A Data-Driven Method to Discover Association Rules of Mode-Dependent Alarms

by

Kai Ru Wang

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

Master of Science

in

Control Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Alarm systems are considered to be essential components of industrial control systems, and are used to communicate the indication of alarm states to operators, ensuring the safe and efficient operation of modern industrial plants. Ideally, alarms should only appear when there are abnormal conditions that require operator responses, and the alarm rate should be kept to a reasonable number. But realistically, that's not always the case, and operators are often overloaded by nuisance alarms and alarm floods, distracting them from truly critical alarms, and potentially causing accidents to occur.

A variety of methods to manage alarm systems and improve their performance have been studied. Among them, state-based alarming is an advanced technique to reduce nuisance alarms and alarm floods by suppressing alarms associated with certain conditions or operating modes. To implement this technique, the key is to find the associations between operating modes and alarms. In practice, this is mainly done based on the experience of plant operators and expert knowledge of process engineers, and thus is very time consuming. Therefore, this thesis proposes a data driven method to discover the association rules of mode-dependent alarms for both single and multiple operating modes from the historical Alarm and Event (A&E) logs in an efficient way based on a data mining approach named FP-Growth (Frequent Pattern-Growth). The effectiveness of the proposed method is demonstrated by two industrial case studies.

Preface

This thesis is an original work by Kai Ru Wang. Parts of this thesis were presented and published as Wang, K., Hu, W., Chen, T., "An Efficient Method to Discover Association Rules of Mode-Dependent Alarms Based on the FP-Growth Algorithm," Proc. of the Annual IEEE Canada Power and Energy Conference, Edmonton, Canada, November 9-10, 2020.

To my parents Qiguo Wang and Shuangyu Ma, for all their love and support.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Tongwen Chen, for all his guidance and support throughout my graduate studies here at the University of Alberta. He helped me in finding the right direction for my research, and kept me focused on my goals through difficult times. This has truly been an experience of a lifetime, and one I will always treasure. Financial support from NSERC in the form of Research Assistantship, made available by Dr. Chen, is also gratefully acknowledged.

Further, I would also like to thank the current and former group members in Dr. Chen's group for their kindness and encouragement. Specifically, I would like to thank Dr. Wenkai Hu, for his contribution and assistance with my conference paper; Iman Amini who reviewed this thesis and provided helpful recommendations; Jing Zhou, Junyi Yang, Iman Amini, M. Rezwan Parvez, and Mani H. Dhullipalla who studied with me and helped me through the courses portion of the degree. I would also like to thank Ying Xiong, Boyuan Zhou, Jun Shang, Hao Yu, M. Hossein Roohi, Haniyeh S. Alinezhad, Li Deng, Shiming Wang and Harry Mohan Rao along the people mentioned above for the happy memories they brought to me.

Last but not least, I would like to thank my parents, Qiguo Wang and Shuangyu Ma for their love and support. My sincere gratitude towards my mother who specifically provided her guidance on the industrial aspects of this thesis with her process control knowledge and experience in oil and gas industry. My parents have always encouraged and supported me to be the best version of myself, and I cannot be more grateful for having them in my life.

Contents

List of Tables

List of Figures

List of Acronyms

A&E	Alarm & Event
DCS	Distributed Control System
EEMUA	Engineering Equipment and Materials Users' Association
HMI	Human-Machine Interface
ICS	Industrial Control Systems
ISA	International Society of Automation
OP	Output
P&ID	Pipeline and Instrumentation Diagram
PFD	Process Flow Diagram
PID	Proportional–Integral–Derivative
PLC	Programmable Logic Controller
PV	Process Variable
RTN	Returned to Normal
SCADA	Supervisory Control and Data Acquisition
SIS	Safety Instrumented System
SP	Setpoint

Chapter 1

Introduction

1.1 Background

According to the industrial standard from International Society of Automation (ISA) [1], "an alarm system is the collection of hardware and software that detects an alarm state, communicates the indication of that state to operators, and records changes in the alarm state." Alarm systems are integral in ensuring the safe and efficient operations of modern industrial facilities such as oil and gas plants, power and utility plants, pharmaceutical factories, and process and manufacturing plants, and are commonly deployed in such industries to monitor and inform operators of any abnormalities in operating conditions such as process measurements, operating equipment among other safety concerns.

In the past, alarm management was done on control panels, which are flat, and often vertical areas loaded with indicators and instruments that operators can use to monitor and control the plant. These control panels were wired to sensors in the process lines and equipment. To alert the operators of abnormal conditions, annunciator horns and lights of different colors (for example, green as good, yellow as warning, and red as bad) were used as indications of alarms in the system. Push buttons and/or switches were used to acknowledge and clear alarms. Due to the limitation of available board space on the control panel and cost of installing sensors and its physical wiring, horns, lights and switches, the amount of alarms and process indicators are often small, and any changes to the control panel were both costly and resource intensive.

Through the development of modern technology, control systems continue to advance and are eventually digitalized. In today's age, industrial control systems (ICS) like the distributed control systems (DCS) and supervisory control and data acquisition (SCADA) systems are used in nearly every industrial sector and critical infrastructure. Configuration and deployment of process indicators and alarms on such systems are very easy and relatively inexpensive. As such, alarms were configured to monitor every aspect of industrial processes. As an example, many systems configured alarm parameters whenever any process variable reaches beyond 20% and 80% of a normal operating range. This often results in a dramatic increase of daily alarm count at operator console station.

Recognizing this as a problem, many standards and guidelines including ANSI/ISA-18.2 [1], EEMUA-191 [20], and IEC-62682 [21] were developed to manage alarm systems across industries. According to such standards, the acceptable average and peak alarm rates are up to 1 and 10 alarms per 10 minutes per operator, respectively. However, the observed alarm rates in industries are much higher. In [20], it was discovered that the average and peak alarm rates were 6 and 220 alarms per 10 minutes in oil and gas industry, 9 and 180 alarms per 10 minutes in petrochemical industry, and 8 and 350 alarms per 10 minutes in power industry. Such high alarm rates can overwhelm operators, distracting them from truly critical alarms, and causing accidents [2][3][4].

According to ANSI/ISA-18.2 [1], an alarm is defined as "an audible and/or visible means of indicating to the operator an equipment malfunction, process deviation, or abnormal condition requiring a response." Thus an alarm that does not require any action by an operator becomes a nuisance alarm, and nuisance alarms are one of the major culprits of alarm overloading [22]. Alarms that transitions between alarm and normal states repeatedly in a short period of time are considered as chattering alarms [1]. They are the most encountered type of nuisance alarms, possibly accounting for 10-60% of alarm occurrences [12]. Other types of nuisance alarms include fleeting alarms, repeating alarms, redundant alarms and correlated alarms. Identification of these nuisance alarms can help process engineers to devise strategies to reduce them, therefore helping operators to focus on the truly important alarms.

Alarm floods are another major culprits of alarm overloading [22]. Alarm floods happen when the alarm rate is too high to be effectively managed by operators. This can be caused by many reasons including nuisance alarms and propagation of alarms from abnormal conditions in one process unit down to other units. An alarm flood begins when 10 or more alarms occur in 10 minutes, and ends when the alarm rate drops below 5 alarm in 10 minutes [1]. When an alarm flood occurs, operators are often overwhelmed, and may overlook critical alarms, which can lead to economic losses, environment impact and even loss of life, as seen in the Gulf of Mexico oil spill [23], Texas City refinery explosion [24] and the Texaco Refinery explosion [25].

Alarm and Event (A&E) logs are databases containing historical information about alarm messages and operator actions on the ICS. Table 1.1 shows an example of an A&E log. Typically, an alarm and event log contains information such as "Time" that the alarm or event occurred, "Tag Name" which is an unique identifier of equipment and process variables, "Event Type" such as operator action (mode) or alarm, "Condition" which shows the type of

Table 1.1: An example of an A&E log.

Time	Tag Name	Event Type	Condition	State
10:05:16	Valve01	Mode	N/A	Close
10:06:51	FI01	Alarm	PV_LOW	N/A
10:07:23	Valve01	Mode	N/A	Open
10:08:35	FI01	Alarm	RTN	N/A
10:09:05	TI01	Alarm	PV_HI	N/A
10:15:18	PI01	Alarm	PV_HI	N/A

alarms such as PV_LOW (Process Variable) or RTN (Returned to Normal), and "State" which indicates mode transition. Such information can be used for offline analysis and improve the efficiency of alarm systems.

1.2 Literature Survey

This section details literature survey on approaches in classification and reduction of nuisance alarms and alarm flooding, and state-based alarming which can be used to handle both nuisance alarms and alarm floods, and is the main technique used in this thesis.

1.2.1 Nuisance Alarms

Nuisance alarms are alarms that does not require operator action and are therefore distracting operators from truly critical alarms, and are one of the major culprits of alarm overloading. Many papers studied nuisance alarms and proposed approaches to reduce them. Wang et al. [29] used deadbands to reduce nuisance alarms. The paper used normalized alarm duration and normalized alarm deviations as two metrics to determine if deadbands should be used to remove nuisance alarms of analog signals. Then, it proposed a method to find the optimal deadband width, balancing between a nuisance-alarm duration ratio (NADR) and a peudo-detection delay. Granger causality (GC) analysis is another technique to reduce nuisance alarms by alarm propagation pathway investigation and alarm source identification. He et al. [30] proposed an improved design of GC models by using an attention-based long short-term memory (ALSTM) method. Cai et al. [7] utilized an alarm clustering method to provide insights towards removal of redundant alarms. Correlated alarms can be very similar to each other and indicate the same abnormality in industrial processes. Wang et al. [6] exploited a weighted fuzzy association rule mining approach to discover correlated alarm sequences. Yang et al. [8] developed a block matching similarity analysis method to detect alarm correlations.

As discussed previously, chattering alarms are the most encountered type of nuisance alarms; thus many studies are devoted to the prediction and removal of chattering alarms. Studies have used machine learning to predict chattering alarms [26, 27]. Using a modified approach based on run lengths distribution, [26] divided historical alarm events into 2 categories according to the likelihood of future alarm chattering. Then, they used the categorized alarms to train a deep neural network, and evaluated the performance of the algorithm for the ability to predict chattering alarms. Tamascelli et al. [27] developed a method for dynamic chattering quantification, and used the results to train three machine learning models: a linear model, a deep neural network model, and a hybrid model of the previous two. The three models were then evaluated and compared for their performance. Other studies have developed techniques to remove chattering alarms [11, 28]. Sun et al. [28] proposed two rules in designing window size of median filters in order to reduce chattering alarms. If alarm probability estimation is available, rule no. 1 is used to choose a window size in order to satisfy requirements on the false alarm rate (FAR) and miss alarm rate (MAR). Rule no. 2 is used if only historical alarm data is available. It uses duration distribution to satisfy the reduction percentage of chattering alarms. Mannani et al. [11] proposed an alarm data preprocessing technique to remove chattering alarms and reconstruct missing alarms. The paper discussed techniques to remove chattering alarms using point-based data and interval-based data, and the selection of removal time frame using a global time frame and a variable-dependent time frame.

1.2.2 Alarm Floods

Alarm floods happen when the alarm rate is so high that operators can no longer handle them effectively, distracting them from important alarms and likely leading to incidents. Many papers studied alarm floods and proposed techniques for reducing alarm floods and its severe effects on alarm systems.

Several studies proposed methods for alarm flood sequencing and root cause analysis. Unlike other traditional statistical or model-based methods, alarm flood sequence alignment (ASFA) provides fault inference from the perspective of alarm sequence similarity assessment. Guo et al. [31] proposed a new ASFA method: the match-based accelerated alignment (MAA) to detect perceptive alarm sequence alignments. To find the root cause of an alarm flood, one approach is to find similar patterns in alarms triggered in different alarm floods. Niyazmand et al. [32] used a modified PrefixSpan sequential pattern recognition algorithm to find these alarm patterns in different alarm floods. Luo et al. [5] applied the Bayesian networks to trace root causes of alarms.

Alarm floods prediction is another area of study in alarm floods. By examining historical alarm flood sequences, the study in [33] made predictions of upcoming alarm events for a current alarm sequence. To calculate the posterior probabilities for all candidates of predicted alarm events and their confidence intervals, Bayesian estimators were used. Lai et al. [34] proposed an online algorithm that could calculate similarity between an online alarm sequence and a pattern database, therefore making early prediction of an incoming alarm flood. By using the proposed algorithm, state identification can potentially be used in the applications of online alarm suppression and predictive alarming.

Studies [35], [36], [10] and [9] proposed methods for classification and detection of alarm floods. Lucke et al. [35] reviewed a number of alarm data analysis methods to reduce the impact of alarm floods, and noticed that online applications were developed in attempting to help operators during an alarm flood, but these applications only considered the problem as sequence mining. To classify ongoing alarm floods, the paper proposed a binary series approach using historical data on alarm floods. Wang et al. [36] reviewed basic criteria for alarm floods detection, and proposed a new criterion based on the number of alarm variables newly appeared in an alarm state. Using the new criterion, the paper then proposed an algorithm to detect both online occurrence of alarm floods and offline detection of the presence of alarm floods in historical data. Shang et al. [10] proposed an exponentially attenuated component analysis method for early classification of alarm floods. Hu et al. [9] proposed an itemset mining method to detect frequent patterns in alarm floods after identifying and extracting alarm floods from historical data based on alarm rates.

1.2.3 State-Based Alarming

The ANSI/ISA-18.2 standard [1] and the EEMUA-191 guideline [20] also recommend state-based alarming as an enhanced and advanced alarm management technique to handle nuisance alarms or alarm floods associated with certain operating states for equipment or processes. The idea is that some alarms are only useful when operating equipment or processes are in a certain state [13]. When the state is changed, these alarms are no longer indicating the true abnormalities and are just reflections of the state change. Thus, by configuring state-based alarming, such nuisance alarms are suppressed and will not be presented to plant operators [39–41, 46].

State-based alarming is also known as dynamic alarming, logic-based alarming or condition-based alarming in the literature and in industry [40]. Using a Bayesian estimation based dynamic alarm management (BEDAM) method, dynamic alarm limits were obtained and used to control alarm floods during chemical process transitions [39]. Advanced logic-based alarming strategies were used to detect faults and monitor hybrid process systems [42]. A condition-based multistage adaptative method was used to detect and reduce false alarms in mechanical systems [43].

As state-based alarming is highly effective in reducing nuisance alarms and alarm floods, it is being applied progressively more in practice [12, 13, 40], and considered to be an important technique of the next generation industrial process control systems [40]. Many industrial facilities have successfully used state-based alarming techniques to improve their alarm systems, including chemical plants [13], an oil sand extraction plant [38], a nuclear power plant simulator [44], and an ethylene plant [45]. A dynamic alarm management technique was used in [45] to reduce alarms during a 7-hours alarm flood from 1450 alarms to a manageable number for operators. Syncrude developed mode-based solutions (MBS) and deployed it on the Honeywell DCS for their oil sand extraction plant to effectively manage alarms. MBS kept unnecessary alarms out of alarm summary page of the HMI based on certain operating modes of equipment or processes, reducing stress on operators [38].

To implement the state-based alarming technique, it requires to find out the associations between the operating modes and alarms. In practice, this is usually done based on the experience of plant operators and expert knowledge of process engineers, and thus is very time consuming [13, 37, 39]. In [15], a completely automated data-driven method based on the Apriori algorithm was proposed to detect the association rules of mode-dependent alarms from Alarm and Event (A&E) logs that record all historical events of operating modes and alarms. This method involves two cases, including the detection of mode-dependent alarms for both single and multiple modes. Additionally, a separate step was used to detect frequent patterns of multiple operating modes first. Thus, the computational efficiency is much compromised due to the separated steps. Additionally, the Apriori algorithm is also computationally inefficient especially for large datasets or long sequences.

1.3 Thesis Contributions

As discussed above, the computational efficiency of the method in [15] is comprised due to the separated steps and the inefficiency of Apriori algorithm for large datasets. Therefore, the question is how to improve the efficiency for the detection of mode-dependent alarms from the historical A&E logs. Motivated by this problem, this thesis proposes a new detection method based on a data mining approach named FP-Growth (Frequent Pattern-Growth). The major contributions of this thesis are summarized below:

1. Proposed an efficient data-driven method to discover the association rules of mode-dependent alarms using historical A&E logs.

2. The proposed method is capable of detecting associations rules of mode-dependent alarms for both single and multiple modes together in one algorithm.

3. Demonstrated the superior performance of the proposed method with two industrial case studies in comparison with a previous study on mode-dependent alarms [15].

1.4 Thesis Organization

The remaining of this thesis is organized as follows.

Chapter 2 provides an introduction on state-based alarming and association rule mining. Apriori, FP-Growth, and other popular algorithms in association rule mining will be discussed and compared.

Chapter 3 introduces the mathematical definitions and formulates the problem of mode dependent alarms detection.

Chapter 4 presents the systematic method for the detection of mode-dependent alarms. There are three major steps:

1. The original A&E data is reorganized in a transactional format through data preprocessing and segmentation.

2. The association rules are obtained using the FP-Growth algorithm and the likelihood ratio.

3. Spurious and redundant rules are determined and excluded through post-filtering.

Chapter 5 demonstrates the effectiveness and practicability of the proposed method using A&E data from two large scale process facilities.

Chapter 6 presents the concluding remarks and potential direction of future work.

Chapter 2

Preliminaries

This chapter introduces the concept of mode dependent alarms, and association rule mining and the different algorithms in this branch of data mining.

2.1 Mode Dependent Alarms

This section introduces the concept of state-based alarming and its usage in practice, and the two parts of mode dependent alarms, namely, operating modes and its consequential alarms.

2.1.1 State-Based Alarming in Practice

As discussed in Chapter 1, state-based alarming is an advanced alarm management technique recommended by the ANSI/ISA-18.2 standard [1] to reduce nuisance alarms and alarm floods associated with certain operating states for equipment or processes. In industrial process plants, these equipment and processes have multiple operating states or operating modes, but most alarms are only useful when they are in a certain state. Alarms that occur when the state is changed are just reflections of the state change, they do not indicate true abnormalities, thus making them nuisance alarms. When too many of these nuisance alarms are presented at the same time, alarm floods can also occur, both of which will distract the operators from the true state of the

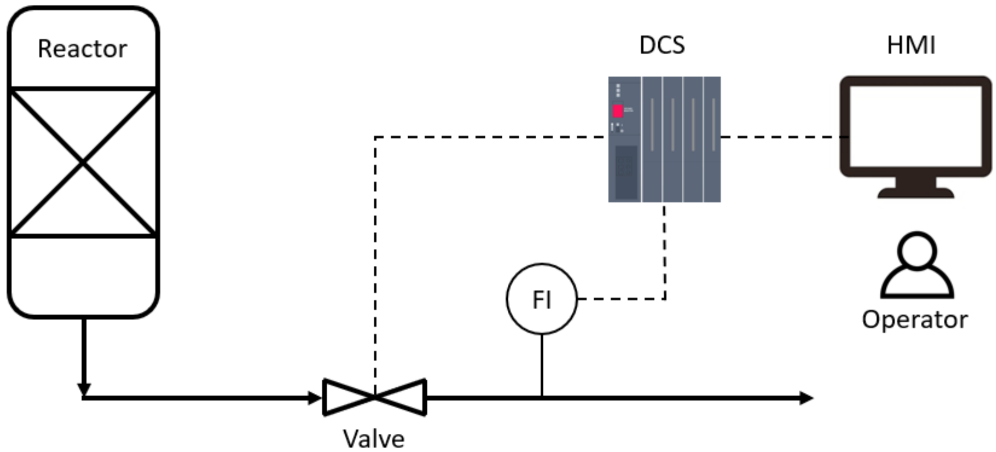

Figure 2.1: An example of state-based alarming: association between a valve and a flow indicator.

plant, and potentially cause huge incidents. By using state-based alarming to find these potential nuisance alarms, and suppress them based on certain operating states, such disasters can be avoided.

A potential industrial example of where the state-based alarm can be used is presented in Figure 2.1. In this example, an operator closed a valve from the HMI screen in the control room, which fully shut off the physical valve in the field through DCS. A flow indicator directly downstream of the valve was configured to monitor the flow rate of the line. In order to warn operators of its abnormal condition, several alarm parameters were configured on the flow indicator, such as PV (process variable) high or PV low, which will alarm whenever the flow rate either raises above a pre-defined high alarm limit or drops below a pre-defined low alarm limit. In this case, after the operator closed the valve, a flow PV low alarm was triggered and annunciated on the HMI. However the PV low alarm was the direct consequence of closing the valve upstream. It is expected, therefore did not indicate a true abnormality, just a reflection of the state change. The alarm is useless as no action was

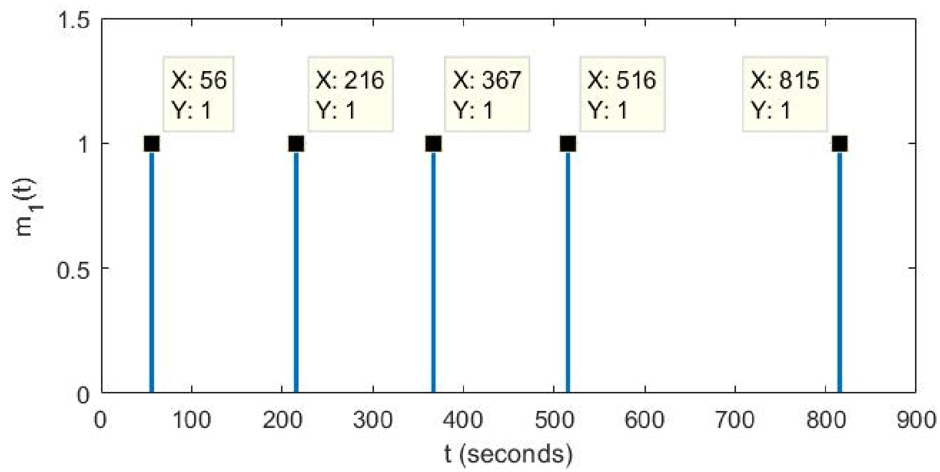

Figure 2.2: An example of multiple occurrences of an operating mode.

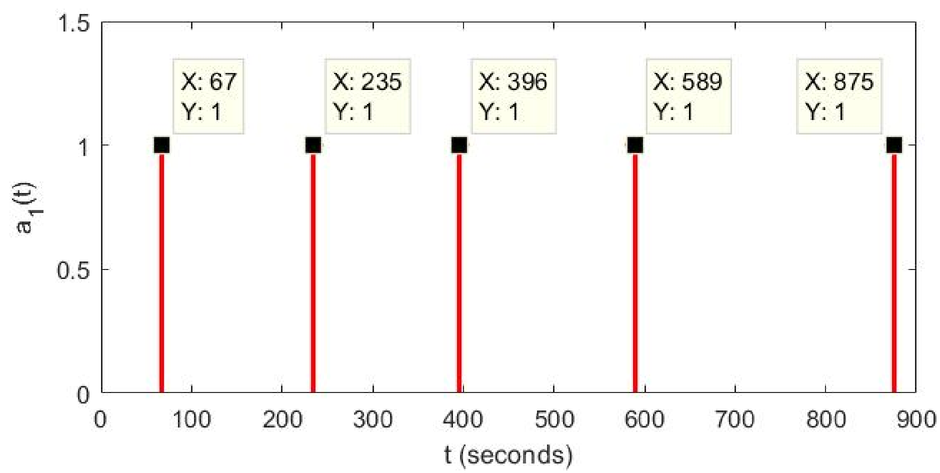

Figure 2.3: An example of multiple occurrences of a consequential alarm.

required and is considered a nuisance alarm. Such alarms can be configured by process control engineers to be suppressed when certain states or conditions are met.

Figure 2.2 shows graph of multiple occurrences of the valve being closed by operators $t^{(e)}_{m_1}$ and Figure 2.3 shows graph of multiple occurrences of the flow PV low alarm $t^{(e)}_{a_1}$. It can be seen that each occurrence of the flow PV low alarm $t^{(e)}_{a_1}$ always follows the valve being closed by operators $t^{(e)}_{m_1}$ very closely. Hence, based on the concept of state-based alarming, $t^{(e)}_{a_1}$ is an consequential alarm of the state change $t_{m_1}^{(e)}$, making it a nuisance alarms. Such alarms should be configured to be suppressed whenever operators close the valve, and re-enabled when the valve is opened again.

2.1.2 Operating Modes and Consequential Alarms

State-based alarming reduces nuisance alarms and alarm floods by suppressing consequential alarms when processes and operating equipment are in a certain state. Processes such as different chemicals being fed into the plant, startup and shutdown of an unit or an entire plant, and change of process to produce different products can be very difficult to determine, and require a combination of process data and expertise process knowledge from plant personnel such as seasoned operators, process engineers and process control engineers. Instead of looking at these complicated processes, the focus of this thesis is on the states of operating equipment, also called operating modes of equipment. These state changes are often done by operators, and as such, are recorded as Alarm and Event (A&E) logs in real time in control systems like DCS or SCADA. The recorded data can easily be exported into Excel tables out of A&E logs. Hence, it can be used in data-driven methods to determine mode dependent alarms without any additional work by plant personnel or plant knowledge.

Table 2.1 shows the most commonly and typically used operating equipment in industrial plants and the various possible operating modes they can be in. Some equipment only has two possible states, like open and close for valves, but others can have multiple states, such as auto, manual, cascade, SP (Setpoint) change and OP (Output) change for controllers. For controllers, there can be digital state changes like auto/manual/cascade and analog state changes like SP and OP changes. Typically for digital changes, under one

Table 2.1: Commonly used equipment and their operating modes.

Equipment	Operating modes
Blower	Start / Stop / Standby
Controller	Auto / Manual / Cascade / SP change / OP change
Fan	Start / Stop / Standby
Motor	Start / Stop / Standby / Forward / Reverse
Pump	Start / Stop / Standby
Valve	Open / Close

state or operating mode of an equipment, everything works normally, and any alarm that may annunciate during that time indicates a true abnormality and requires attention from operators. While under other states, an alarm can become a consequential alarm of the state change, and is therefore a nuisance alarm. According to Syncrude's definitions, operating states of equipment can also be determined by "a defined process variable that reaches a specific limit" [38]. Such state changes are analog changes. Similarly, when analog changes are within limits, any alarms annunciated indicate true abnormalities, but when these changes reach and surpass a specific limit, alarms annunciated become consequential alarms of the analog change.

Determination of these consequential alarms of state changes can then be used to configure them in such a way to suppress these nuisance alarms whenever the state changes, therefore reducing potential alarm floods and disasters. In industries, the determination of mode dependent alarms are typically done using process knowledge of plant personnel like operators and engineers. However, this detection method has proven to take a lot of time and resources, and sometimes even then, potential mode dependent alarms are missed. This thesis follows the study in [15], and continue to use data driven methods to automate the detection process. The results can then be used by plant process control engineers to configure and suppress these consequential alarms associated with changes of certain operating modes.

2.2 Association Rule Mining

Association rule mining is one of the major branches of data mining. It is a rule-based machine learning method used to mine interesting connections, co-occurrences, or associations between items in a large database. Discovered associations can be used by organizations to make important decisions and increase overall profit. When association rule mining was first introduced in [47], it was originally used in market basket analysis. “Beer and diaper” is a famous tale often told to illustrate data mining concepts. The idea is that supermarkets discovered through data that customers who buy diapers also buy beer at the same time, and it was presumed that husbands sent out to buy diapers by their wives also purchase beer as they no longer get as much time drinking at pubs. It is dubious how much of the story is true, however it did become a popular example of how through association rule mining, everyday data can be used to find unexpected associations. Association rule mining has since expanded its use to many application areas including bioinformatics, telecommunication networks, website navigation examination, market and risk management and intrusion detection [48] [49].

An example of its original use in market basket analysis is shown in Table 2.2, and used to introduce some definitions and concepts of association rule mining as defined in [47]. Some definitions are presented first:

1. Let $I = \{i_1, i_2, \cdots, i_n\}$ be a set of n binary attributes called items.

2. Let $D = \{t_1, t_2, \cdots, t_m\}$ be a set of transactions called the database.

Table 2.2: An example database of market-basket transactions.

Transaction ID	Items
1	Bread, Beer, Diaper, Egg
2	Diaper, Beer, Milk
3	Beer, Milk, Diaper, Bread
4	Bread, Milk
5	Bread, Milk, Diaper

Table 2.3: An example database of market-basket transactions in binary vector form.

Transaction ID	Beer	Bread	Diaper	Egg	Milk
1	1	1	1	1	0
2	1	0	1	0	1
3	1	1	1	0	1
4	0	1	0	0	1
5	0	1	1	0	1

3. Each transaction t in database D has an unique transaction ID and contains a subsets of the items i in I.

4. Each t is represented as a binary vector, where $t[k] = 1$ if t includes the item i_k, and $t[k] = 0$ if t doesn't.

5. An association rule is defined as an implication of the form $X \Rightarrow Y$, where $X \subseteq I$ is a set of some items in I, and $Y \subseteq I$ is a set of some items in I, also known as itemsets.

6. An itemset that contains k items is called k-itemset.

From the example, the set of items is $I = \{beer, bread, diaper, egg, milk\}$. Table 2.3 shows the database in its binary vector form, where, the value 1 means the appearance of the item i in the corresponding transaction t, and value 0 means the absence of i in t.

Next, some important concepts of association rule mining are discussed. The support of an itemset is the frequency of appearance of the itemset in the database D. The support of itemset X in database D is defined as the proportion of transaction t in D which contains X, and is calculated as:

$$s(X) = \frac{|t \in T; X \subseteq t|}{|T|} \tag{2.1}$$

$|\cdot|$ indicates the size of a set or vector.

The itemset $X = \{beer, diaper\}$ has a support of $s(\{beer, diaper\}) = 3/5 = 0.6$, which means that X occurred in 60% of all transactions. An itemset whose support is larger than or equal to a minimum support threshold S_{min} is a frequent itemset. The support of the association rule $X \Rightarrow Y$ where $X = \{bread, diaper\}$ and $Y = \{beer\} = 2/5 = 0.4$. Confidence is a measure of how often items in Y appear in transactions that contain X. The confidence of an association rule $X \Rightarrow Y$, with respect to a set of transactions in database D is calculated as:

$$c(X \Rightarrow Y) = \frac{s(X \cup Y)}{s(X)} \tag{2.2}$$

The confidence of the association rule $X \Rightarrow Y$ where $X = \{bread, diaper\}$ and $Y = \{beer\}$ is:

$$c(X \Rightarrow Y) = \frac{s(\{bread, diaper, beer\}}{s(\{beer, diaper\})} = \frac{0.4}{0.6} = \frac{2}{3} = 0.67 \tag{2.3}$$

Both the support and confidence are important rule evaluation metrics of association rule mining. The support shows the coverage of the rule and the confidence shows the accuracy of the rule.

2.2.1 Apriori Algorithm

The Apriori algorithm is one of the most well-known algorithms as well as the most original association rule mining algorithm proposed in [47]. This algorithm was also used in previous study [15] to detect mode dependent alarms. The Apriori algorithm uses a two-step approach: candidate itemset generation and test for frequency. First, it generates candidate itemsets of length k from candidate itemsets of length $k-1$. Then, it tests these candidate itemsets against the entire database to find their supports. If they are bigger than the minimum support S_{min} or frequency threshold F_{th} (as discussed later in Chapter 3), they are moved onto the next level, where another item is added to the candidate itemset. The concept is if an itemset is frequent, subsets of this itemset must also be frequent. If an itemset is infrequent, supersets of this itemset must also be infrequent. For example, assume there are 3 items in a transaction, $D_l = (e_{l,1}, e_{l,2}, e_{l,3})$, if candidate itemset $C_n = (e_{l,1})$ is not frequent, then $C_{n+1} = (e_{l,1}, e_{l,2})$ cannot be frequent, and subsequently $C_{n+2} = (e_{l,1}, e_{l,2}, e_{l,3})$ cannot be frequent; thus this branch of items is pruned from the database. If $C_{n+3} = (e_{l,2})$ is frequent, then $C_{n+4} = (e_{l,2}, e_{l,3})$ is the next level of candidate itemset and is tested for frequency and so on. After all potential association rules are found, they are tested for confidence values, and only rules with confidence values higher than the minimum confidence C_{min} are considered as final association rules.

However when dealing with larger dataset, this method is very time-costly to generate candidate itemsets. It also repeatedly scans the entire database to count support for every candidate itemset, adding to the time and memory burden.

Table 2.4: An example of a transactional database.

Transactions	Events
D_1	e_{k1}, e_{k2}, e_{k3}
D_2	e_{k2}, e_{k3}, e_{k4}
D_3	e_{k1}, e_{k3}
D_4	e_{k1}, e_{k2}
D_5	e_{k2}, e_{k3}

2.2.2 FP-Growth Algorithm

In this thesis, instead of the Apriori algorithm, the FP-Growth (Frequent Pattern Growth) algorithm was used to detect mode dependent alarms. The FP-Growth algorithm is a newer, more efficient method of mining patterns. It was proposed by Han in his paper [16], where it was compared with Apriori [14] and TreeProjection [17] (a new proposed method at the time), and proved to have a better performance. In later works, it was also proved that FP-Growth outperforms Apriori and ECLAT (another popular method) [18][19]. In one of the comparison papers [19], it was found that FP-Growth and ECLAT are the fastest algorithms in frequent itemset mining (also called association rule mining) at the time of the publication of the paper. Finding better ways to filter both frequent itemsets and association rules or to produce less in the first place is considered an ongoing challenge [19]. Filtering can effectively reduce data set and make the algorithm run faster and use less memory. Many filters are set within the new method for this purpose, and will be discussed later in Chapter 4.

In comparison to Apriori, FP-Growth discovers association rules without the expensive candidate itemset generation step or repeatedly scanning the entire database. Instead, it uses a two-step approach (Algorithms 1 and 2).

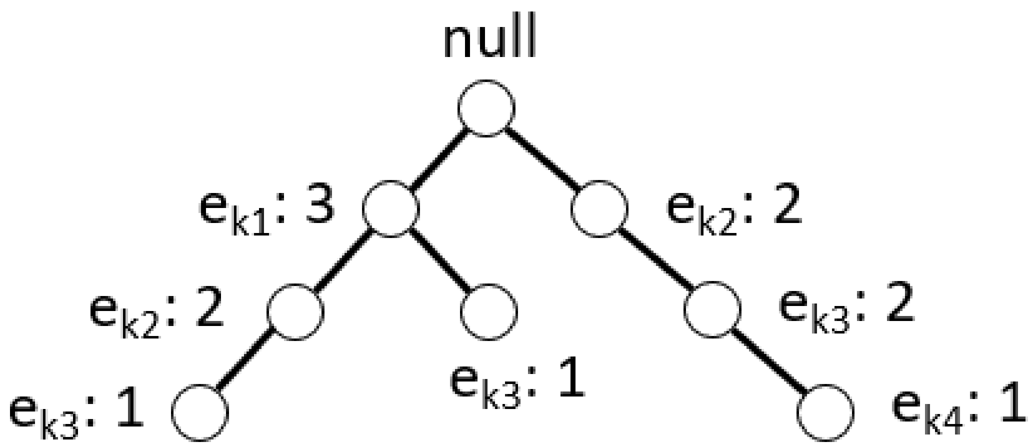

Figure 2.4: An example of a frequent pattern tree.

Algorithm 1 is first used to generate a compact data structure called FP-tree (frequent pattern tree) using A&E log $\mathbb{E}$ organized in a transactional database format $\tilde{\mathbb{E}} =< D_1, D_2, \cdots, D_L >$ and minimum threshold F_{th}. Then, association rules $\phi : M \Rightarrow A$ are mined directly from the more compact FP-tree. The FP-tree is an extended prefix-tree structure that stores quantitative, key information of association rules in a database. Additionally, this step only uses 2 scans of the database, as opposed to the many scans Apriori uses. Algorithm 2 is then applied to mine frequent patterns directly from the more compact FP-tree. A divide and conquer search technique is used instead of the bottom-up generation technique of the Apriori method. The algorithm first divides the FP-tree into smaller trees called conditional FP-trees, each with a set of frequent items, then looks for short patterns recursively and concatenating them in long frequent patterns. As a simple example, using a set of transactions as shown in Table 2.4 and a $F_{th} = 2$, the FP-tree (as shown in Figure 2.4) is constructed, with each node containing an event e_k and its frequency ρ. From the FP-tree, association rules $\phi_1 = (e_{k1}, e_{k2})$ and $\phi_2 = (e_{k2}, e_{k3})$ are obtained.

To better understand FP-Growth, algorithmic outlines of Algorithms 1 and 2 are summarized and provided in Section 4.2, more details regarding the algorithms can be found in [16].

Chapter 3

Problem Description

This chapter introduces the mathematical definitions used in this thesis and formulates the problem of mode dependent alarms detection.

3.1 Mathematical Definitions

The objective of this thesis is to discover the association rules of mode-dependent alarms for both single and multiple operating modes based on the historical A&E log that contains events for both alarms and operating modes. To better present the method, a series of basic notations are given as follows:

1. An unique operating mode is denoted as m_i

2. The set of all unique modes in an alarm system is then denoted as $\mathcal{M} = \{m_i : i = 1, 2, \cdots, |\mathcal{M}|\}$, where $|\cdot|$ indicates the size of a set or vector.

3. An unique alarm is denoted as a_j

4. The set of all unique alarms in an alarm system is then denoted as $\mathcal{A} = \{a_j : j = 1, 2, \cdots, |\mathcal{A}|\}$.

5. The kth timed event of a mode or alarm $e_k \in \mathcal{M} \cup \mathcal{A}$ at the time instant t_k is denoted as (e_k, t_k)

6. An A&E log consists of timed events over a certain time period T and is represented as $\mathbb{E} = \{(e_k, t_k) : e_k \in \mathcal{M} \cup \mathcal{A}, t_k \in T, k = 1, 2, \cdots, N\}$, where N is the number of events in $\mathbb{E}$

7. The sub-databases of all mode events are denoted as $\mathbb{E}(\mathcal{M}) = \{(m_i, t_k) : m_i \in \mathcal{M}, t_k \in T, k = 1, 2, \cdots, N\}$

8. The sub-databases of all alarm events are denoted as $\mathbb{E}(\mathcal{A}) = \{(a_j, t_k) : a_j \in \mathcal{A}, t_k \in T, k = 1, 2, \cdots, N\}$

9. $\mathbb{E} = \mathbb{E}(\mathcal{M}) \cup \mathbb{E}(\mathcal{A})$

10. Given all events for an unique mode $e = m_i$ or alarm $e = a_j$, a corresponding time vector is obtained as $\mathbf{t}^{(e)} = [t_1^{(e)}, t_2^{(e)}, \cdots, t_{|\mathbf{t}^{(e)}|}^{(e)}]^T$, where $t_{l_1}^{(e)} < t_{l_2}^{(e)}$ for $l_1 < l_2$. The time vectors for $m_i \in \mathcal{M}$ and $a_j \in \mathcal{A}$ are $\mathbf{t}^{(m_i)}$ and $\mathbf{t}^{(a_j)}$, respectively.

11. An association rule of mode-dependent alarms is denoted as $\phi : M \Rightarrow A$, where $M \subseteq \mathcal{M}$ represents a set of operating modes and $A \subseteq \mathcal{A}$ denotes a set of alarms. The symbol $\Rightarrow$ indicates that the alarms in A are consequential alarms of operating modes in M. It should be noted that either M or A should contain at least one element, i.e., $M \neq \emptyset$ and $A \neq \emptyset$.

3.2 Problem Formulation

The problem is formulated as: given an A&E log $\mathbb{E}$ on $\mathcal{M}$ and $\mathcal{A}$, the objective is to detect all possible association rules $\phi : M \Rightarrow A$, such that

1. the alarms in $\mathcal{A}$ must happen after the occurrence of the last operating mode in $\mathcal{M}$, i.e., $t_{|t^{(m_i)}|}^{(m_i)} < t_{l_2}^{(a_j)}$ for $\forall a_j \in A$, and

2. the time difference between the first operating mode in $\mathcal{M}$ and the last alarm in $\mathcal{A}$ must be within a certain time window W, i.e., $\left(t^{(a_j)}_{|t^{(a_j)}|} - t^{(m_i)}_1\right) \in (0, W]$, and

3. the association rule is frequent, i.e., the frequency of the rule $\sigma(M \Rightarrow A)$ is no less than the required minimum threshold F_{th}.

The first and second conditions guarantees that the alarms in an association rule always happen after the operating modes and all events $e = m_i \cup a_j$ happen within a pre-determined time window W. Otherwise, if an alarm happens before a mode or the delay is too large, it is very likely not a consequential alarm of this mode. The third condition calculates how many times a set of modes $\mathcal{M}$ is followed by a set of alarms $\mathcal{A}$ and makes sure that the rule is frequent in the historical database. If a rule is rarely seen, it cannot be an interesting association rule and thus should be excluded.

Chapter 4

Methodology

This chapter presents a systematic method for the detection of association rules of mode-dependent alarms. The main steps include preprocessing and segmentation of A&E data, detection of association rules, as well as determination and removal of spurious and redundant rules.

4.1 Preprocessing and Segmentation of A&E Data

The raw A&E data $\mathbb{E}$ is essentially a long sequence of timed events. In this thesis, the FP-Growth algorithm [16] is applied to discover the association rules for both single mode and multiple modes of mode-dependent alarms from the A&E data. The algorithm requires the data organized in a transactional database format, i.e., the data consists of a number of transactions and each transaction is comprised by a series of items. Thus, the A&E data needs to be broken into transactions before inputting it into the FP-Growth algorithm for the association rule mining.

According to the objective described in Chapter 3, all alarms A of a detected association rule must follow the last operating mode in M, and the time difference between the first operating mode in $t_1^{(m_i)}$ and the last alarm

$t^{(a_j)}_{|t^{(a_j)}|}$ must be within a certain time window W. Otherwise, the occurrences of these alarms A are regarded to be irrelevant with the switching of the set of operating modes M if their time difference is beyond $(0, W]$, and are therefore not considered as consequential alarms associated with this set of operating modes M.

Here, the A&E log $\mathbb{E} = \{(e_k, t_k) : e_k \in \mathcal{M} \cup \mathcal{A}, t_k \in T, k = 1, 2, \cdots, N\}$ is assumed to be chronologically ordered, i.e., $t_{k_1} \leq t_{k_2}$ for $1 \leq k_1 \leq k_2 \leq N$. If the time gap of two adjacent events e_k and e_{k+1} is more than a pre-defined time window W, i.e., $t_{k+1} - t_k > W$, the relation between e_k and e_{k+1} is not of interest, and thus the A&E log $\mathbb{E}$ can be divided at the position between t_k and t_{k+1} such that e_k and e_{k+1} are included in two separated transactions. Then, the A&E log $\mathbb{E}$ is reorganized in a transactional database format as $\tilde{\mathbb{E}} =< D_1, D_2, \cdots, D_L >$, where L is the number of transactions; each transaction is represented by

$$\begin{gathered} D_l = (e_{l,1}, e_{l,2}, \cdots, e_{l,|D_l|}), \\ \text{such that } t_{l+1,1} - t_{l,|D_l|} > W, \end{gathered} \tag{4.1}$$

where $t_{l+1,1}$ and $t_{l,|D_l|}$ are time stamps of $e_{l+1,1}$ in D_{l+1} and $e_{l,|D_l|}$ in D_l, respectively. In other words, the time gap between the first element of D_{l+1} and the last element of D_l should be larger than W.

An example is presented in Table 4.1 to demonstrate how an A&E log is divided. Based on the time stamps $t^{(e)}$ in the second column, the time differences between adjacent event occurrences are calculated and presented in the fourth column. The time window is set to $W = 600$s. Then, it is found that the 6th event has a large time gap of 605s with the 5th event, and thus the data is divided at the instant between the 5th and 6th events. Analogously, the second breaking point is found at the instance between the 8th and 9th

Table 4.1: An example of dividing an A&E log.

No.	Time stamp (s)	Event	Time gap (s)	Break
1	0	mode 2	0	
2	10	alarm 2	10	
3	30	mode 1	20	
4	40	alarm 1	10	
5	50	mode 3	10	
6	655	mode 1	605	yes
7	680	alarm 1	25	
8	720	alarm 3	40	
9	1380	mode 1	660	yes
10	1420	alarm 1	40	

Table 4.2: Example of a transactional database converted from the A&E log.

Transactions	Events
1	mode 2, alarm 2, mode 1, alarm 1, mode 3
2	mode 1, alarm 1, alarm 3
3	mode 1, alarm 1

events, as the two events has a time gap of 660s in between them. Finally, the A&E data is reorganized in a transactional database format as shown in Table 4.2.

However, in real industrial data, there usually exist chattering alarms, which may lead to the difficulty in data segmentation. A chattering alarm is referred to as an alarm repeatedly transiting between the alarm and non-alarm states within a short time period [1]. Since the time gap between two adjacent occurrences of a chattering alarm is small, all chattering instants will be included in one transaction, which may lead to a large transaction depending on how long the alarm is in a chattering situation. If there exist a large number of chattering alarms, breaking the A&E log $\mathbb{E}$ into short transactions could be difficult. Thus, the chattering alarms should be reduced from $\mathbb{E}$.

Two typical techniques recommended in ANSI/ISA-18.2 [1] are off-delay and on-delay timers, which can be directly applied to the alarm data. In addition, a window filter can also be used to reduce chattering alarms; the method is formulated as follows: For an alarm a, its time vector is $\mathbf{t}^{(a)} = [t_1^{(a)}, t_2^{(a)}, \cdots, t_{|\mathbf{t}^{(a)}|}^{(a)}]^T$. A binary index vector $\mathbf{B}^{(a)} = [b_1^{(a)}, b_2^{(a)}, \cdots, b_{|\mathbf{t}^{(a)}|}^{(a)}]^T$ is obtained with each element given by

$$b_k^{(a)} = \begin{cases} 1, & \text{if } t_k^{(a)} - t_{k-1}^{(a)} < \omega, \\ 0, & \text{otherwise}, \end{cases} \tag{4.2}$$

where $k = 2, 3, \cdots, |\mathbf{t}^{(a)}|$; $b_1^{(a)} = 0$; ω is a pre-defined window size. Accordingly, the timed event (e, t) with $e = a, t = t_k^{(a)}, b_k^{(a)} = 1$ is removed from the A&E log $\mathbb{E}$. Since some operating modes are set to change automatically, it is possible that some mode events are repeating within a short period. Thus, the window filter technique can also be used to reduce such repeating events for operating modes.

Before the reconstructing of the A&E data $\mathbb{E}$ into transactional A&E data $\tilde{\mathbb{E}}$, modes and alarms with a frequency less than the required minimum threshold F_{th} are also pruned:

1. If there exists an unique mode m_i with a frequency less than F_{th}, i.e., $\sigma(m_i) < F_{th}$, the mode m_i should be pruned from the A&E log $\mathbb{E} = \{(e_k, t_k) : e_k \in \mathcal{M} \cup \mathcal{A}, t_k \in T, k = 1, 2, \cdots, N\}$

2. If there exists an unique alarm a_j with a frequency less than F_{th}, i.e., $\sigma(a_j) < F_{th}$, the alarm a_j should be pruned from the the A&E log $\mathbb{E} = \{(e_k, t_k) : e_k \in \mathcal{M} \cup \mathcal{A}, t_k \in T, k = 1, 2, \cdots, N\}$

Considering that if an event e that is either an unique mode $e = m_i$ or unique alarm $e = a_j$ has a frequency less than F_{th}, any association rule $\phi : M \Rightarrow A$ containing such an event would also have a frequency less than F_{th}, and such a rule would not be of interest since it is rarely seen in the historical database.

Further, in the reconstructed transactional A&E data $\tilde{\mathbb{E}} =< D_1, D_2, \cdots, D_L >$, there may exist some transactions which do not meet the needs for the mining of mode-dependent alarms, i.e., only the scenarios that the modes are followed by alarms are of interest. Thus, the transactional A&E data $\tilde{\mathbb{E}}$ is further processed by excluding and refining some undesirable transactions in the following scenarios:

1. If there exist alarm events before the first mode event in D_l, i.e., $e_{l,k} \in \mathcal{A}, k = 1, 2, \cdots, \tilde{k}$ and $e_{l,\tilde{k}+1} \in \mathcal{M}$, where $1 \leq \tilde{k} \leq |D_l| - 1$, the transaction D_l should be updated by $D_l = (e_{l,\tilde{k}+1}, e_{l,\tilde{k}+2}, \cdots, e_{l,|D_l|})$.

2. If there exist mode events after the last alarm event in D_l, i.e., $e_{l,k} \in \mathcal{M}, k = \tilde{k}, \tilde{k}+1, \cdots, |D_l|$ and $e_{l,\tilde{k}-1} \in \mathcal{A}$, where $2 \leq \tilde{k} \leq |D_l|$, the transaction D_l should be updated by $D_l = (e_{l,1}, e_{l,2}, \cdots, e_{l,\tilde{k}-1})$.

3. If all events in a transaction D_l are mode events ($e_{l,k} \in \mathcal{M}, k = 1, 2, \cdots, |D_l|$) or alarm events ($e_{l,k} \in \mathcal{A}, k = 1, 2, \cdots, |D_l|$, the transaction D_l should be excluded from the transactional A&E data $\tilde{\mathbb{E}}$.

As a result, any undesired and useless transactions or events are excluded, and a more refined transactional A&E dataset $\tilde{\mathbb{E}}$ is obtained for the mining of association rules in the next section.

4.2 Mining Association Rules Using FP-Growth

This section describes mining association rules from the preprocessed transactional A&E data $\tilde{\mathbb{E}}$. In [15], a classical data mining algorithm named Apriori was adopted to find frequent patterns that represented the concurrences of multiple operating modes and alarms. As discussed in Chapter 2, the Apriori algorithm detects frequent itemsets in a join-and-prune manner [14]. However, it needs to generate a large number of candidates and repeatedly scan the database, making the computation inefficient. Different from the method in [15], this thesis generates association rules for both single and multiple operating modes simultaneously, and does not need to mine frequent mode patterns in an individual step. Since both mode and alarm events are involved, there may exist long patterns, which may pose challenges to the computation. Therefore, a more efficient data mining method named FP-Growth (Frequent Pattern Growth) is proposed and applied in this thesis. This method detects frequent patterns without generating the expensive candidate itemsets or repeatedly scanning the entire database, and thus is more computationally efficient [16]. Then, based on the detected patterns, association rules are formed. According to [19], filtering association rules is considered as an ongoing challenge. In this thesis, a post-processing step is proposed to filter out undesirable association rules so as to output the rules of mode-dependent alarms of interest.

To apply the FP-Growth algorithm, Algorithm 1 is used to construct a compact data structure called FP-tree (frequent pattern tree) based on the sorted frequent-item list, which is collected and preprocessed from the transactional A&E data $\tilde{\mathbb{E}}$ as described above. Then, Algorithm 2 is used to mine the association rules $\phi : M \Rightarrow A$ directly from the FP-tree, which is an extended prefix-tree structure that stores the quantitative key information of association rules. The inputs of the FP-Growth are the transactional A&E data $\tilde{\mathbb{E}}$ and the support threshold F_{th}, and the output is the collection of frequent itemsets with supports no less than F_{th}. The support $\sigma(S)$ of an itemset S is calculated as

$$\sigma(S) = \left| \left\{ D_l : S \subseteq D_l, D_l \in \tilde{\mathbb{E}} \right\} \right|. \tag{4.3}$$

The detailed procedures of the FP-Growth algorithm can be found in [16].

Algorithm 1 FP-tree Construction

Input: A transaction database DB and a minimum support threshold, ϵ.
Output: FP-tree.
Method: The FP-tree is constructed as follows:
1. Scan DB once to collect F, the set of frequent items, and the support of each F. Sort F in descending order of support as the list of frequent items, L.
2. Create T, the root of an FP-tree, labelled as "null".
for each transaction, $Trans$ in DB **do**
    3. Select F in $Trans$ and sort them according to the order of L. Let $[p|P]$ be the sorted frequent-item list in $Trans$, where p is the first element and P is the remaining list. Call insert tree($[p|P], T$).
    4. The insert tree($[p|P], T$) function is presented as the following:
    if T has a child $N|N$.item-name $=$ p.item-name **then**
        increment N's count by 1;
    else
        create a new node N, with its count initialized to 1, its parent link to be linked to T, and its node-link to be linked
    end if
end for

Following the hypothesis test for the conditional probability in [15], the association rules can be determined by calculating a logarithmic likelihood ratio Λ, which is asymptotically χ^2 distributed with one degree of freedom. Given two frequent itemsets S_1 and S_2 such that $\sigma(S_1) \geq F_{th}$, $\sigma(S_2) \geq F_{th}$ and $S_1 \cap S_2 \neq \emptyset$, the logarithmic likelihood ratio Λ for a candidate association

Algorithm 2 FP-Growth - Pattern Mining.

Input: FP-tree constructed in Algorithm 1, and a minimum support threshold, ϵ.
Output: The complete set of frequent patterns.
Method: call FP-Growth(FP-tree, null).
Procedure FP-Growth(Tree, α)
{
if Tree contains only one path, P **then**
  for each combination, β of the nodes in P **do**
    Generate pattern $\beta \bigcup \alpha$ with support=min support of nodes in β
  end for
else
  for each α_i in the header of Tree **do**
    1. Generate pattern B=ai U a with support=support of ai
    2. Construct the conditional pattern base of B and then the conditional FP-tree of B, denoted as TreeB
    3. If TreeB !=0, then call FP-Growth(TreeB, B)
  end for
end if
}

rule $S_1 \Rightarrow S_2$ is

$$\begin{aligned} \Lambda \approx \quad & -2 \left[\sigma(S_1) \log \tfrac{\sigma(S_1)+\sigma(S_1 \cup S_2)}{2\sigma(S_1)} \right. \\ & \left. +\sigma(S_1 \cup S_2) \log \tfrac{\sigma(S_1)+\sigma(S_1 \cup S_2)}{2\sigma(S_1 \cup S_2)} \right], \end{aligned} \tag{4.4}$$

where $\sigma(S_1 \cup S_2)$ denotes the number of transactions containing both S_1 and S_2. If $\Lambda \leq \Lambda_{th}$, $S_1 \Rightarrow S_2$ is determined to be an association rule, where Λ_{th} indicates the χ^2 value at a certain significance level.

At a significance level of 0.05, or confidence level of 95%, the χ^2 value is about 3.841. Using $\Lambda_{th} = 3.841$ as an example, if $\Lambda > 3.841$, $S_1 \Rightarrow S_2$ is rejected as an association rule at a signficance level of 0.05, thus concluding $S_1 \nRightarrow S_2$. And if the reverse is true, i.e., $\Lambda \leq 3.841$, $S_1 \Rightarrow S_2$ is accepted as an association rule with a confidence level of 95%. It should be noted that the smaller the value of Λ, the higher the confidence level, making the association rule of $S_1 \Rightarrow S_2$ more significant.

4.3 Determination and Removal of Spurious and Redundant Rules

Using the FP-Growth algorithm, a complete set of association rules $\phi : S_1 \Rightarrow S_2$ are detected. However, not all rules are of interest. Thus, it is necessary to determine and filter out spurious and redundant rules, and reserve the rules that represent the mode-dependent alarms. The following filtering steps are involved:

1. An association rule $\phi : S_1 \Rightarrow S_2$ should be pruned if $S_1 \subseteq \mathcal{A}$ (i.e., S_1 contains only alarms) or $S_2 \subseteq \mathcal{M}$ (i.e., S_2 contains only modes), since it conflicts with the objective to detect mode-dependent alarms, i.e., $S_1 \subseteq \mathcal{M}$ and $S_2 \subseteq \mathcal{A}$.

2. An association rule $\phi : S_1 \Rightarrow S_2$ should be updated by $\phi : M_1 \Rightarrow S_2$ if $S_1 = M_1 \cup A_1$, $M_1 = \{m_i \in \mathcal{M} : i = 1, 2, \cdots, |M_1|\}$, $A_1 = \{a_j \in \mathcal{A} : j = 1, 2, \cdots, |A_1|\}$.

3. An association rule $\phi : S_1 \Rightarrow S_2$ should be updated by $\phi : S_1 \Rightarrow A_2$ if $S_2 = M_2 \cup A_2$, $M_2 = \{m_i \in \mathcal{M} : i = 1, 2, \cdots, |M_2|\}$, $A_2 = \{a_j \in \mathcal{A} : j = 1, 2, \cdots, |A_2|\}$.

4. An association rule $\phi : S_1 \Rightarrow S_2$ should be pruned if either S_1 or S_2 is an empty set, i.e., $S_1 \cup S_2 = \emptyset$

5. An association rule $\phi : S_1 \Rightarrow S_2$ such that $S_1 \subseteq \mathcal{M}$ and $S_2 \subseteq \mathcal{A}$ is pruned if S_2 containing 2 or more alarms, i.e., $|S_2| \geq 2$. In this case, a subset of this rule with $\phi : S_1 \Rightarrow S_2'$ where $S_2' \subseteq S_2$ and S_2' containing only a single alarm, i.e., $|S_2| = 1$ already exist, and the larger rule is redundant.

6. Given an association rule $\phi : S_1 \Rightarrow S_2$ such that $S_1 \subseteq \mathcal{M}$ and $S_2 \subseteq \mathcal{A}$, recalculate the support $\sigma(S_1 \cup S_2)$ by counting how many transactions have the property that S_1 is followed by S_2 within the time window W. If it is detected that $\sigma(S_1 \cup S_2) < F_{th}$ using W, the rule $\phi : S_1 \Rightarrow S_2$ is pruned. Due to the method for reorganization of A&E log $\mathbb{E}$ into transactional format $\tilde{\mathbb{E}} =< D_1, D_2, \cdots, D_L >$, each transaction's time period could be bigger than W, i.e., $t_{l,|D_l|} - t_{l,1} > W$, thus the time period of association rules computed from $\tilde{\mathbb{E}}$ could also be bigger than W, i.e., $t_{S_2,|S_2|} - t_{S_1,1} > W$. Hence, in order to ensure the association rule $\phi : S_1 \Rightarrow S_2$ is frequent within W as according to the problem formulated in Chapter 3, support was recalculated with W.

7. If association rules $\phi : S_1 \Rightarrow S_2$ and $\phi' : S_1' \Rightarrow S_2'$ have the same set of modes, i.e., $S_1 = S_1'$, then the rules are combined into one rule $\phi : S_1 \Rightarrow S_2 \cap S_2'$, each a_j with its own respective frequency.

Eventually, the spurious and redundant association rules are excluded from the list, and the remaining results are the rules of mode-dependent alarms that meet the requirements in the objective in Chapter 3.

4.4 Time Window

In the method of multiple mode detection used in the previous study in [15], frequent patterns of modes only $\phi : S_1 \Rightarrow S_2$, $S_1 \subseteq \mathcal{M}$, $S_2 \subseteq \mathcal{M}$ are first found using Apriori algorithm using the user-defined time window W_{th} and frequency threshold F_{th}. Then, using the last timestamp of the modes $t^{(m_i)}_{|t^{(m_i)}|}$ within the frequent mode-mode pattern, and alarm data $\mathcal{A}$, associations between multiple modes and their consequential alarms $\phi : S_1 \Rightarrow S_2$, $S_1 \subseteq \mathcal{M}$, $S_2 \subseteq \mathcal{A}$ are found. Instead of this approach of detection which first identify frequent patterns of multiple modes, then find associations of identified frequent multiple modes pattern with alarms, the proposed method in this thesis achieved the same tasks in one step.

In the proposed method, all mode $\mathcal{M}$ and alarm data $\mathcal{A}$ are inputted into the FP-Growth algorithm together and a total time window W is used, rather than using the same time window W_{th} twice. In the previous study, the maximum time between the first and last modes is W_{th}, the maximum time between last mode and the last alarm is also W_{th}. In the new method, there is only one user-defined time window W as the maximum allowed time between the first mode and the last alarm.

Different industrial plants and their respective processes may have varying residence time. Process knowledge is typically required to learn the residence time of various processes flowing through multiple units depending on product phases (such as steam, liquid or solid). Thus, a time window spanning from first mode to last alarm can reduce errors and avoid missing useful association of modes and alarms due to lack of process knowledge, and can potentially cover more and different variety of association rules.

Two examples are presented in Figure 4.1 and Figure 4.2 for comparison. Suppose that W_{th} in the previous method [15] is set at 5 min, and W in the new method is set at 10 min. If mode 1 is at 9:00 am, mode 2 is at 9:02 am, and alarm 1 is at 9:07 am, then both methods will identify the association between mode 1, mode 2 and alarm 1 as a multiple mode association rule, as shown in Figure 4.1. However, if mode 1 is at 9:00 am, mode 2 is at time 9:06 am, and alarm 1 is at 9:09 am, then the new method will identify mode 1, mode 2 and alarm 1 as a multiple mode association rule, but the previous method will not, as the time between the 2 modes are longer than W_{th} in the previous method, as shown in Figure 4.2.

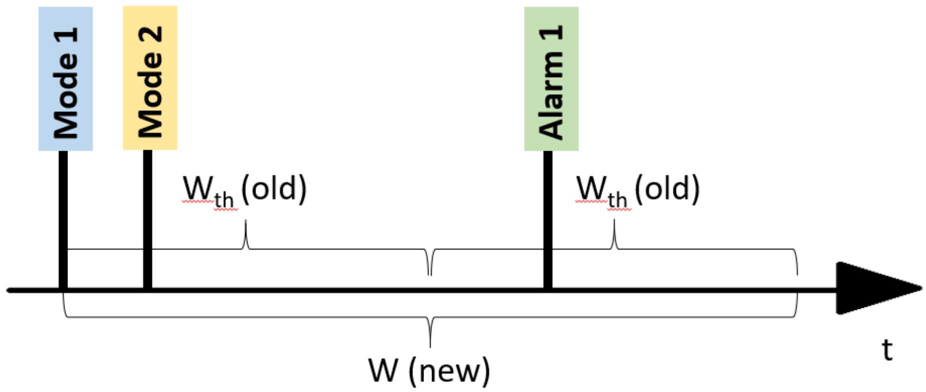

Figure 4.1: Time window example 1.

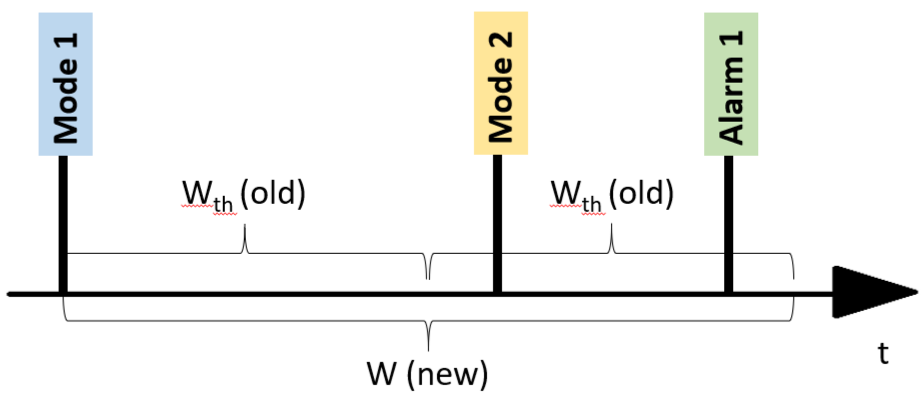

Figure 4.2: Time window example 2.

In summary, the proposed method in this thesis uses only one step to detect association rules of both single and multiple modes whereas the method in [15] requires two steps for multiple mode detection. In the previous method, the same time window W_{th} is used twice, first as the maximum time between the first and last modes, then as the maximum time between the last mode and the last alarm. However, in the proposed method, a total time window W of maximum time between the first mode and the last alarm is used to find association rules of mode dependent alarms instead. This new approach can reduce errors due to lack of process knowledge in residence time of various processes, covering more and a variety of association rules, as shown in the examples provided.

Chapter 5

Industrial Case Study

This chapter presents two industrial case studies to demonstrate the effectiveness of the proposed method of mode-dependent alarm detection from A&E logs.

5.1 Case Study I

The historical A&E data records were extracted from a large-scale process facility with over six consecutive months of data. The data set included 237 unique modes and 426 unique alarms. The total numbers of mode and alarm events were 17,389 and 288,716, respectively. The alarm rate during the six months period for the plant is calculated to be 11.1 alarms/10min, which is above the 1 alarm/10 min benchmark from ISA [1].

To apply the proposed method, there are two key user-defined parameters, namely, the support threshold F_{th} and the time window W, which may affect the number of generated rules and their quality. Either a small F_{th} or a large W may lead to more association rules, but among them, some could be spurious rules. By contrast, a large F_{th} may produce less but frequent rules, and a small W may results in less but more significant association rules. Thus, to investigate how the two parameters affect the results, multiple simulations

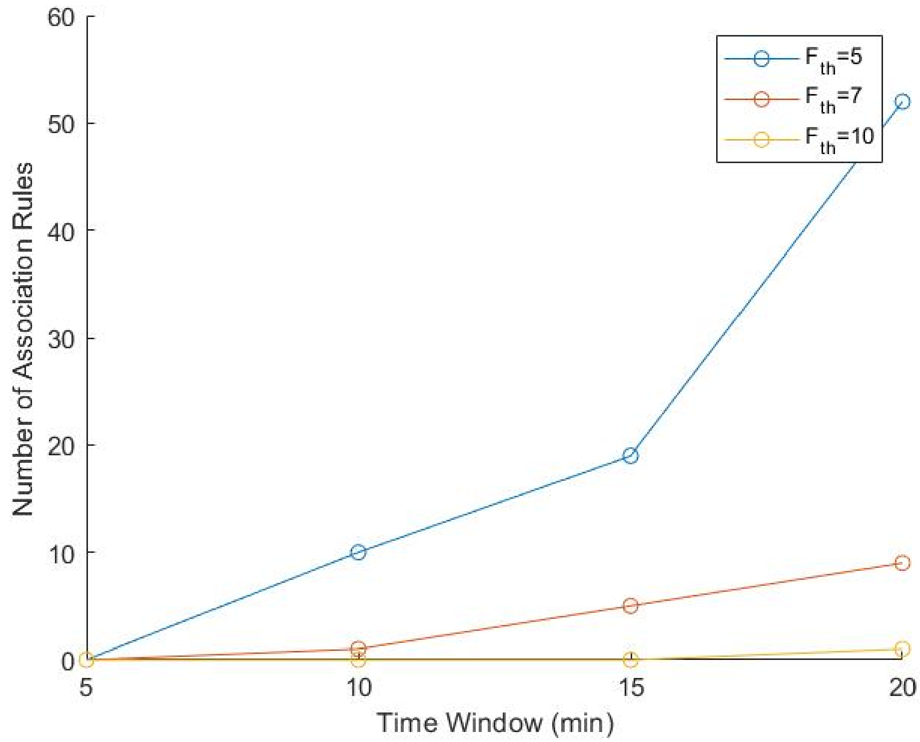

Figure 5.1: Numbers of association rules based on different W and F_{th} values.

were conducted with different combinations of W and F_{th}. The chi-squared value χ^2 of 3.841 was used; it corresponds to a significance level of 0.05 or a confidence level of 95%.

Figure 5.1 and Figure 5.2 show the number of detected association rules and the mean χ^2 value of the rules for each pair of W and F_{th}. As shown in Figure 5.1, the number of rules grows with the increasing of either W or the decreasing of F_{th}. In Figure 5.2, it can be seen that the mean χ^2 value mostly rises with the increasing of W; a smaller mean χ^2 value implies that the detected rules are more significant. The conclusion is coincidence with that in [15]. In practice, the two parameters must be carefully set, so as to balance the number and the quality of the detected association rules.

In the following study, the results with $F_{th} = 5$ and $W = 20$ are pre-

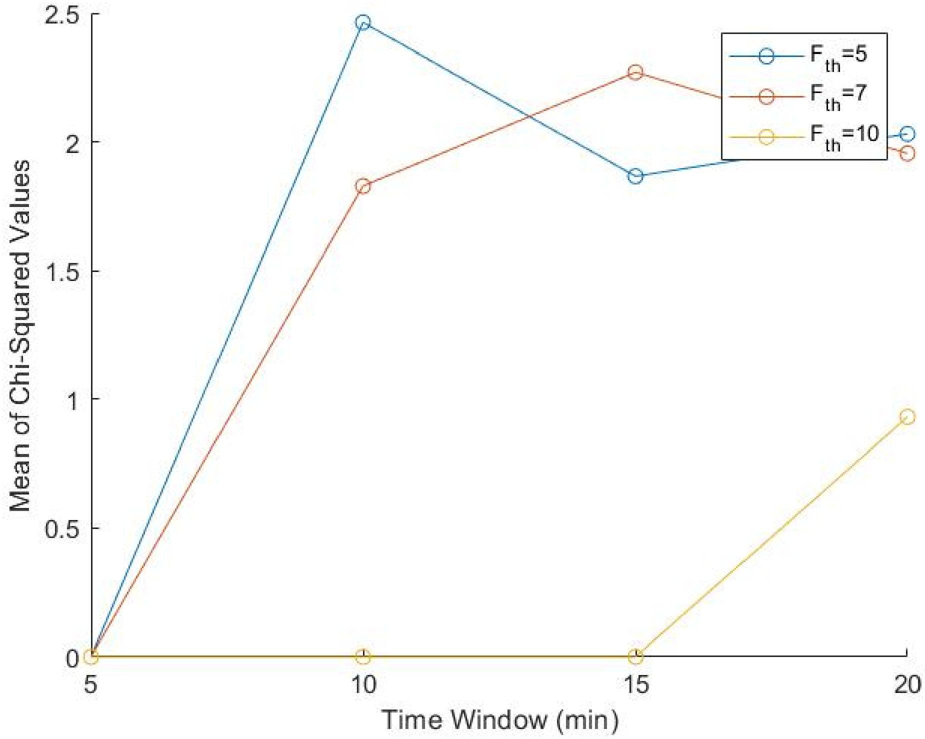

Figure 5.2: Mean χ^2 values based on different W and F_{th} values.

sented. For comparison, simulations were also conducted using the method in [15]. Table 5.1 presents the computation time of the proposed method and the method in [15]. It is obvious that the proposed method has a better computational efficiency. It is noteworthy that the detection of association rules for multiple modes accounts for the majority of the consumed computation time in [15]. Table 5.2 gives the number of detected association rules using the proposed method and the method in [15]. It can be seen that the method in [15] detected more rules for single modes but failed in identifying rules of multiple modes from the given A&E data. By contrast, the proposed method detected much more association rules of mode-dependent alarms, but the number of rules for single modes was less than that using the method in [15]. This is because some rules for single modes were contained in rules for

Table 5.1: Computation time of the proposed method and the method in [15].

	The method in [15]	The proposed method
Time	35006 s	1391 s

Table 5.2: Number of detected association rules in Case Study I using the proposed method and the method in [15].

	The method in [15]	The proposed method
1 Mode	10	5
2 Modes	0	19
3 Modes	0	24
4 Modes	0	2
5 Modes	0	2
Total Modes	10	52

multiple modes and thus were pruned.

To better understand the results, examples and discussions of both case studies will be presented together in Section 5.3.

5.2 Case Study II

Alarm and event (A&E) data for another large scale process facility was used to further analyze mined associations and verify the effectiveness and feasibility of the proposed method. The extracted A&E data from this facility includes 275 unique modes and 495 unique alarms. The total numbers of mode and alarm events were 8,939 and 21,009, respectively. The alarm rate is calculated to be 4.9 alarms/10min, which is above the 1 alarm/10 min benchmark from ISA[1].

To determine the two user-defined parameters, the support threshold F_{th} and the time window W, Case Study II followed the method used in Case

Table 5.3: Number of detected association rules in Case Study II using the proposed method and the method in [15].

	The method in [15]	The proposed method
1 Mode	3	9
2 Modes	0	35
3 Modes	0	26
4 Modes	0	9
Total Modes	3	79

Study I, and conducted multiple simulations with different combination of F_{th} and W, with a Chi-squared value χ^2 of 3.841. Through these simulations, it was found $F_{th} = 5$ and $W = 10$ are best fitted for the dataset in Case Study II; thus these values are used in the following study.

For comparison, simulations were conducted using the proposed method and the method in [15]. Table 5.3 shows the number of association rules detected using the proposed method and the method in [15]. It can be seen that the proposed method detected significantly more rules in both single and multiple modes, whereas the method in [15] failed to identify any multiple modes rules. Upon further investigation, it was found that all 3 single mode rules found using the method in [15] were also found using the proposed method.

To better understand the results, examples and discussion are presented in Section 5.3.

5.3 Discussions

Several technical documents were provided from the plants in both case studies, including: screenshots of HMI (Human Machine Interface) graphics as operator interface, process overview document which describes the main process streams, major equipment and the process steps with PFDs (Process Flow Diagrams), and P&IDs (Pipeline and Instrumentation Diagram) (provided for the plant in Case Study I only). By careful studying the process flows in these documents, and thorough examination on finding locations of equipment and/or controllers from which operator changes (mode changes) and their associated alarms were detected by our method, all of the results were checked and verified. Based on extent of relative distance of mode changes and resulting alarms, these rules were placed into 5 categories, used to summarize the different types of associations between modes changes and their consequential alarms, and the potential impact these results could bring to the control and alarm systems. The 5 categories are as follows:

- Category 0: modes and/or alarms could not be found in documents provided (only for Case Study II as P&IDs were not provided).
- Category 1: modes and alarms are in the same simple or complex control loop (for example: PID control loop as simple loop, and cascade control or override control as complex control loop).
- Category 2: mode changes caused alarms downstream or recovered stream in the same process stream (pipeline).
- Category 3: mode changes caused alarms in an adjacent process stream going into or coming out of the same vessel, or within the same unit.
  - In this case, alarms can be both upstream or downstream of the mode change.
  - As an example for alarms at downstream of the mode change: when an operator increased flow set point for a flow controller valve sit-

uated on an outlet stream of a vessel, it caused the vessel level low alarm.

  - As an example for alarm at upstream of the mode change: when an operator increased flow set point for a flow controller valve on a process stream feeding into a vessel caused a level high alarm of the vessel.

- Category 4: mode changes caused alarms in a downstream unit, or a upstream unit due to returned stream (100% returned like steam supply) or recovered stream (partially returned).

It should be noted that by a mode change, it can be either a digital change such as “open” and “close”, or an analog change such as change of SP (setpoint) or OP (output) of a controller, as discussed in Chapter 2. In the results of the two case studies, a majority of the modes are controllers, which means mode change can be change of state between “auto”, “manual” and “cascade”, or it can be SP or OP change. In this thesis, it is not specifically determined what particular type of change it was. However judging from the results and documents, it is most likely a majority of the mode changes for controllers were SP or OP changes.

The number of associations found in each category for Case Study I and Case Study II are presented in Table 5.4 and Table 5.5, respectively. It is noteworthy that the total number of associations is different from the total number of association rules, as each association is considered between 1 mode and 1 alarm, so a 2 modes mode-dependent alarm has 2 associations in its rule.

To further understand the categories, a few examples from both case studies are presented and discussed.

Table 5.4: Number of detected associations in each category in Case Study I.

Category	Number of Associations	Percentage
0	0	0%
1	14	16%
2	13	15%
3	27	32%
4	31	36%
Total	85	100%

Table 5.5: Number of detected associations in each category in Case Study II.

Category	Number of Associations	Percentage
0	15	14%
1	43	40%
2	26	24%
3	8	7%
4	15	14%
Total	107	100%

In Case Study I, it was detected that an alarm "Alarm 119" was a mode-dependent alarm of two operating modes "Mode 4" and "Mode 34", and such association occurs in 15 transactions. "Alarm 119" is a PV (Process Variable) low alarm of a pressure controller in a vessel, which means the pressure detected at this controller is lower than a preset alarm limit. "Mode 4" is the mode change of a pressure controller from an outlet stream of the same vessel for "Alarm 119", which places this association in Category 3. As "Mode 34" is the mode change on the same pressure controller as "Alarm 119", this association is one of simple control loop: the PID (proportional–integral–derivative) control in the controller itself, thus is placed in Category I.

As a second example for Case Study I, "Alarm 140" was detected to be a mode dependent alarm of "Mode 9" and "Mode 13". Such association occurs in 5 transactions in total. "Alarm 140" is a PV low alarm of a temperature controller situated on the top outlet of a vessel while "Mode 13" is the mode change of a flow controller on feeding stream line into the vessel. The temperature controller cascades to control the flow controller which means that the OP of the master temperature controller sets the SP of the slavery flow controller in a cascade mode. Therefore, this association was placed in Category 1. On the other hand, "Mode 9" is a hand switch valve in an upstream unit to "Alarm 140", thus placing it in Category 4.

For example 3 in Case Study I, it was detected that "Alarm 124" was a mode-dependent alarm of "Mode 13" and "Mode 33", and they appear together in 8 transactions. "Alarm 124" is a PV high alarm of a pressure controller. "Mode 13" and "Mode 33" are the mode changes of two flow controllers. The process stream "Mode 13" is on splits into two process streams where one has the flow controller with "Mode 33", and the other with "Alarm 124". As "Alarm 124" is situated downstream of "Mode 13" within the same process line/section, thus placing it in Category 2. When an operator decreases the feed flow towards the process stream that "Mode 33" is on, excessive flow downstream, thus leads to "Alarm 124". As such, this association is placed in Category 3.

In Case Study II, it was detected that both alarms "Alarm 199" and "Alarm 238" were mode-dependent alarms of four operating modes "Mode 9", "Mode 16", "Mode 42" and "Mode 43", and they appear together in 5 transactions. Figure 5.3 shows a diagram of the relations of these modes and alarms. "Alarm 199" is a flow low low differential alarm, which happens when any 2 out of the 3 flow indicators on the second branch line has a flow difference greater than a preset value. "Alarm 238" at the same location is a flow low low alarm. As mentioned, there are 3 flow indicators, the alarm uses a 2 out of 3 voting logic, which means if 2 out of the 3 flow indicators are having an alarm, "Alarm 238" will also alarm. This type of logic is used to ensure the accuracy of highly important control, commonly used in the SIS (Safety Instrument System). "Mode 9" and "Mode 16" are flow controllers, and "Mode 42" and "Mode 43" are hand switch valves. As outputs from the flow controllers increase and the hand switch valves open, the flow on their process streams also increases. This leads to an increased flow to unit 1 and unit 2. To ensure chemicals are distributed to the 4 units equally, less chemicals are pumped into line 1 and line 2, causing the two alarms. As they are all located at the same part of process lines, these associations in this example fall into Category 2.

The second example of Case Study II is an association rule between "Mode 9", "Mode 16", "Mode 42" and "Alarm 238", which appears in 6 transactions. It can be seen that this is a subset of the previous example, but it has a higher frequency. Depending on how much adjustment on each mode is made, any combination of these modes can lead to the same alarms. Thus, all association rules that are supersets and subsets of these two rules are kept in the final results to present to the plant engineers and operators, and they can determine which set of rules are more useful to them.

Delving into evaluation of these rules in different categories, we project that rules in Categories 3 and 4 could be more valuable because when the mode changes and alarms are further apart from each other, they often become harder to observe and associate. If proven useful by plant engineers and operators, these rules can be used to develop smart alarm suppression application or used for operation guidance and/or prediction. Different units

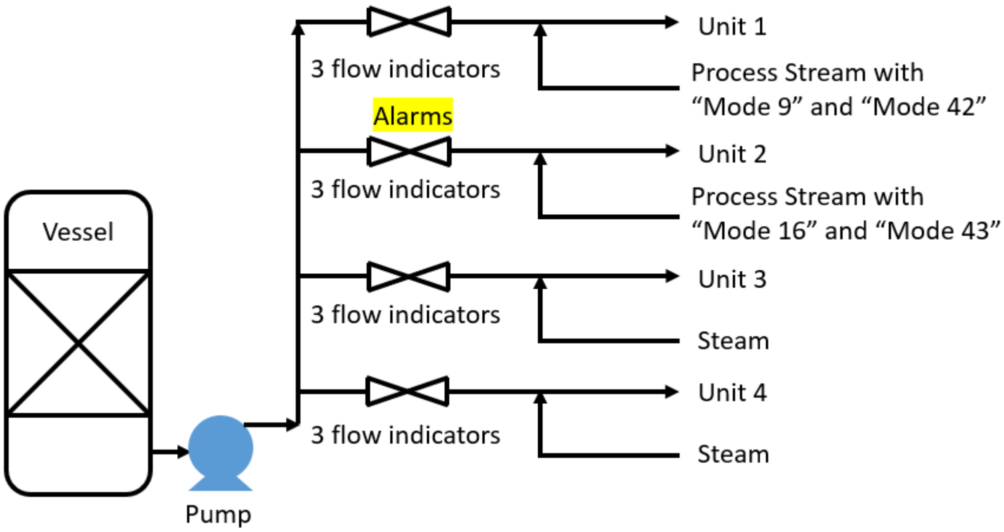

Figure 5.3: Diagram of example 1 of Case Study I.

are generally operated on separate console stations and by different operators. The rules in Category 4 where mode changes in one unit led to alarms in a downstream unit can be used to develop automatic messages to warn operators in downstream unit that an alarm is coming when the upstream unit operator makes mode changes.

In summary, the newly proposed FP-Growth model was developed and applied to industry alarm data sets in order to detect significant associations between operation mode changes and their consequential alarms. Two large A&E data sets provided by industrial partners were used to test and verify the effectiveness of the algorithm. The same two sets of data were also applied on Apriori model from [15] for performance comparison with the FP-Growth model. Results from both models indicated that the FP-Growth model proposed in this thesis discovered many more associations (in both single and mutiple modes) than using the Apriori algorithm where only single mode associations were detected. To evaluate and verify the accuracy and effectiveness of the model, The detected rules (52 in Case Study I and 79 in Case Study II) were divided into 5 categories based on the relative space of modes and alarms and features of their related control loops. On top of the discussed examples as samples of each category in this section, all generated rules are examined through provided documents. It has been proved that all rules detected in Case Study I are 100% accurate with true meaningful relationships between modes and alarms. Additionally, all rules detected as Categories 1 to 4 in Case Study II are accurate with true meaningful relationships as indicated by examples while rules in Category 0 cannot be found due to insufficient documentation.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis proposed an efficient method to automatically discover association rules of mode-dependent alarms from historical A&E logs. The proposed method consists of three major steps. Initially, the preprocessing step reorganizes the data into the required transactional format, and further refines the transactions to meet the objective set in Chapter 3. Then, the mining step discovers all association rules using the FP-Grwoth algorithm. The logarithmic likelihood ratio was calculated and compared with the chi-squared value χ^2 of 3.841 to determine association rules with confidence level 95%. Finally, the post-processing step screens the results to discover mode-dependent alarms that meet the requirements of the problem formulated. Compared to [15], the proposed method combines the detection for both single and multiple modes, therefore is more straightforward and has much better computational efficiency. Furthermore, additional mode-dependent alarms were detected. This was demonstrated by two industrial case studies. Using documents provided by the plant engineers, results were examined and validated, and potential opportunities to improve alarm management performance from these results were discussed. Upon verification by plant engineers and operators, the detected results can be used to assist these industrial practitioners in developing smart alarm suppression applications and/or using them to configure state based alarming modules.

6.2 Future Work

Several possible future work based on the study in this thesis are summarized as follows:

1. This thesis focuses on mining associations between operating modes and their resulting alarms. However, the reverse associations can also be found using the same or similar methods. The associations between alarms annunciated and operators' response and actions to an alarm or a set of alarm can be a study of great interest. Such results can be used to train new operators, whether it is set up as a guide, or even a simulation course for new operators to complete as part of on-boarding training. Furthermore, if detected associations are proved to be significant, they can be included in online document such as AOA (alarm objective analysis) which covers cause, consequence of an alarm, and operation action. Such information can be very useful as a daily operation support document. Eventually, it may also be possible to replace operator actions that are proven to be repetitive by functions and control logics on the control system, allowing operators to focus on performing activities that require more human factors and cannot be done by a computer.

2. Similarly, associations between a group of more than two alarms can also be a study of interest. The results can be used to determine whether they are potentially nuisance alarms like correlated alarms or redundant

alarms, and suppress or remove as needed.

3. Similar to the study in this thesis, the same method can also be used to determine if one alarm in a group of alarms is always the first, and the rest are its consequential alarms. This can be used to potentially identify the root cause of the issue that triggered this group of alarms. The results can also be used to program first out alarming logics in the PLC (Programmable Logic Controller), where the first alarm is determined and the information sent to the DCS (Distributed Control System) to be presented on the HMI (Human Machine Interface), and the consequential alarms potentially suppressed based on operation needs.

Bibliography

[1] Management of Alarm Systems for the Process Industries, document ISA 18.02, ISA (International Society of Automation), Durham, NC, USA, 2009.

[2] P. Goel, A. Datta and M. Mannan, "Industrial alarm systems: Challenges and opportunities," Journal of Loss Prevention in the Process Industries, vol. 50, pp. 23-36, 2017.

[3] W. Hu, J. Wang, and T. Chen, "A local alignment approach to similarity analysis of industrial alarm flood sequences," Control Engineering Practice, vol. 55, pp. 13-25, Oct. 2016.

[4] D. Beebe, S. Ferrer, and D. Logerot, "The connection of peak alarm rates to plant incidents and what you can do to minimize," Process Safety Progress, vol. 32, no. 1, pp. 72-77, Nov. 2012.

[5] Y. Luo, B. Gopaluni, Y. Xu, L. Cao and Q. Zhu, "A novel approach to alarm causality Analysis Using Active Dynamic Transfer Entropy," Industrial & Engineering Chemistry Research, vol. 59, no. 18, pp. 8661-8673, 2020.

[6] J. Wang, H. Li, J. Huang and C. Su, "Association rules mining based analysis of consequential alarm sequences in chemical processes," Journal of Loss Prevention in the Process Industries, vol. 41, pp. 178-185, 2016.

[7] S. Cai, L. Zhang, A. Palazoglu and J. Hu, "Clustering analysis of process alarms using word embedding," Journal of Process Control, vol. 83, pp. 11-19, 2019.

[8] B. Yang, J. Li, C. Qi, H. Li and Y. He, "Novel correlation Aanalysis of alarms based on block matching similarities," Industrial & Engineering Chemistry Research, vol. 58, no. 22, pp. 9465-9472, 2019.

[9] W. Hu, T. Chen, and S. L. Shah, "Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods," IEEE Transactions on Industrial Electronics, vol. 65, no. 9, pp. 7290-7300, 2018.

[10] J. Shang and T. Chen, "Early classification of alarm floods via exponentially attenuated component analysis," IEEE Transactions on Industrial Electronics, vol. 67, no. 10, pp. 8702-8712, 2020.

[11] Z. Mannani, I. Izadi and N. Ghadiri, "Preprocessing of alarm data for data mining," Industrial & Engineering Chemistry Research, vol. 58, no. 26, pp. 11261-11274, 2019.

[12] D.H. Rothenberg, Alarm Management for Process Control: A Best-practice Guide for Design, Implementation, and Use of Industrial Alarm Systems. New York: Momentum Press, 2009.

[13] B. R. Hollifield and E. Habibi, Alarm Management: Seven Effective Methods for Optimum Performance. Research Triangle Park, NC, USA: ISA, 2007.

[14] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in Proc. 20th Int. Conf. Very Large Data Bases (VLDB), pp. 487-499, 1994.

[15] W. Hu, T. Chen, and S. L. Shah, "Discovering association rules of mode-dependent alarms from alarm and event logs," IEEE Transactions on Control Systems Technology, vol. 26, no. 3, pp. 971-983, 2018.

[16] J. Han, J. Pei and Y. Yin, "Mining frequent patterns without candidate generation," ACM SIGMOD Record, vol. 29, no. 2, pp. 1-12, 2000.

[17] R. Agarwal, C. Aggarwal and V. Prasad, "A tree projection algorithm for generation of frequent item sets," Journal of Parallel and Distributed Computing, vol. 61, no. 3, pp. 350-371, 2001.

[18] J. Heaton, "Comparing dataset characteristics that favor the Apriori, Eclat or FP-Growth frequent itemset mining algorithms," IEEE South East Conference, Norfolk, VA, April 2016.

[19] C. Borgelt, "Frequent item set mining," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, no. 6, pp. 437-456, 2012.

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

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

[22] J. Wang, F. Yang, T. Chen and S.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, 2015.

[23] "Deepwater Horizon oil spill | Summary, Effects, Cause, Clean

Up, & Facts." https://www.britannica.com/event/Deepwater-Horizon-oil-spill. Accessed: 2020-12-05.

[24] "BP America Refinery Explosion | CSB." https://www.csb.gov/bp-america-refinery-explosion. Accessed: 2020-12-05.

[25] The Explosion and Fires at the Texaco Refinery, Milford Haven, 24 July 1994: A Report of the Investigation by the Health and Safety Executive into the Explosion and Fires on the Pembroke Cracking Company Plant at the Texaco Refinery, Milford Haven on 24 July 1994, Health and Safety Executive, 1997.

[26] N. Tamascelli, T. Arslan, S. L. Shah, N. Paltrinieri, & V. Cozzani, "A machine learning approach to predict chattering alarms," Chemical Engineering Transactions, vol. 82, p. 187-192, 2020.

[27] N. Tamascelli, N. Paltrinieri, and V. Cozzani, "Predicting chattering alarms: A machine Learning approach," Computers & Chemical Engineering, vol. 143, p. 107122, 2020.

[28] Y. Sun, W. Tan and T. Chen, "A method to remove chattering alarms using median filters," ISA Transactions, vol. 73, pp. 201-207, 2018.

[29] Z. Wang, X. Bai, J. Wang and Z. Yang, "Indexing and designing deadbands for industrial alarm signals," IEEE Transactions on Industrial Electronics, vol. 66, no. 10, pp. 8093-8103, 2019.

[30] R. He, G. Chen, S. Sun, C. Dong and S. Jiang, "Attention-based long short-term memory method for alarm root-cause diagnosis in chemical processes," Industrial & Engineering Chemistry Research, vol. 59, no. 25, pp. 11559-11569, 2020.

[31] C. Guo, W. Hu, S. Lai, F. Yang and T. Chen, “An accelerated alignment method for analyzing time sequences of industrial alarm floods,” Journal of Process Control, vol. 57, pp. 102-115, 2017.

[32] T. Niyazmand and I. Izadi, “Pattern mining in alarm flood sequences using a modified PrefixSpan algorithm,” ISA Transactions, vol. 90, pp. 287-293, 2019.

[33] Y. Xu, J. Wang and Y. Yu, “Alarm event prediction from historical alarm flood sequences based on bayesian estimators,” IEEE Transactions on Automation Science and Engineering, vol. 17, no. 2, pp. 1070-1075, 2020.

[34] S. Lai, F. Yang and T. Chen, “Online pattern matching and prediction of incoming alarm floods,” Journal of Process Control, vol. 56, pp. 69-78, 2017.

[35] M. Lucke, M. Chioua, C. Grimholt, M. Hollender and N. Thornhill, “Advances in alarm data analysis with a practical application to online alarm flood classification,” Journal of Process Control, vol. 79, pp. 56-71, 2019.

[36] J. Wang, Y. Zhao and Z. Bi, “Criteria and Algorithms for Online and Offline Detections of Industrial Alarm Floods,” IEEE Transactions on Control Systems Technology, vol. 26, no. 5, pp. 1722-1731, 2018.

[37] Y.-H. Lee, H. D. Jin, and C. Han, “On-line process state classification for Adaptive monitoring,” Industrial & Engineering Chemistry Research, vol. 45, no. 9, pp. 3095-3107, Apr. 2006.

[38] S. Bhaumik, J. MacGowan, and V. Doraj, “Mode based alarm solutions at Syncrude Canada,” IFAC-PapersOnLine, vol. 48, no. 8, pp. 653-656, 2015.

[39] J. Zhu, Y. Shu, J. Zhao, and F. Yang, “A dynamic alarm management strategy for chemical process transitions,” Journal of Loss Prevention in the Process Industries, vol. 30, pp. 207-218, Jul. 2014.

[40] E. Jerhotova, M. Sikora, and P. Stluka, “Dynamic alarm management in next generation process control systems,” in Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services. Berlin Heidelberg: Springer, 2013.

[41] D. Vernon, C. Reising, and T. Montgomery, “Achieving effective alarm system performance: Results of ASM consortium benchmarking against the EEMUA guide for alarm systems,” The Proceedings of the 20th Annual CCPS International Conference, Atlanta, GA, 11-13 April 2005.

[42] Y. Hu and N. H. El-Farra, “Robust fault detection and monitoring of hybrid process systems with uncertain mode transitions,” AIChE Journal, vol. 57, no. 10, pp. 2783-2794, Dec. 2010.

[43] Y. Cui, J. Shi, and Z. Wang, “Multi-state adaptive BIT false alarm reduction under degradation process,” IEEE Transactions on Instrumentation and Measurement, vol. 64, no. 3, pp. 671-682, Mar. 2015.

[44] C. Nihlwing, and M. Kaarstad, “The development and usability test of a state based alarm system for a nuclear power plant simulator,” in 8th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, no. 3, pp. 1904-1913, 2012.

[45] L. D. Jensen, Dynamic Alarm Management on an Ethylene Plant, Honeywell Users Group, Nice, France, 1995.

[46] B. Parker, "How to avoid alarm overload with centralized alarm management," Power, vol. 154, no. 2, pp. 38-41, 2010.

[47] R. Agrawal, T. Imieliński and A. Swami, "Mining association rules between sets of items in large databases," ACM SIGMOD Record, vol. 22, no. 2, pp. 207-216, 1993.

[48] R. Z. I. Hussain and S. K. Srivatsa, "A study of different association rule rining techniques," International Journal of Computer Applications, vol. 108, no. 16, pp. 10-15, 2014.

[49] S. Kotsiantis and D. Kanellopoulos, "Association rules mining: a recent overview," GESTS International Transactions on Computer Science and Engineering, vol. 32, no. 1, pp. 71-82, 2006.