and Charles C Tappert. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, 8(1):43–48, 2010.

[23] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley, Boston, MA USA, 2010.

[24] Yiqian Cui, Junyou Shi, and Zili Wang. Multi-state adaptive BIT false alarm reduction under degradation process. *IEEE Transactions on Instrumentation and Measurement*, 64(3):671–682, 2015.

[25] Carl De Boor. *A Practical Guide to Splines*. Springer-Verlag New York, 1978.

[26] Gyula Dorgo and Janos Abonyi. Sequence mining based alarm suppression. *IEEE Access*, 6:15365–15379, 2018.

[27] Gyula Dorgo and Janos Abonyi. Learning and predicting operation strategies by sequence mining and deep learning. *Computers & Chemical Engineering*, 128:174–187, 2019.

[28] Gyula Dorgo, Kristof Varga, and Janos Abonyi. Hierarchical frequent sequence mining algorithm for the analysis of alarm cascades in chemical processes. *IEEE Access*, 6:50197–50216, 2018.

[29] Ping Duan, Tongwen Chen, Sirish L Shah, and Fan Yang. Methods for root cause diagnosis of plant-wide oscillations. *AIChE Journal*, 60(6):2019–2034, 2014.

[30] Ping Duan, Fan Yang, Tongwen Chen, and Sirish L Shah. Direct causality detection via the transfer entropy approach. *IEEE Transactions on Control Systems Technology*, 21(6):2052–2066, 2013.

[31] Ping Duan, Fan Yang, Sirish L Shah, and Tongwen Chen. Transfer zero-entropy and its application for capturing cause and effect relationship between variables. *IEEE Transactions on Control Systems Technology*, 23(3):855–867, 2015.

[32] EEMUA (Engineering Equipment and Materials Users' Association), London, UK. *Alarm Systems: A Guide to Design, Management and Procurement*, 2013.

[33] Fabio Fumarola, Pasqua Fabiana Lanotte, Michelangelo Ceci, and Donato Malerba. CloFAST: closed sequential pattern mining using sparse and vertical id-lists. *Knowledge and Information Systems*, 48(2):429–463, 2016.

[34] Pankaj Goel, Aniruddha Datta, and M Sam Mannan. Industrial alarm systems: challenges and opportunities. *Journal of Loss Prevention in the Process Industries*, 50:23–36, 2017.

[35] Anshu Gupta, Arun Giridhar, Venkat Venkatasubramanian, and Gintaras V Reklaitis. Intelligent alarm management applied to continuous pharmaceutical tablet manufacturing: an integrated approach. *Industrial & Engineering Chemistry Research*, 52(35):12357–12368, 2013.

[36] Liu Han, Huihui Gao, Yuan Xu, and Qunxiong Zhu. Combining FAP, MAP and correlation analysis for multivariate alarm thresholds optimization in industrial process. *Journal of Loss Prevention in the Process Industries*, 40:471–478, 2016.

[37] Health and Safety Executive. 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. 1997.

[38] Bill R Hollifield and Eddie Habibi. *Alarm Management: A Comprehensive Guide: Practical and Proven Methods to Optimize the Performance of Alarm Management Systems*. ISA (International Society of Automation), Durham, NC USA, 2011.

[39] Wenkai Hu, Ahmad W Al-Dabbagh, Tongwen Chen, and Sirish L Shah. Design of visualization plots of industrial alarm and event data for enhanced alarm management. *Control Engineering Practice*, 79:50–64, 2018.

[40] Wenkai Hu, Tongwen Chen, and Sirish L Shah. Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods. *IEEE Transactions on Industrial Electronics*, 65(9):7290–7300, 2018.

[41] Wenkai Hu, Tongwen Chen, and Sirish L Shah. Discovering association rules of mode-dependent alarms from alarm and event logs. *IEEE Transactions on Control Systems Technology*, 26(3):971–983, 2018.

[42] Wenkai Hu, Sirish L Shah, and Tongwen Chen. Framework for a smart data analytics platform towards process monitoring and alarm management. *Computers & Chemical Engineering*, 114:225–244, 2018.

[43] Wenkai Hu, Jiandong Wang, and Tongwen Chen. A local alignment approach to similarity analysis of industrial alarm flood sequences. *Control Engineering Practice*, 55:13–25, 2016.

[44] Wenkai Hu, Jiandong Wang, Tongwen Chen, and Sirish L Shah. Cause-effect analysis of industrial alarm variables using transfer entropies. *Control Engineering Practice*, 64:205–214, 2017.

[45] Ye Hu and Nael H El-Farra. Robust fault detection and monitoring of hybrid process systems with uncertain mode transitions. *AIChE Journal*, 57(10):2783–2794, 2011.

[46] Alan J Hugo. Estimation of alarm deadbands. *IFAC Proceedings Volumes*, 42:663–667, 2009.

[47] Sheue-Ling Hwang, Jhih-Tsong Lin, Guo-Feng Liang, Yi-Jan Yau, Tzu-Chung Yenn, and Chong-Cheng Hsu. Application control chart concepts of designing a pre-alarm system in the nuclear power plant control room. *Nuclear Engineering and Design*, 238(12):3522–3527, 2008.

[48] Nitin Indurkhya and Fred J Damerau. *Handbook of Natural Language Processing*. Chapman and Hall/CRC, Cambridge, UK, 2010.

[49] ISA (International Society of Automation), Durham, NC USA. *ANSI/ISA-18.2: Management of Alarm Systems for the Process Industries*, 2009.

[50] Eva Jerhotova, Marek Sikora, and Petr Stluka. Dynamic alarm management in next generation process control systems. In *IFIP International Conference on Advances in Production Management Systems*, pages 224–231. Springer, 2012.

[51] Hailei Jiang, Rohit Patwardhan, and Sirish L Shah. Root cause diagnosis of plant-wide oscillations using the concept of adjacency matrix. *Journal of Process Control*, 19(8):1347–1354, 2009.

[52] Ian Jolliffe. *Principal Component Analysis*. Springer, 2011.

[53] Radhia Kaced, Abdelmalek Kouadri, and Karim Baiche. Designing alarm system using modified generalized delay-timer. *Journal of Loss Prevention in the Process Industries*, 61:40–48, 2019.

[54] Jakob Kinghorst, Henry Bloch, Alexander Fay, and Birgit Vogel-Heuser. Integration of additional information sources for improved alarm flood detection. In *IEEE 21st International Conference on Intelligent Engineering Systems*, pages 19–26, 2017.

[55] Jakob Kinghorst, Mina Fahimi Pirehgalin, and Birgit Vogel-Heuser. Graph-based grouping of statistical dependent alarms in automated production systems. *IFAC PapersOnLine*, 51(24):395–400, 2018.

[56] Sandeep R Kondaveeti, Iman Izadi, Sirish L Shah, David S Shook, Ramesh Kadali, and Tongwen Chen. Quantification of alarm chatter based on run length distributions. *Chemical Engineering Research and Design*, 91(12):2550–2558, 2013.

[57] Hans Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Loop: local outlier probabilities. In *The 18th ACM conference on Information and knowledge management*, pages 1649–1652. ACM, 2009.

[58] Jason C Laberge, Peter Bullemer, Mischa Tolsma, and C Reising Dal Vernon. Addressing alarm flood situations in the process industries

through alarm summary display design and alarm response strategy. *International Journal of Industrial Ergonomics*, 44(3):395–406, 2014.

[59] Shiqi Lai and Tongwen Chen. A method for pattern mining in multiple alarm flood sequences. *Chemical Engineering Research and Design*, 117:831–839, 2017.

[60] Shiqi Lai, Fan Yang, and Tongwen Chen. Online pattern matching and prediction of incoming alarm floods. *Journal of Process Control*, 56:69–78, 2017.

[61] Shiqi Lai, Fan Yang, Tongwen Chen, and Liang Cao. Accelerated multiple alarm flood sequence alignment for abnormality pattern mining. *Journal of Process Control*, 82:44–57, 2019.

[62] Dewen Li, Jinghong Hu, Hao Wang, and Wenjun Huang. A distributed parallel alarm management strategy for alarm reduction in chemical plants. *Journal of Process Control*, 34:117–125, 2015.

[63] Yupeng Li, Weihua Cao, Wenkai Hu, and Min Wu. Abnormality detection for drilling processes based on Jensen-Shannon divergence and adaptive alarm limits. *IEEE Transactions on Industrial Informatics*, in press.

[64] Jun Liu, Khiang Wee Lim, Weng Khuen Ho, Kay Chen Tan, Rajagopalan Srinivasan, and Arthur Tay. The intelligent alarm management system. *IEEE Software*, 20(2):66–71, 2003.

[65] Matthieu Lucke, Moncef Chioua, Chriss Grimholt, Martin Hollender, and Nina F Thornhill. Online alarm flood classification using alarm coactivations. *IFAC PapersOnLine*, 51(18):345–350, 2018.

[66] Matthieu Lucke, Moncef Chioua, Chriss Grimholt, Martin Hollender, and Nina F Thornhill. Advances in alarm data analysis with a practi-

cal application to online alarm flood classification. *Journal of Process Control*, 79:56–71, 2019.

[67] Matthieu Lucke, Moncef Chioua, Chriss Grimholt, Martin Hollender, and Nina F Thornhill. Integration of alarm design in fault detection and diagnosis through alarm-range normalization. *Control Engineering Practice*, 98:104388, 2020.

[68] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.

[69] Qianqian Meng, Qunxiong Zhu, Huihui Gao, Yanlin He, and Yuan Xu. A novel scoring function based on family transfer entropy for Bayesian networks learning and its application to industrial alarm systems. *Journal of Process Control*, 76:122–132, 2019.

[70] Elham Naghoosi, Iman Izadi, and Tongwen Chen. Estimation of alarm chattering. *Journal of Process Control*, 21(9):1243–1249, 2011.

[71] Junya Nishiguchi and Tsutomu Takai. IPL2 and 3 performance improvement method for process safety using event correlation analysis. *Computers & Chemical Engineering*, 34(12):2007–2013, 2010.

[72] Tahereh Niyazmand and Iman Izadi. Pattern mining in alarm flood sequences using a modified prefixspan algorithm. *ISA Transactions*, 90:287–293, 2019.

[73] Masaru Noda, Fumitaka Higuchi, Tsutomu Takai, and Hirokazu Nishitani. Event correlation analysis for alarm system rationalization. *Asia-Pacific Journal of Chemical Engineering*, 6(3):497–502, 2011.

[74] Mark Rafferty, Xueqin Liu, David M Laverty, and Sean McLoone. Real-time multiple event detection and classification using moving window PCA. *IEEE Transactions on Smart Grid*, 7(5):2537–2548, 2016.

[75] Douglas H Rothenberg. *Alarm Management for Process Control: a Best-Practice Guide for Design, Implementation, and Use of Industrial Alarm Systems*. Momentum Press, NY, USA, 2009.

[76] Ellis Rubinstein and John F Mason. An analysis of Three Mile Island: the accident that shouldn't have happened. *IEEE Spectrum*, 16(11):33–42, 1979.

[77] Eduardo Navarra Satuf, Eugenius Kaszkurewicz, Roberto Schirru, and Mario Cesar Mello Massa de Campos. Situation awareness measurement of an ecological interface designed to operator support during alarm floods. *International Journal of Industrial Ergonomics*, 53:179–192, 2016.

[78] Markus Schleburg, Lars Christiansen, Nina F Thornhill, and Alexander Fay. A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system. *Journal of Process Control*, 23(6):839–851, 2013.

[79] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[80] Jun Shang and Tongwen Chen. Early classification of alarm floods via exponentially attenuated component analysis. *IEEE Transactions on Industrial Electronics*, 67(10):8702–8712, 2019.

[81] Bernard W Silverman. *Density Estimation for Statistics and Data Analysis*. CRC Press: Chapman & Hall, 1986.

[82] Harry Smith, Colin Howard, and Tony Foord. Alarms management priority, floods, tears or gain? introduction to the "problem". *Measurement and Control*, 36(4):109–113, 2003.

[83] Temple Ferris Smith and Michael Spencer Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[84] Jianjun Su, Cen Guo, Hao Zang, Fan Yang, Dexian Huang, Xiaoyong Gao, and Yan Zhao. A multi-setpoint delay-timer alarming strategy for industrial alarm monitoring. *Journal of Loss Prevention in the Process Industries*, 54:1–9, 2018.

[85] Yongkui Sun, Wen Tan, and Tongwen Chen. A method to remove chattering alarms using median filters. *ISA Transactions*, 2017.

[86] Wen Tan, Yongkui Sun, Ishtiza Ibne Azad, and Tongwen Chen. Design of univariate alarm systems via rank order filters. *Control Engineering Practice*, 59:55–63, 2017.

[87] Aditya Tulsyan, Feras Alrowaie, and Bhushan Gopaluni. Design and assessment of delay timer alarm systems for nonlinear chemical processes. *AIChE Journal*, 64(1):77–90, 2018.

[88] Aditya Tulsyan and R Bhushan Gopaluni. Univariate model-based deadband alarm design for nonlinear processes. *Industrial & Engineering Chemistry Research*, 58(26):11295–11302, 2019.

[89] Birgit Vogel-Heuser, Daniel Schütz, and Jens Folmer. Criteria-based alarm flood pattern recognition using historical data from automated production systems (aPS). *Mechatronics*, 31:89–100, 2015.

[90] Thumeera R Wanasinghe, Raymond G Gosine, Lesley Anne James, George KI Mann, Oscar de Silva, and Peter J Warrian. The internet of things in the oil and gas industry: a systematic review. *IEEE Internet of Things Journal*, 7(9):8654–8673, 2020.

[91] Jia Wang, Hongguang Li, Jingwen Huang, and Chong Su. Association rules mining based analysis of consequential alarm sequences in chemical

processes. *Journal of Loss Prevention in the Process Industries*, 41:178–185, 2016.

[92] Jia Wang, Hongguang Li, Jinwen Huang, and Chong Su. A data similarity based analysis to consequential alarms of industrial processes. *Journal of Loss Prevention in the Process Industries*, 35:29–34, 2015.

[93] Jiandong Wang and Tongwen Chen. An online method for detection and reduction of chattering alarms due to oscillation. *Computers & Chemical Engineering*, 54:140–150, 2013.

[94] Jiandong Wang and Tongwen Chen. An online method to remove chattering and repeating alarms based on alarm durations and intervals. *Computers & Chemical Engineering*, 67:43–52, 2014.

[95] Jiandong Wang and Tongwen Chen. Main causes of long-standing alarms and their removal by dynamic state-based alarm systems. *Journal of Loss Prevention in the Process Industries*, 43:106–119, 2016.

[96] Jiandong Wang, Fan Yang, Tongwen Chen, and Sirish L Shah. An overview of industrial alarm systems: main causes for alarm overloading, research status, and open problems. *IEEE Transactions on Automation Science and Engineering*, 13(2):1045–1061, 2016.

[97] Jiandong Wang, Zijiang Yang, Kuang Chen, and Donghua Zhou. Practices of detecting and removing nuisance alarms for alarm overloading in thermal power plants. *Control Engineering Practice*, 67:21–30, 2017.

[98] Jiandong Wang, Yan Zhao, and Zhenfu Bi. Criteria and algorithms for online and offline detections of industrial alarm floods. *IEEE Transactions on Control Systems Technology*, 26(5):1722–1731, 2018.

[99] Zhen Wang, Xingzhen Bai, Jiandong Wang, and Zijiang Yang. Indexing and designing deadbands for industrial alarm signals. *IEEE Transactions on Industrial Electronics*, 66(10):8093–8103, 2018.

[100] Iris Weiß, Jakob Kinghorst, Thomas Kröger, Mina Fahimi Pirehgalin, and Birgit Vogel-Heuser. Alarm flood analysis by hierarchical clustering of the probabilistic dependency between alarms. In *IEEE 16th International Conference on Industrial Informatics*, pages 227–232, 2018.

[101] Wanqi Xiong, Jiandong Wang, and Kuang Chen. Multivariate alarm systems for time-varying processes using Bayesian filters with applications to electrical pumps. *IEEE Transactions on Industrial Informatics*, 14(2):504–513, 2017.

[102] Wanqi Xiong, Jiandong Wang, and Kuang Chen. Multivariate alarm systems for time-varying processes using Bayesian filters with applications to electrical pumps. *IEEE Transactions on Industrial Informatics*, 14(2):504–513, 2018.

[103] Jianwei Xu, Jiandong Wang, Iman Izadi, and Tongwen Chen. Performance assessment and design for univariate alarm systems based on FAR, MAR, and AAD. *IEEE Transactions on Automation Science and Engineering*, 9(2):296–307, 2012.

[104] Xiaobin Xu, Shibao Li, Xiaojing Song, Chenglin Wen, and Dongling Xu. The optimal design of industrial alarm systems based on evidence theory. *Control Engineering Practice*, 46:142–156, 2016.

[105] Xiaobin Xu, Xu Weng, Dongling Xu, Haiyang Xu, Yanzhu Hu, and Jianning Li. Evidence updating with static and dynamical performance analyses for industrial alarm system design. *ISA Transactions*, 99:110–122, 2020.

[106] Xiaobin Xu, Haiyang Xu, Chenglin Wen, Jianning Li, Pingzhi Hou, and Jing Zhang. A belief rule-based evidence updating method for industrial alarm system design. *Control Engineering Practice*, 81:73–84, 2018.

[107] Yizhou Xu, Jiandong Wang, and Yan Yu. Alarm event prediction from historical alarm flood sequences based on Bayesian estimators. *IEEE*

Transactions on Automation Science and Engineering, 17(2):1070–1075, 2019.

[108] Fan Yang, Sirish Shah, and Deyun Xiao. Signed directed graph based modeling and its validation from process knowledge and process data. *International Journal of Applied Mathematics and Computer Science*, 22(1):41–53, 2012.

[109] Fan Yang, Sirish L Shah, Deyun Xiao, and Tongwen Chen. Improved correlation analysis and visualization of industrial alarm data. *ISA Transactions*, 51(4):499–506, 2012.

[110] Zijiang Yang, Jiandong Wang, and Tongwen Chen. Detection of correlated alarms based on similarity coefficients of binary data. *IEEE Transactions on Automation science and Engineering*, 10(4):1014–1025, 2013.

[111] Yan Yu, Jiandong Wang, and Zijian Ouyang. Designing dynamic alarm limits and adjusting manipulated variables for multivariate systems. *IEEE Transactions on Industrial Electronics*, 67(3):2314–2325, 2019.

[112] Yan Yu, Jiandong Wang, and Zijiang Yang. Design of alarm trippoints for univariate analog process variables based on alarm probability plots. *IEEE Transactions on Industrial Electronics*, 64(8):6496–6505, 2017.

[113] Yoshitaka Yuki. Alarm system optimization for increasing operations productivity. *ISA Transactions*, 41(3):383–387, 2002.

[114] Mohammed J Zaki and Ching Jui Hsiao. CHARM: an efficient algorithm for closed itemset mining. In *Proceedings of the 2002 SIAM International Conference on Data Mining*, pages 457–473, 2002.

[115] Hao Zang and Hongguang Li. Optimization of process alarm thresholds: a multidimensional kernel density estimation approach. *Process Safety Progress*, 33(3):292–298, 2014.

[116] Zhiyong Zeng, Wen Tan, and Rong Zhou. Performance assessment for generalized delay-timers in alarm configuration. *Journal of Process Control*, 57:80–101, 2017.

[117] Yan Zhang, Chi Yung Chung, Fushuan Wen, and Jinyuan Zhong. An analytic model for fault diagnosis in power systems utilizing redundancy and temporal information of alarm messages. *IEEE Transactions on Power Systems*, 31(6):4877–4886, 2016.

[118] Jianfeng Zhu, Yidan Shu, Jinsong Zhao, and Fan Yang. A dynamic alarm management strategy for chemical process transitions. *Journal of Loss Prevention in the Process Industries*, 30:207–218, 2014.