# DOKTORI (PhD) ÉRTEKEZÉS

Dörgő Gyula Ádám

Pannon Egyetem
2021

Machine learning techniques for alarm management

Thesis for obtaining a PhD degree in the Chemical Engineering and Material Sciences Doctoral School of the University of Pannonia

in the branch of Chemical Engineering Sciences

Written by Gyula Ádám Dörgő

Supervisors: Dr. János Abonyi, Dr. Ahmet Palazoglu

propose acceptance (yes / no)

............................
Dr. János Abonyi
(supervisor)

propose acceptance ([✓ yes] / no)

............................
Dr. Ahmet Palazoglu
(supervisor)

As reviewer, I propose acceptance of the thesis:

Name of Reviewer: …...................... …................ yes / no

............................
(reviewer)

Name of Reviewer: …...................... …................ yes / no

............................
(reviewer)

The PhD-candidate has achieved …..........% at the public discussion.

Veszprém,

............................
(Chairman of the Committee)

The grade of the PhD Diploma …..................................... (……. %)

Veszprém,

............................
(Chairman of UDHC)

Pannon Egyetem

Doktori (PhD) Értekezés

# Gépi tanulási technikák fejlesztése alarm managementben

DOI:10.18136/PE.2021.789

*Szerző:*
Dörgő Gyula Ádám

*Konzulensek:*
Prof. Palazoglu Ahmet
Prof. Dr. habil. Abonyi János

*Értekezés doktori (PhD) fokozat elnyerése érdekében*
*a **Pannon Egyetem***

Vegyészmérnöki- és Anyagtudományok
*Doktori Iskolájához tartozóan*

Folyamatmérnöki Intézeti Tanszék

Pannon Egyetem
2021

University of Pannonia

Doctoral (PhD)Thesis

# Machine learning techniques for alarm management

*Author:*
Gyula Ádám Dörgő

*Supervisors:*
Prof. Ahmet Palazoglu
Prof. Dr. habil. János Abonyi

*A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy*

*in the*

Doctoral School in Chemical Engineering and Material Sciences
*of University of Pannonia*

Department of Process Engineering

University of Pannonia
2021

*A kutatói hivatás szeretetével és tiszteletével ajánlom*
*e néhány fricskát mindazoknak,*
*akik már tudnak saját magukon nevetni.*
*Vagy legalábbis másokon.*

Dévényi Tibor

*With the love and respect of the research profession, I recommend*
*these few flicks to all those*
*who can already laugh at themselves.*
*Or at least on others.*

Tibor Dévényi

PANNON EGYETEM

# *Kivonat*

Mérnöki Kar
Folyamatmérnöki Intézeti Tanszék

Philosophiæ Doctor

**Gépi tanulási technikák fejlesztése alarm managementben**

írta: DÖRGŐ Gyula Ádám

A vegyipari üzemekben a biztonságos működés fenntartása során az operátorok a folyamatirányító rendszerek által adott vészjelzésekre támaszkodnak. A számítógépek terjedésével azonban a vészjelzések definíciója pusztán digitálisan történik és nem jár a fizikai installációhoz köthető többletköltségekkel. Ennek eredményeképpen megnövekedett a rossz minőségű jelzések száma, melyek túlterhelik a technológiát üzemeltető operátorokat. Ez a tendencia csak további lendületet kapott az Ipar 4.0 forradalom irányelveinek köszönhető robbanásszerű szenzorszám növekedéssel. Ennek eredményeképpen a jelzéseket tartalmazó alarm & event-log adatbázisok elemzése komplex adatelemzési feladattá vált, amelynek elsődleges célja az üzemeltetői munka támogatása. Dolgozatomban alarm management feladatok megoldására szolgáló gépi tanulási technikák gyűjteményét mutatom be. Első lépésben az alarm management rendszerek mérését tekintem át, és bemutatom hogyan lehet a manapság alkalmazott adatalapú mérőszámokat azok céljai alapján hét kategóriába sorolni. Ezt követően egy döntési fa alapú módszert javaslok a vészjelzések definiálására, mely alapján a vészjelzéseket kiváltó meghibásodások azonosítására informatív vészjelzések definiálhatók. Ezt követi az alarm & event-log adatbázisok gyakori szekvencia alapú elemzési módszertanának bemutatása. Multi-temporális gyakori szekvencia bányászat alapú megközelítést javaslok a folyamatok diszkrét esemény alapú azonosítására, majd az azonosított folyamat alapján az események előrejelzésére és a meghibásodások továbbgyűrűző hatásának elemzésére. Végezetül, szemléltetve a mélytanulás területének alarm management célú alkalmazási lehetőségeit, egy rekurrens neurális hálózat alapú osztályozási megoldást mutatok be, mely alkalmas a folyamatokban megjelenő meghibásodások azonosítására a meghibásodást követő vészjelzések alapján.

UNIVERSITY OF PANNONIA

# *Abstract*

Faculty of Engineering
Department of Process Engineering

Doctor of Philosophy

**Machine learning techniques for alarm management**

by Gyula Ádám DÖRGŐ

In modern industrial plants, operators maintain a safe operation based on the indications of the process control systems. However, as with the rise of computers, the definition of alarm messages is determined purely computationally and entails no significant costs at all, the operators of the process become overloaded by uninformative and low-quality alarm messages. This trend just gained momentum with the exploding number of sensors of the Industry 4.0 revolution. As a result, the analysis of industrial alarm & event-log databases has become a complex data analysis task aiming to determine meaningful alarms and the support of operator work. In the present thesis, a collection of machine learning techniques is introduced, presenting various possibilities for the reduction of operator workload. First, an overview of the evolution of alarm system performance metrics is presented, and the current data-based approaches are grouped into seven categories based on the goals associated with each metric. Then, a data-driven method based on decision trees is proposed for the design of alarm messages being informative for fault detection, which considers that the occurring alarm messages initially should be optimal for fault detection and identification. This is followed by the frequent sequence-based analysis of alarm & event-log datasets. A multi-temporal frequent sequence mining-based approach is proposed for the identification of the processes based on the occurring discrete events, for the prediction of future events and for the hierarchical detection of the spillover effect of malfunctions. Finally, illustrating the application possibilities of the more and more trending field of deep learning, a recurrent neural network-based classifier is proposed for alarm-based fault detection and isolation of chemical processes.

PANNONISCHE UNIVERSITÄT

# *Auszug*

Fakultät für Ingenieurwissenschaften
Abteilung für Verfahrenstechnik

Doktor der Philosophie

**Machine-Learning-Techniken für das Alarmmanagement**

von Gyula Ádám Dörgő

Bei einen moderne chemische Anlage kann eine Operator die sichere Betrieb mit Hilfe von die Alarme des Prozessleitsystems gewehrleisten. Heutzutage werden die tatsächliche Bedeutung von Alarmmeldungen mit Hilfe von Computern, rein rechnerisch bestimmt, ohne zusätzlichen physischen Installationen, welche zusatzkosten verursachen würden. Dadurch kommen aber immer häufiger Meldungen mit schlechten Qualität vor, welche die Operatoren überfordern können. Dadurch ist die Auswertung von Alarm & Event Logs einen komplexen Aufgabe geworden. Diese Datenanalyse sollte in erste Reihe die sicheren Betrieb unterstützen. In meinen Thesis werde ich eine Sammlung von Techniken des maschinellen Lernens vorgestellten, welche vereinfachen die Lösung von Alarmmanagement Aufgaben und damit reduziert den Workload von Operatoren. Als erstes gebe ich ein Überblick über die Entwicklung der Leistungskennzahlen von Alarmsystemen. Die aktuellen datenbasierten Kennzahlen werden basierend ihren Zwecken in sieben Kategorien eingeteilt. Danach wird ein auf Entscheidungsbäumen basierendes Verfahren vorgestellt. Diese Methode ermöglicht die Erstellung von Alarmmeldungen, die ausreichend informativ sind über die auslösenden Fehler. Darauf folgt die häufige sequenzbasierte Analyse von Alarm-&-Ereignisprotokoll-Datensätzen. Zur Identifikation der Prozesse anhand der auftretenden diskreten Ereignisse, zur Vorhersage zukünftiger Ereignisse und zur hierarchischen Detektion des Spillover-Effekts von Fehlfunktionen wird ein multitemporaler Frequent Sequence Mining-basierter Ansatz vorgeschlagen. Zur Veranschaulichung der Anwendungsmöglichkeiten des immer mehr im Trend liegenden Gebiets des Deep Learning wird schließlich ein rekurrenter neuralen Netzwerk-basierter Klassifikator zur alarmbasierten Fehlererkennung chemischer Prozesse vorgeschlagen.

# *Acknowledgements*

Above all, I am very grateful to my supervisors. Prof. Dr. habil. Janos Abonyi, with your constant momentum, you pushed me further and further on my path, even when I made it harder not seeing the light. I am unspeakably grateful for your support and encouragement throughout the years. Prof. Ahmet Palazoglu, thank you very much for undertaking my supervision and making my time possible at the University of California, Davis. Your comments, aspiration and thoroughness shaped me greatly and truly put my research work on another level.

I am also very grateful to the supporting professional community around me. My special thanks and gratefulness go out to Dr. András Dallos for setting me off on this rough path. The colleagues and friends at the Department of Process Engineering really made my time during the years of the doctoral training. Moreover, I need to say thanks to the colleagues at the MOL Group for making invaluable practical comments on my work. From the professional community, a supporting friend and former fellow PhD student, Dr. Tamás Ruppert, cannot be left out, thank you very much.

Family and Friends. I know that I can be unbearable when I work, thanks for not letting me down. I am not just thankful, but lucky as well to have so many and so good Friends. The joy you made helped me through the everyday ups and downs. My doctoral training would not have been possible without the unconditional love and support of my Family. And finally, thank you, Szabina.

*Dedicated to my Family and Friends. Because without them it wouldn't be fun at all.*

# Contents

# Chapter 1

# Introduction

In process control system-supported production plants, alarm messages are raised when a specific process variable exceeds its associated limits. According to the Engineering Equipment and Materials Users' Association (EEMUA) [1] the purpose of an alarm system is to redirect the operator's attention towards plant conditions requiring timely assessment or action. Accordingly, the process of alarm management means the efficient design, implementation, operation, and maintenance of industrial process alarms.

At the dawn of the process control systems, the deployment of a new alarm message required the physical implementation of a light or sound signal (lamp or horn) to the system. The definition of each alarm was thoroughly reasoned and considered. However, with the computer systems gaining more and more space in distributed control systems (DCS) or supervisory control and data acquisition (SCADA) systems, the definition of alarm messages is determined purely computationally and entails no significant costs at all. This resulted in an explosive increase in the number of defined alarm messages in the process control systems. As a consequence, multiple and often redundant messages are implemented and the operators have become overloaded by the received alarm messages as schematically depicted in Figure 1.1. The process engineers faced the problem of recursively analysing and fine-tuning the master alarm databases containing the alarm definitions and monitoring the alarm load through the alarm logs containing the alarms and operator actions that occurred during the production. And here comes the topic of the present thesis work to the picture: the analysis of these databases of enormous size without the application of advanced data analysis and machine learning techniques is a very time-consuming problem, requiring a significant human workload. In the following, the development of data analysis and machine learning methods for alarm management are discussed.

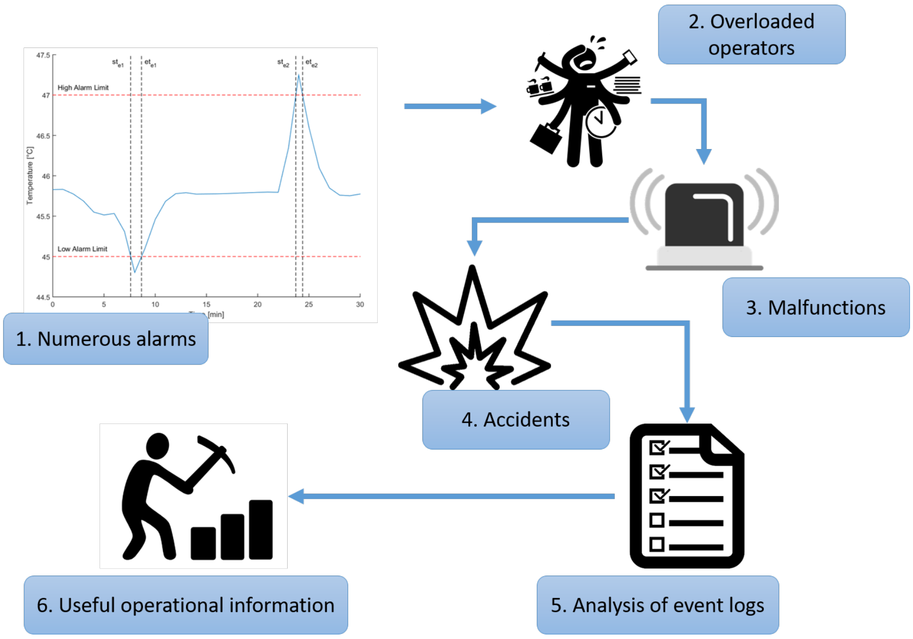


FIGURE 1.1: The process operators have become overloaded by uninformative alarm messages, turning an increased attention towards the process of alarm management. The core topic of this thesis is the development of data-driven solutions in process alarm management.

The application of data-related solutions is well reflected in the literature as well. Figure 1.2 provides a bird's eye view of the literature in the Scopus database with the search term "alarm management" and "machine learning" and "sequence" or "pattern" in the engineering, chemical engineering, energy and computer science domains with the omission of the articles from the medical and social sciences. The figure shows the co-occurrence of keywords in the found 71 articles. Two keywords are connected if they co-occur at least three times. The different colours indicate the clusters of the network, representing different data-related fields in alarm management: the blue cluster represents the field of machine learning, data visualization, forecasting of events, *etc.*, the red cluster illustrates the sequence-based representation of alarm floods and root cause analysis-related techniques, while in the green cluster, the topic is shifting towards the terms closer to industrial production plants. The yellow keywords form a collection of decision support-based terms, while the purple cluster provides more insight to the terms in connection with the modern aspects of networks and artificial intelligence.

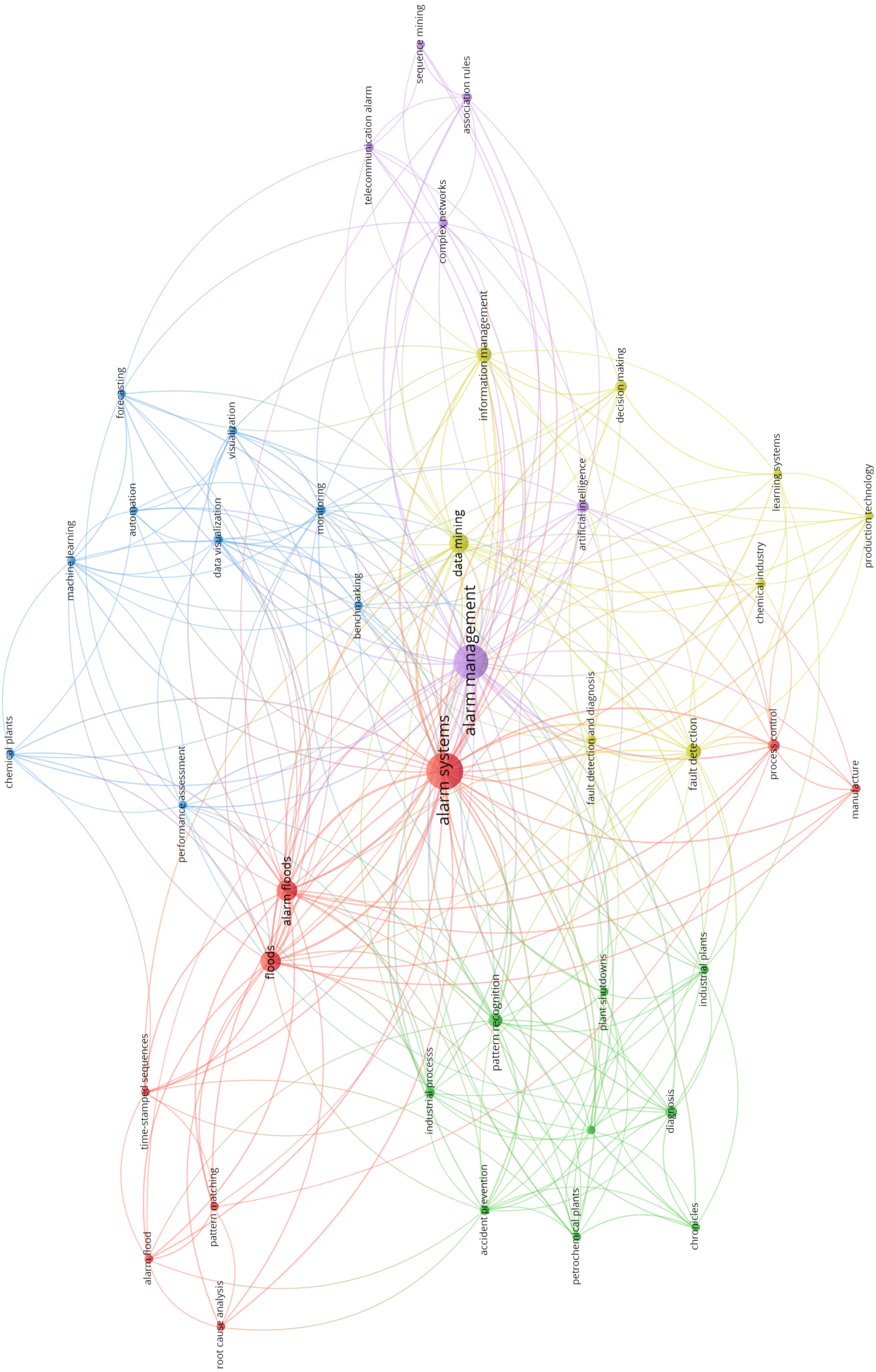


FIGURE 1.2: The co-occurrence of the keywords of the articles in the Scopus database for the search term "alarm management" and "machine learning" and "sequence" or "pattern" represents the data-based tasks in alarm management. Two keywords are connected in the network if they co-occur at least three times. (Accessed: May 11, 2021)

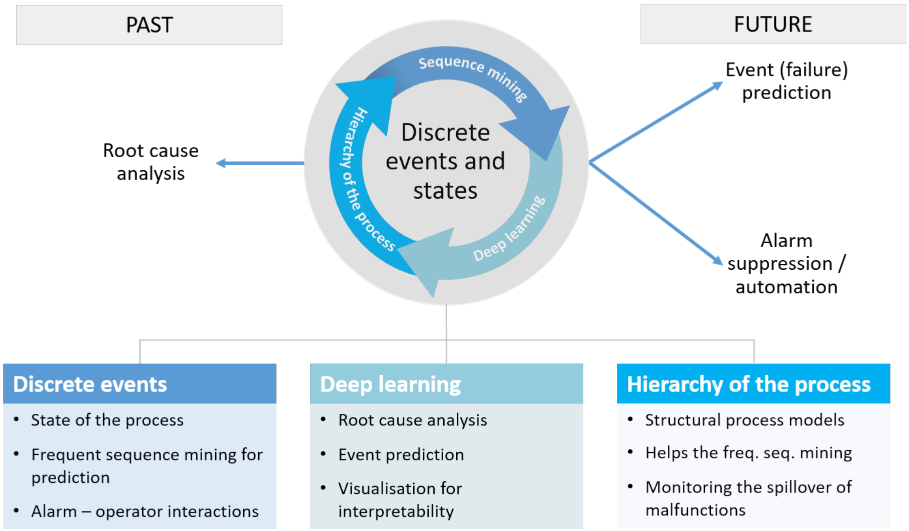

PAST
FUTURE
Sequence mining
Hierarchy of the process
Deep learning
Discrete events and states
Root cause analysis
Event (failure) prediction
Alarm suppression / automation
Discrete events
State of the process
Frequent sequence mining for prediction
Alarm – operator interactions
Deep learning
Root cause analysis
Event prediction
Visualisation for interpretability
Hierarchy of the process
Structural process models
Helps the freq. seq. mining
Monitoring the spillover of malfunctions

Figure 1.3: The schematic representation of the data-driven tasks of alarm management and the core focus points and driving forces of the present thesis.

Based on the literature overview, Figure 1.3 illustrates the widespread data-driven tasks in alarm management, whose development is also the main motivation and driving force of the present thesis. My focus is on the building of a model of the discrete events and states (represented mainly by alarm messages and operator actions) occurring in a chemical technology. During this task, different approaches and information are utilized: the frequent sequence-based model of discrete events, the currently trending field of deep learning and the application of traditional engineering knowledge in the form of process relevant information. Based on a well-functioning data-driven model built with these techniques, several tasks can be fulfilled. Starting with the supervision of alarm settings, the definition of alarm messages can be revised and optimized with more informative alarm thresholds, or the complex state of the process can be identified and similar alarm floods can be categorized. However, information anterior to the alarm messages can be deduced as well in the form of root cause analysis. On the other hand, predictions are possible as well, providing more time for the operators to avoid or prepare for a specific situation. Once the future alarm is known, assuming that it has no relevant information for the operators, it can be suppressed as well.

## 1.1 Research questions and thesis outline

The main research questions discussed in the present thesis are:

- *How to measure the performance of an alarm system and what aspects characterize a well functioning alarm system?*

  Comparing the alarm management guidelines in the past decades, despite numerous publications on the subject, little has changed regarding the metrics used to measure the alarm system performance. The more and more complex production plants and the evolving of alarm management tasks from simple operative ones to be an ever-present problem for both management and engineering teams gives immediate rise to the questions: what makes a good alarm system and how to measure or even point out the problems of an alarm system? In Chapter 2, I provide an overview of the different approaches for the assessment of an alarm system, propose a categorization of the different solutions and introduce the results of my investigations on the evolution of the different metrics.

- *How is it possible to define informative process alarms for alarm-based fault classification using decision trees?*

  The primary indicators for the operators of (chemical) technologies from faulty regions of operation are the alarm variables. Based on the assumption that the regimes of normal operation and different malfunctions should be distinguishable based purely on the alarm messages that have been raised, I formulate the problem of alarm limit definition as a data-based classification problem in Chapter 3. For the determination of the optimal alarm limits and description of the application of the generated alarm messages for fault classification, I propose a traditional machine learning technique, a decision tree-based solution.

- *Is it possible to represent the complex state of an alarm management system by the events recorded in the alarm & event-log database and apply this representation for prediction purposes?*

  The discrete events recorded in the alarm & event-log database are the primary indicators of conditions requiring timely assessment or action for the operators. Relying on the assumption that based on the analysis of these events, the main operation scenario should be traceable, in Chapter 4, I introduce a frequent sequence mining-based approach for the mining of frequently occurring alarm messages. Accompanying the identified frequent sequence models with a Bayesian probability-based mathematical description, I present how such models are appropriate for prediction purposes.

- *How can we incorporate the information on the hierarchical structure of the process into the data-driven alarm management techniques in order to facilitate the generation of more informative operational sequences?*

  Once I have presented how the frequent alarm sequences are capable of representation of the state of the process in a chemical production plant, a straightforward question is raised: How can we incorporate the vast knowledge of process engineers and operators into the data-driven analysis techniques? In Chapter 5, I present how the process hierarchy can be incorporated into the frequent sequence-based analysis of alarm management systems and how efficient this technique is in the detection of the spillover effect of malfunctions between different process units.

- *Can deep recurrent neural networks be applied for the extraction of hidden relationships of discrete events in large process datasets?*

  The potential of the recently trending field of deep learning for the support of the process industries is unquestionable. However, its application for fault diagnosis in alarm management and the additional advantages of such models were previously undefined. In Chapter 6, a novel deep recurrent neural network-based model for fault diagnosis is proposed, utilizing the occurring alarm messages as inputs of the model. Based on the weights of the hidden layers of the model, I also present how the similarity of alarm messages can be visualized and their interpretation is discussed.

Each chapter of the present thesis starts with an introduction to the discussed topic with the overview of the related literature. Then, the discussion of the applied methodology and tools are followed by the description of the applied dataset and the analytical results. The chapters are ended with concluding remarks and discussions. The chapters discussing the research works are followed by a conclusion section, discussing the general ideas and results. Finally, the thesis is ended by the thesis findings, containing the contributions to the topic of data-driven alarm management. The discussion of the vinyl acetate process simulation, used as a case study in several chapters throughout the thesis is presented in the Appendix.

# Chapter 2

# How to measure the performance of an alarm system? Quality vs. quantity of alarm messages

Despite significant efforts to measure and assess the performance of alarm systems, to this day, no silver bullet has been found. The majority of the existing standards and guidelines focus on the alarm load of the operators, either during normal or faulty plant conditions, and only a small fraction takes into consideration the actions performed by the operators. In this chapter, an overview of the evolution of alarm system performance metrics is presented and the current data-based approaches are grouped into seven categories based on the goals of and the methodologies associated with each metric. Deriving from the categorical overview, the terminological differences between the academic and industrial approaches of alarm system performance measurement are reflected. Moreover, I highlight how extremely unbalanced the performance measurement of alarm systems is towards quantitative metrics instead of focusing on qualitative assessment, invoking the threat of excessive alarm reductions resulting from such a unilateral approach. The critical aspects of qualitative performance measurement of alarm systems is demonstrated in terms of the comparison of the alarm system of an industrial hydrofluoric acid alkylation unit before and after the alarm rationalization process. The quality of the alarm messages is measured via their informativeness and actionability, in other words, how appropriate the parameter settings are for the everyday work and how actionable they are by the operators of the process.

## 2.1 Introduction

Possibly the poorest alarm message most people may encounter in their everyday lives is the "check engine" light of a motorized vehicle [2]. This alarm message can indicate the presence of a number of malfunctions, with a wide range of consequences. Moreover, the optimal actions taken by the driver are also undetermined: it can vary from pulling over and stopping the vehicle immediately to a fault that does not affect the safety of transportation and can be handled at the next maintenance opportunity. While having just a single alarm message may be a very good situation for a chemical plant operator who is often overloaded with lots of messages, it is clear that this is neither optimal nor realistic. In the present chapter, I overview the different approaches of measuring the performance of an alarm system and attempt to shift the practice from the quantitative aspect of alarm load analysis to the incorporation of quality-based solutions.

According to the definition of the Engineering Equipment and Materials Users Association (EEMUA) in the well-known industrial standard [3], the purpose of an alarm system is to direct the operator's attention towards plant conditions requiring timely assessment or action. However, as the cost of the software-based definition of alarms is negligible, the number of configured alarms has increased drastically due to intentions for enhanced productivity and safety. Moreover, the alarm messages of the more and more ubiquitous smart field devices generated another wave of alarm messages (many of which should be handled by maintenance rather than operations) [4]. The result is counterproductive, as the poor performance of alarm system can undermine both productivity and safety. Moreover, the performance of an alarm system decays over time [5], and needs to be continuously monitored by the responsible team. The above mentioned examples, the one with the bad configuration of alarm messages (*e.g.*, the check engine light) and the one explaining the problems of big alarm systems with decaying performance, highlight that during the past decades, the alarm management tasks have evolved from simple operative ones to be an ever-present problem for both management and engineering teams. The immediate questions that come to mind are: what makes a good alarm system and how to measure or even point out the problems of an alarm system?

Inquiry into these questions gained some momentum with the achievements of Industry 4.0 and the revolution in cyber-physical systems, as an adequate alarm management system is also a prerequisite for the development of high performance human-machine interfaces [6], which is the focus of many production companies. However, comparing the 1999 edition of the Engineering Equipment and Materials Users Association publication, the EEMUA-191 Alarm Systems - A Guide to design, Management and Procurement [7] and the IEC-626822014 Management of alarm systems for the process industries from 2014 [8], one sees that little has changed in the fifteen years between the two publications regarding the metrics used to measure the alarm system performance [9].

As a result, despite numerous publications on the subject, no universal solution has been found providing a simple and comprehensive answer to challenges in the measurement and comparison of different alarm systems. Moreover, even with the arrival and penetration of Industry 4.0 principles, the focal points of the measurement of alarm system performance remained unchanged. The present chapter provides an overview of the different approaches for the assessment of an alarm system, categorizes the different solutions and investigates the evolution of the different metrics. While this chapter aims at a general overview of existing approaches for data-based alarm system performance measurement and serves as a discussion starter in the topic, it also offers some novel contributions:

- A categorical analysis of the existing methods of data-based alarm system performance measurement is provided.
- The unbalanced approach of performance measurement metrics weighted heavily towards quantitative instead of qualitative ones is stressed.
- Two metrics are proposed for the measurement of the quality of the alarm messages in terms of their informativeness and actionability: the appropriateness of alarm parameter settings for everyday operations and the level of actionable alarms by the process operators.

The roadmap of the chapter is as follows. A summary of the background and evolution of alarm system performance metrics are provided in the next section

(Section 2.2). The categories of the data-based alarm system performance metrics are introduced in Section 2.3. The different approaches for the data-based measurement of alarm systems is presented through the comparison of the alarm system of an industrial hydrofluoric acid alkylation unit before and after the alarm rationalization process in Section 2.4. Next, discussion of the results, the outlook and future research challenges are provided in Section 2.5, followed by concluding remarks.

## 2.2 The trends of alarm system performance measurement

The fundamental step to achieve a good alarm system performance is to survey the existing alarm configuration, in other words, to collect data and benchmark the analysed system [2]. This involves the calculation of key performance indicators (KPIs) for the alarm system together with the assignment of target values, the reviews of alarm (and controller) configurations and the collection of operator feedback regarding past alarms and events [4]. However, the measurement of the performance of alarm systems and thus the measured and targeted KPIs are not universal as different approaches exist in the academia and the industry and the resulting metrics, together with the targeted values, differ not only between industrial sectors but between sites inside the sectors as well. It was no surprise that Goel *et al.* identified this as the key challenge in the performance measurement of alarm systems during their review of the challenges and opportunities of industrial alarm systems [10].

During the past few decades, several studies have been published on the measurement of alarm system performance. Just to mention a few now and leave the details for the latter part of the present chapter, several standards in alarm management mention the topic: all three editions of the EEMUA-191 standard (1999 [7], 2007 [3] and 2014 [1]), the work of the Norwegian Petroleum Directorate [11], the NAMUR standard from 2003 [12], as well as the editions of ANSI/ISA-18.2 (2009 [13] and 2016 [14]), its supporting document, the ISA TR18.2.1-2018 [15], the API RP 1167 ([16] and [17]) and the IEC-626822014 from 2014 [8]. Regarding academic publications, a very early article focusing on the industrial perspective is proposed by Nochur *et al.* in 2001 [18]. Later, the well-known work of Hollifield provided [2] a comprehensive guide to practical solutions in alarm management. Moreover, in the recent years two review articles were published on this topic, one discussing the challenges and opportunities of alarm systems [10] and the other focusing on the main causes of alarm overloading [19]. Finally, several approaches are present in the literature for the graphical assessment of alarm systems as well [20].

A good bird's eye view picture of the trends in alarm system performance monitoring can be obtained by the analysis of the number of academic and industrial publications on this topic. The middle bar chart of Figure 2.1 illustrates the number of scholarly works (orange) and patents (blue) published annually. The data was collected from Lens.org with the search keywords, Alarm system performance in the engineering field, (Accessed: January 25, 2021). The total number of scholarly works is naturally significantly higher than the number of patents in the topic, as not all research work leads to a successful industrial application. However, the trends actually also reveal a more interesting detail: as the number of published scholarly works is over its peak and turns into a decreasing trend in the recent years, the number of patents gained momentum with an increasing trend. This shift from academic research to industrial patents is a general trend in the case of technologies that turn from research into applied solution. However, the path to this increased industrial attention was paved by several unfortunate incidents fully or partly due to the faults of alarm management (top part of Figure 2.1) and the guidelines and standards published as an attempt to improve the safety of operations (below of the bar chart in Figure 2.1). The bottom of the figure illustrates a specific change in the field, the evolution of the average number of alarms/time unit type of measures, where the time unit used for the standardization of alarm numbers undergone a change from 10 minutes to first an hour, and later a day-based norm. Recently the monitoring of this daily average alarm rate is removed from the recommendations. The reason for this change is going to be discussed in detail in Section 2.3.2. The incidents and guidelines were adapted and re-edited from [21].

Regarding the topic of academic publications, a systematic examination of the literature of the chemical engineering domain in the Scopus database was carried out with the search term "alarm system performance". Moreover, as a significant amount of articles discussed the topic of blood insulin level related alarms, the articles containing the "blood" word were excluded from the results to narrow

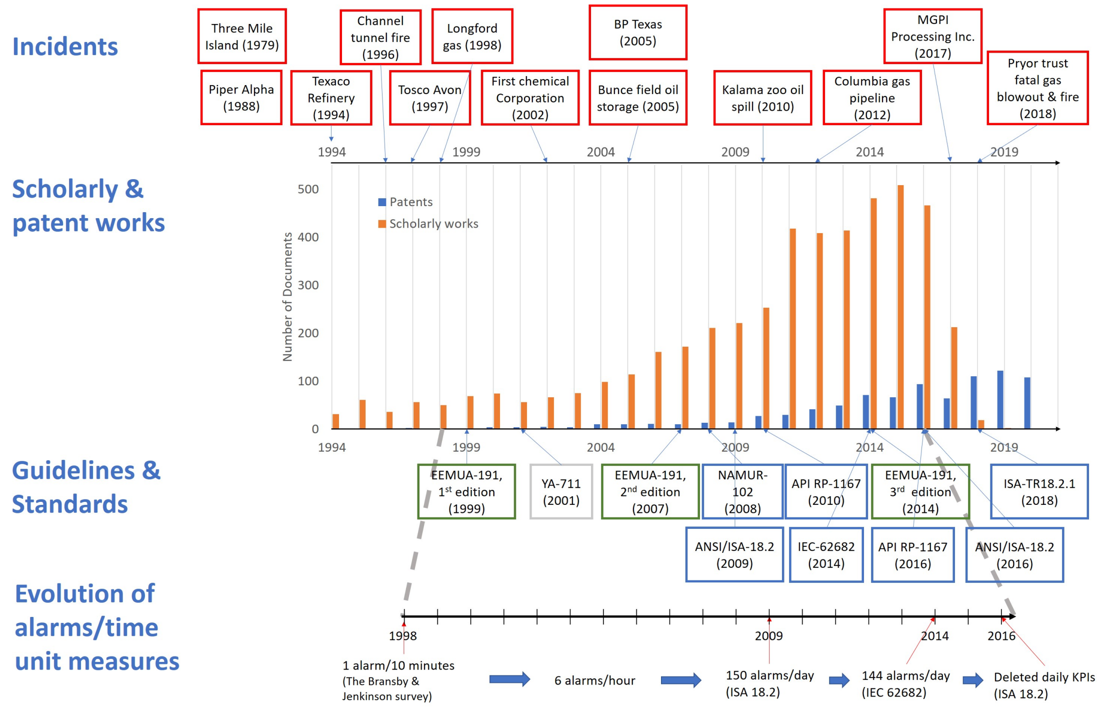


FIGURE 2.1: The research of alarm system performance more and more turns to industrial applications as the number of academic works published annually seems to be over its peak, giving space to the increasing number of patents in the field. However, this trend is noted by several unfortunate incidents fully or partly due to the faults of alarm management (top) and the guidelines and standards published as a result (below of the bar chart). The bottom of the figure illustrates a specific change in the field, the evolution of the average number of alarms/time unit type of measures.

down the topic to the chemical process control systems (search term: (alarm AND system AND performance) AND NOT (blood), Chemical engineering domain). Figure 2.2 illustrates the co-occurrences of the article keywords in the Scopus database. Two keywords are connected in the network if they co-occur among the keywords of the articles at least four times. The different colours indicate the different clusters in the network, which can indicate the major topics of alarm system performance related publications: hazards/safety (blue), human factors (yellow), performance monitoring (red) and fault detection (green).

According to the guidelines of the EEMUA [7], the performance measures of an alarm system can be used:

- as performance targets for acceptability criteria of a new alarm system,
- to assess the performance of an existing alarm system,

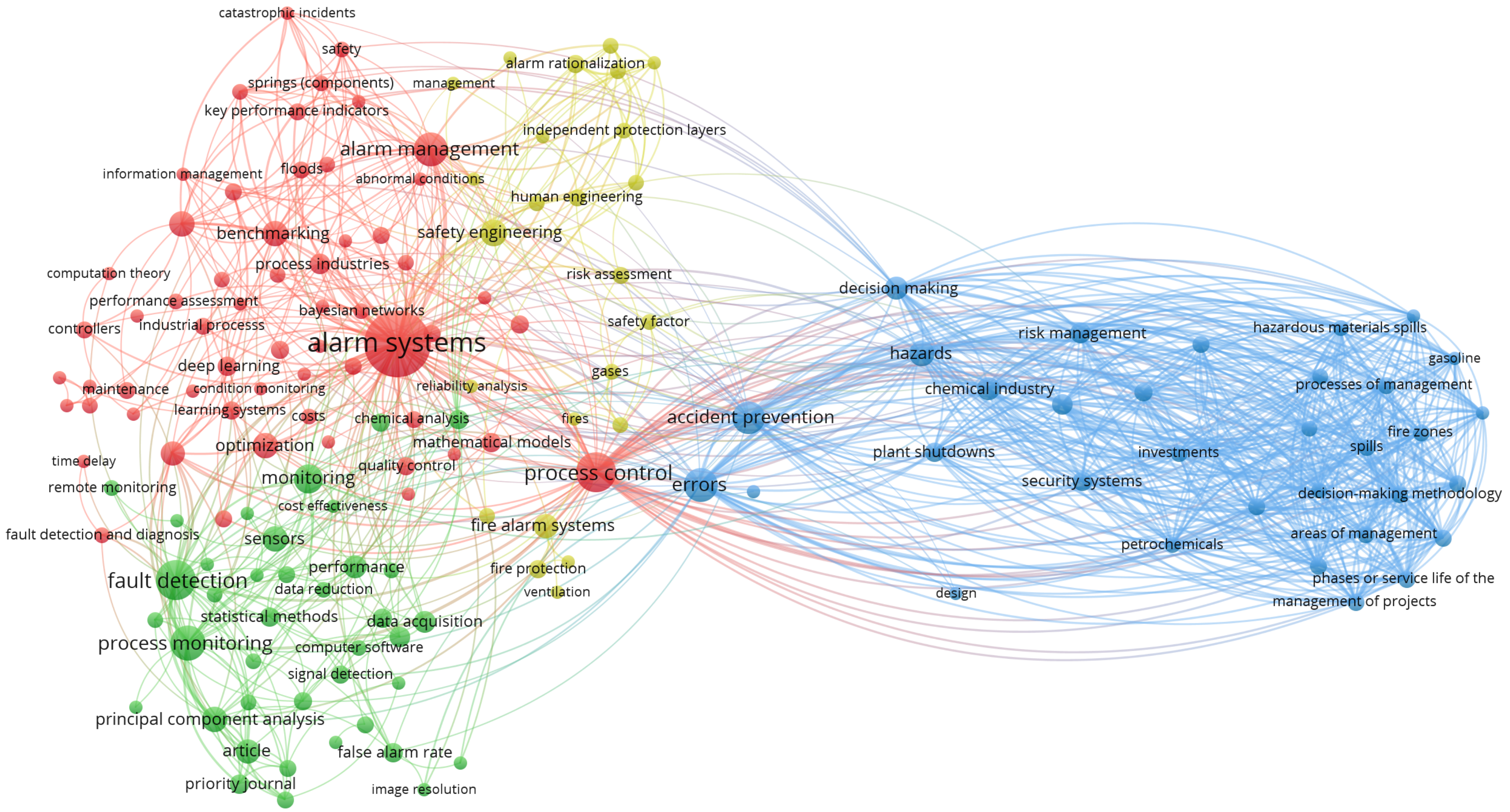


Figure 2.2: The co-occurrence of key words in the Scopus database for the search term "alarm system performance" limited to the chemical engineering domain excluding the "blood" word. Two keywords are connected in the network if they co-occur at least four times. (Accessed: January 06, 2021)

- as management tools to monitor the effectiveness of on-going improvement programs, and
- to identify specific nuisance alarms.

As it is apparent in the previous list of applications, the literature views the performance of an alarm system from different perspectives and hence, measure its efficiency with varying goals. This diversity motivated the present work, aiming to classify the different approaches of alarm system performance performance measurement into seven groups, which are related to the different tasks or applications of an alarm system.

Operational problems with the often used univariate data-based alarm strategies has been well-known for a long time [22]. As it was stated by Hollifield and Habibi [2], the measurement of the direct contribution of alarm management improvements to the plant-based performance indicators is problematic, as there are numerous other factors tightly related to the performance of a chemical plant, *e.g.*, the maintenance of the plant, its control system or the operator performance, among others. The effect of a well-functioning alarm system is more to be traced through the prevention and minimization of the impact of operational anomalies than through direct production improvement. Moreover, as in the present day control systems, complete automation is still a challenge in the chemical industry, and the process operator's increased ability to reduce the consequences of plant anomalies becomes the key element of a sound alarm system. While we have known for a long time "..that both objective and subjective measurements determine the performance level. Numbers alone do not; we engineers tend to focus on numbers!" [2], little has changed in the field of alarm system performance measurement to fully recognize this shortcoming.

This complex picture and the challenges associated with the direct measurement of alarm system performance constitute the key reasons why research in this area could not truly engage the industry and fell short of offering fundamental monitoring solutions. As a result, the industrial professionals face alone the problems of constantly improving and re-investigating the performance of alarm systems with the guidelines provided by industrial standards. The available performance indicators and solutions together with their target values are highly industry and case-specific, which makes the tasks of alarm management hard-to-automatise and extremely labor-intensive by the experts in the field.

## 2.3 Measurement of alarm system performance

This section introduces the key starting point of the present chapter, namely, the different approaches to the measurement of alarm system performance. There are various solutions available for the categorization of alarm system performance metrics, and the main directives are the following:

- The EEMUA standard identifies performance and design metrics, basically referring to alarm activation based metrics and the metrics describing the efforts to produce an alarm system capable of achieving the performance metrics in the other category [1]. These metrics are often referred to as dynamic (or activation) and static (or configuration) metrics, respectively, as well [23, 24]. Koene and Vedam [25] also divide the performance metrics of alarm systems to two categories, the static (or configuration) based metrics and the dynamic ones.
- The ANSI/ISA-18.2-2009 standard basically defines the same categories, however, approaches them from the type of data being evaluated: the first category deals with the alarm records (alarms produced during the operation), while the second category comprises the alarm attribute-based evaluation of a system (describing the underlying structure of alarm messages) [14].
- Another categorization of alarm performance metrics is the definition of the objective (quantitative) and subjective (qualitative) group of metrics [26]. The quantitative metrics can be calculated by algorithms and query tools based on the data recorded in the alarm & event log databases, whereas in the case of qualitative metrics, other factors, *e.g.*, operator workload, experience, environment and other areas that are difficult to assess are considered.

The shortfall of such generic metrics, however, is that they strongly vary between process plants, and they are influenced heavily by the complexity of the process itself and the level of the plant automation and instrumentation [25].

In the present chapter, a new categorization is offered, using seven groups distinguished by the aspects that the examination of the performance of the alarm system is based on: Load-based metrics (Section 2.3.2), Diagnostic metrics (Section 2.3.3), Deployment metrics (Section 2.3.4), Scaling metrics (Section 2.3.5), Audit metrics (Section 2.3.6), Design metrics (Section 2.3.7), and finally the Operator's perspective (Section 2.3.8). In the following subsections, each of these categories is introduced and discussed, and examples of the related metrics are provided.

### 2.3.1 Alarm attributes

The first task for the measurement of alarm system performance is the determination of what can be measured. Naturally, the performance of an alarm system can be most evidently captured using the attributes of an alarm message, let it be the timestamp of its occurrence, its priority, temporal length, *etc.* In this section, a brief overview of alarm attributes is provided, or rather, characteristics that can be assigned to an alarm message similar to [27]. Various attributes are directly assigned to alarm messages in industrial alarm systems and several more attributes or characteristics can be derived from the information systems related to the operation, *e.g.*, process layouts, process control systems, shift schedules and human resource databases, *etc.* Table 2.1 provides an overview of the most generally accepted alarm attributes that are often present in processing industries. As can be seen in the first column, the different data sources are divided by horizontal lines in the table. However, it is important to highlight that every industrial database is unique, therefore Table 2.1 only aims to provide a general picture of information content for alarm systems and the specific definition of the source of the information or its data format is only for illustration purposes.

The first group of attributes is often stored in the Alarm & Event Log database and contains the information closely related to the message presented to or performed by the operator in the case of an alarm event. Naturally, an alarm has a *tag* or *tag ID*, a unique identifier of the alarm message. The *units* of the alarmed variable indicate the unit of measure of the process variable, while the *alarm type* or interval identifier of the alarm attribute provides information on the level of the process variable related to the assigned alarm thresholds. Besides the presence of low (L) and high (H) alarms, several other levels can be defined, in most of the cases the more severe alarms are indicated as low-low (LL) or high-high ones (HH), but these definitions are often arbitrary. In support of the operator work, a brief *description* of the problem is provided as well. The actual alarm threshold set by the operators is also present in the database. The *event type* describes

| Source | Alarm attribute | | Example |
|---|---|---|---|
| Alarm & Event Log database | Tag | | 43TI2085, LIG502, |
| | Units | | °C, %, m3, - |
| | Alarm type | | LL, L, H, HH |
| | Alarm threshold (actual) | | 12, 0.4 |
| | Start time | | 2021-06-26T14:30:59+00:00 (timestamp) |
| | End time | | 2021-06-26T15:10:22+00:00 (timestamp) |
| | Description | | PV2085 temperature, T502 level |
| | Event type | | Alarm, Return to normal, Acknowledge, Operator action, Operator message, Ignore, Suppress, Unsuppress, Shelved, UnShelved |
| | Priority | | Low, emergency, high, normal, journal |
| | Suppressed | | Binary, 1 - 0 |
| | Shelved | | Binary, 1 - 0 |
| | From value | | Unit/state based |
| | To value | | Unit/state based |
| Master Alarm Database | Alarm threshold (master) | | 10, 0.5 |
| | Consequence of deviation | | Pump cavitation, Tank overflow, Pressure drop |
| | Consequence category | | Equipment, safety, environmental |
| | Corrective action | | Stop pump, Close inlet valve, Bypass inlet flow |
| | Allowable response time | | 0.5 min, 4 min |
| | Basis | | Pump cavitation at 2%, tank overflow at 107% |
| | Retention period | | 1 year, 5 years |
| | Report requirements | | Pump report, Safety report, Environmental report |
| | Notification requirements | | None, Environmental/Safety coordinator |
| Process Layout | Hierarchical classification | Production unit | Coker, Absorber, Distillation |
| | | Unit | Furnace, Pump, Column |
| | | Controller | Outlet pressure, inlet temperature |

Table 2.1: The different attributes and available information for an alarm message.

what happened in the related event with the specific tag: alarmed, returned to normal, was acknowledged by an operator, an operator action was performed or the alarm was suppressed or shelved. An important indicator of a well designed alarm system is the distribution of the different *alarm priorities* indicating the severity of the event to the operators. *Suppressing* an alarm refers to any mechanism to prevent the indication of the alarm to the operator when the base alarm condition is present [14]. For example, an alarm that is caused by another alarm that is already handled can be suppressed. *Shelving* an alarm temporally suppresses it for a specific duration of time, and this action is usually performed by the operator. The alarm is automatically unshelved when the duration runs out. The suppressed and shelved alarms are not raised for the operators, however, they are listed in the alarm and event summaries. The suppressing and shelving of alarm messages can be indicated by various data formats: by binary variables indicating the shelved or suppressed alarms, or by the timestamp of the start and end times or by a state indicator, which indicates whether the alarm is in an active, suppressed, shelved, or another state. In the case of alarm threshold modification, the 'from' and 'to' values can also be recorded.

The second group of attributes contains the original definition and implementation logic of the alarm message. This information is usually stored in the alarm philosophy documents, the alarm definitions of the master alarm database and the deployment descriptions. These attributes are often not displayed to the operators during the production as they are either aware of this information or it would only unjustifiably increase their information load. As such, the original *alarm threshold* of the master database is not displayed directly to the operator together with the alarm message, as only the actual threshold setting is relevant during the current situation. Moreover, in the case of a well-established and rationalized alarm message, the existence of the alarm message is well-proven, therefore, its *consequence* and its *consequence category*, *corrective action*, the *response time* for the corrective action and a brief explanation of the *basis* of alarm message is provided in the documentation of the alarm system. The *notification* and *report requirements* can also be defined.

The third category of alarm message attributes describes the place of alarmed variable within the process, which is important for the monitoring of the spillover effects of malfunctions [28]. Several hierarchical decomposition structures can be defined for chemical processes, following the guidelines of the ISA-95 standard [29].

The application of the following hierarchical levels is recommended (enumerated in top-down order): enterprise, site, area, production unit, unit, and the level of sensors and actuators (although the standard focuses on the first four levels). Therefore, if an alarm is raised, its origin can be defined hierarchically: the production units (such us columns, reactors, separators, *etc.*) are composed of units (furnaces, pumps, *etc.*). At the bottom level of the presented hierarchy, the sources of event signals are presented, *i.e.*, the sensors and actuators.

### 2.3.2 Load-based metrics

In industrial environments, the group of *load-based metrics* is most likely referred to as the measurement of alarm system performance, as these measures are mainly monitored and targeted by the operations. These metrics usually do not provide any insight to the problems of an alarm system, as their aim is to provide a general, bird's eye view of whether the alarm system is overloaded or functioning in a manageable or predictive intensity. Another characteristic feature of these measures is that they may not identify any specific problems regarding the location, cause or origin of the problems of an alarm system, either.

**Univariate alarm system load**

The *load-based metrics* are based on the event occurrences recorded in the Alarm & Event Log database of the system. According to the recommendations, in general, at least 30 days of alarm data is required for calculating most of the metrics presented in this subsection [14]. The calculated alarm loads are normalized to different time units (10 minutes, hours, days, *etc.*) providing an alarm occurrence frequency-base of measure. The following are the typical metrics in this category:

- Average alarms per time unit and per operator position
- Percentage/number of stale alarms
- Peak (maximum number of) alarms per time unit
- Distribution of activated alarm priority in the specific temporal period
- Percentage of time units containing more than a defined number of alarms, usually a special case is considered when the threshold on the number of

alarms is set to the same number above which an alarm flood is defined on the system

- Percent of time the system is in a flood condition

There are two tables that cannot be left out from any work discussing the performance metrics of industrial alarm systems. These popular tables compare the performance of an alarm system to the recommendations of the guidelines using the general, alarm log based performance metrics. The alarm system performance recommendations of the ANSI/ISA-18.2 standard, which are used as a reference in many works is presented in Table 2.2. As can be seen in the similarly popular comparison of the alarm management performance of different industrial sectors presented in Table 2.3, these metrics are far from the state of the industry (Table 2.3 is adapted from [23] and the presented data is extended with the metrics of [14]). The significant underachievement of performance targets is also reinforced by the work of Kim VanCamp [9].

An evolutionary trend is indicated by the red-highlighted rows in Table 2.2, as these performance indicators have been deleted from the recommendations between the 2009 [13] and 2016 [14] versions of the standard. The deletion of the authorization related metrics, namely the "Unauthorized Alarm Suppression" and "Unauthorized Alarm Attribute Changes" performance indicators, is simply due to the new approach that the monitoring of the unauthorized actions should be a separate process from alarm system performance assessment. This can be understood as some of the unauthorized suppression methods cannot be detected algorithmically (cutting wires or silencing a horn) [30] and the alarm authorization has been removed from the monitored KPIs of alarm management. However, for the sake of completeness, these metrics are discussed in Section 2.3.6.

The details of the road leading to the deletion of the "average alarm number per time unit" type of performance metrics for hours and days are illustrated in the bottom timeline of Figure 2.1. The original recommendation of 1 alarm/10 minutes by the Bransby & Jenkinson survey [31] aimed for the maintenance of a well-distributed alarm load on the operators. However, this metric was quickly turned to 6 alarms/hour, which was upscaled to 150 or 144 alarms/day around 2010. However, in 2016, the monitoring of the daily alarm numbers was deleted and the standards returned to the 10-minute and hour-based metrics. The reason was simple: the human operators cannot maintain an increased performance for a

| Metric | Target value | |
|---|---|---|
| Annunciated Alarms per Time | Target Value: Very Likely to be Accept. | Target Value: Maximum Manageable |
| Annunciated Alarms per Day per Operating Position | $\sim$ 150 alarms per day | $\sim$ 300 alarms per day |
| Annunciated Alarms per Hour per Operating Position | $\sim$ 6 (average) | $\sim$ 12 (average) |
| Annunciated Alarms Per 10 Minutes per Operating Position | $\sim$ 1 (average) | $\sim$ 2 (average) |
| Metric | Target Value | |
| Percentage of hours containing more than 30 alarms | $\sim< 1\%$ | |
| Percentage of 10-minute periods containing more than 10 alarms | $\sim< 1\%$ | |
| Maximum number of alarms in a 10-minute period | $\leq 10$ | |
| Percentage of time the alarm system is in a flood condition | $\sim< 1\%$ | |
| Percentage contribution of the top 10 most frequent alarms to the overall alarm flood | $\sim< 1\%$ to 5% maximum, with action plans to address deficiencies | |
| Quantity of chattering and fleeting alarms | Zero, action plans to correct any that occur. | |
| Stale alarms | Less than 5 percent on any day, with action plans to address | |
| Annunciated Priority Distribution | 3 priorities: $\sim$ 80% Low, $\sim$ 15% Medium, $\sim$ 5% High or<br>4 priorities: $\sim$ 80% Low, $\sim$ 15% Medium, $\sim$ 5% High, $\sim< 1\%$ ‘highest’<br>Other special-purpose priorities excluded from the calculation | |
| Unauthorized Alarm Suppression | Zero alarms suppressed outside of controlled or approved methodologies | |
| Unauthorized Alarm Attribute Changes | Zero alarm attribute changes outside of approved methodologies or MOC | |

Table 2.2: The recommended alarm performance metrics in [14]). The red rows have been removed from the recommendations between 2009 [13] and 2016 [14] versions of the standard.

| | EEMUA 191 | ANSI/ISA 18.2 | Oil & Gas | Petrochemical | Power | Other |
|---|---|---|---|---|---|---|
| Average alarms per day | 144 | 150 | 1200 | 1500 | 2000 | 900 |
| Average standing alarms | 9 | 5 per day | 50 | 100 | 65 | 35 |
| Peak alarms per 10 minutes | 10 | 10 | 220 | 180 | 350 | 180 |
| Average alarms/ 10-minute interval | 1 | 1 | 6 | 9 | 8 | 5 |
| Distribution % (Low/Med/High) | 80/15/5 | 80/15/5 | 25/40/35 | 25/40/35 | 25/40/35 | 25/40/35 |

TABLE 2.3: Cross-industry comparison of alarm activation numbers (adapted from [23] and extended with the metrics of [14]).

longer time period. For comparison, consider the following analogy: The current world record holder of 100 m run (as of 2021), Usain Bolt, could run 100 meters in 9.58 sec in 2012, but it is clear that no sprinter can maintain this speed for longer time periods, as this would mean 100 secs for 1000 meters or a little bit more than an hour for a marathon (67.37 minutes). By comparison, the current official world record for marathon running is 2:01:39, held by Eliud Kipchoge (unofficially, he could already break the two-hour barrier). Similarly, a well-trained and good-performing operator is able to express a superhuman-like performance and attention for a short period of time in the case of a serious malfunction, but cannot maintain this performance for a longer time period. This is a significant problem as the worker fatigue costs $77-$155 billion per year to U.S. companies due to increased health costs, production loss and damage due to accidents [32].

Based on the above logic, one can ask what a reasonable time unit should be. The answer is an expected one in the process industries: it depends on the process. For a new alarm system, the time required for an alarm to be handled needs to be estimated (*e.g.*, based on the expected rate of change of the process variable). As in the case of a new alarm system, every alarm should have a specific response from the operator, which necessitates the estimation of the average response time for all configured alarms in the system. This way, the operators will have enough time to acknowledge every alarm message properly and the optimal value of this metric should be 1 alarm/average response time/operator. If this target value of 1 is significantly exceeded, then the process is clearly not in a well-operated and safe state. In the case of an existing alarm system, the average time when an alarm is active can be estimated. This would be the time limit for an alarm message to be acknowledged or responded by a specific operator action and it should be used for the calculation of an informative performance metric. However, the calculation of an average time for operator response holds some practical problems: there should be no alarm message disappearing without being acknowledged or being acknowledged after the variable has returned to normal. Similarly, there should be no alarm acknowledged but not returned to normal over the time period chosen for the calculations. Therefore, there should be no stale, chattering or fleeting alarms during the calculations, which leads to a significant practical calculation problem, requiring a systematic and careful cleaning of the data. Another process specific question that we need to consider during the determination of the informative unit is the tasks of the operator outside of taking operator actions on the human machine interface. If the operators working in front of their operator console need to rely on the work of other operators or communicate with the field operators to perform some specific tasks, the tasks handled by the operator can increase significantly (or the other operators can provide support as well) and this should be also incorporated in the calculations. These logics are well explained and documented in [30].

### Multivariate performance metrics

Based on the above discussion, one aspect is clear, regardless of the choice of time unit, the average alarm rate does not provide an overall summary of the performance of the alarm system and certainly not support the problem diagnosis: if low, then the alarm system is probably good; however, if it is high, we need to know more about the system. A high alarm rate can simply indicate the presence of a process upset and in this case the analysed alarm system just performs its task, yet this can also indicate a highly under-performing system. This leads us to the discussion of multivariate alarm load-based performance measure of alarm systems, that are gaining more and more emphasis in recent years.

EEMUA 191 addresses the monitoring of the average alarm rate per 10 minutes in steady-state operation and the maximum number of alarms in a 10-minute period per day in upset conditions, averaging these metrics over a month of operation. Here again, several approaches are present with several categories and thresholds defined, but the general picture is presented in Figure 2.3. The two performance metrics used for monitoring are usually the average and the maximum alarm rates per time unit per operator and based on their respective values, categories as overloaded, reactive, stable, robust and predictive (in the order of increasing performance) are defined. The region where the average alarm rate is higher than the maximum alarm rate is of course impossible to reach. Similar charts are defined in the second and third editions of the guidelines of the EEMUA [3, 1], but different versions of similar performance evaluation charts are present in different processes. Hu *et al.* introduced how these multivariate plots can be extended to three variables by pointing out the number of unique alarms in the related temporal period on a bubble chart [27]. As discussed in [33], the improvement of an alarm system over time is well-traceable in these charts.

While a highly intuitive metric of the alarm load-based performance measurement would be the economic analysis of alarm systems, this is a highly sensitive and problematic question and a separate section is devoted to it in Section 2.5.

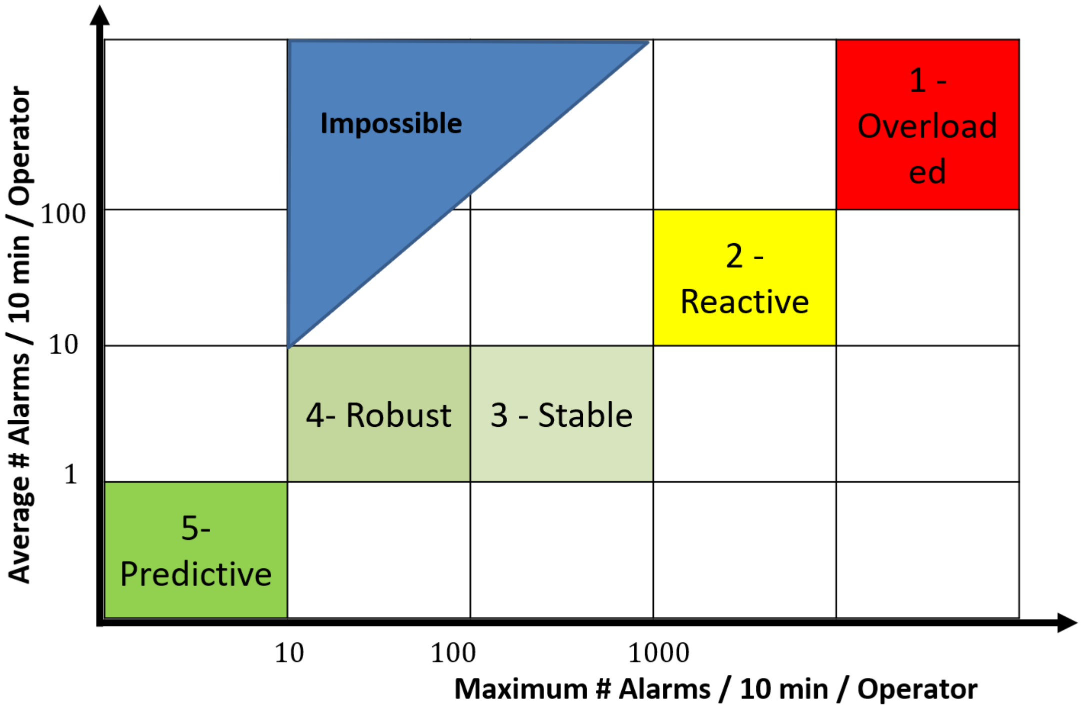


Figure 2.3: The multivariate monitoring of alarm system performance

### 2.3.3 Diagnostic metrics

The *diagnostic metrics* of an alarm system are aimed at the identification of specific problems on specific alarms. These metrics are usually monitored by the owner of the alarm system or the personnel responsible to take actions for its maintenance or efficiency improvement. The various combination of the measures from the *performance* and *diagnostic metrics* form the basis of the most often monitored metrics in industrial plants and usually the construction of alarm management reports, dashboards and KPIs are based on them.

Some commonly used metrics are the following:

- Listing and quantity of the most frequent alarms (Top N bad actors)
- Listing and quantity of chattering alarms
- Listing of stale alarms
- Listing of shelved alarms, possibly with shelving duration
- Listing of out-of-service alarms, possibly with durations

- Listing of potentially redundant alarms shown by analysis

### Top N bad actors

A majority of alarms are usually generated by a small number of process variables known as bad actors. According to the literature, a bad actor is "*an alarm that is suspect and cannot be relied upon to deliver accurate information to the operator, such as stale, chattering, duplicate or suppressed alarms*" [34]. In practice, by bad actors, the most frequent alarms in the Alarm & Event log database are referred, which are responsible for the bulk of the alarm load. According to the ISA-18.2 standard [13], "*Relatively few individual alarms (e.g., 10 to 20 alarms) often produce a large percentage of the total alarm system load (e.g., 20 % to 80 %). The most frequent alarms should be reviewed at regular intervals (e.g., daily, weekly, or monthly). Substantial performance improvement can be made by addressing the most frequent alarms*". This Pareto-like, 80-20 rule of distribution that 20 % or less of alarm variables are responsible for the 80 % or more alarms is common, extreme cases where only one variable contributes to more than 50 % of the total alarm load are known [35]. In other cases, the top 5 bad actors contributed to 87 % [36], or the top 10 bad actors contributed to more than 75 % of the total number of annunciated alarm messages [37]. Numbers like this are not considered outliers based on our experience as well. Therefore, the monitoring of the top 10 most frequent alarms is strongly recommended by alarm standards, and they should not contribute to more than 5 % of the total alarm load (no bad actors are acceptable) [35]. Usually, top (5-)10 alarms are monitored regularly on bar plots indicating their number and a line plot is applied to show their cumulative proportional contribution to the overall alarm number [37], but their temporal comparison is common as well [27]. Thanks to their practical importance and good problem-solving efficiency, several frameworks incorporate their usage [35, 38, 21].

### Chatter index

Less informative alarms, namely nuisance or constant ones, significantly increase the operator workload by nonactionable distractions. The most common form of nuisance alarms are the chattering ones, which do not sound for a sufficient time to allow the operators to perform corrective actions and in critical plant conditions can significantly hinder the work of the operators. In a properly rationalized and designed alarm system, after the elimination of nuisance (chattering) alarms, the resultant alarm rate reflects the ability of the control system to keep the operation in the normal operating zone without operator interactions [14]. The chattering alarms are essentially in conflict with the philosophy that each alarm should be actionable. Hollifield *et al.* claim that chattering alarms are the most common type of alarm, constituting about 70% of all alarms [2]. Similarly, constantly sounding alarms are also harmful to the quality of the alarm data. In an industrial environment, these long-standing alarms are ignored by the operators either as a result of their uninformativeness, or the fact that they are hidden from the operators as shelved or forbidden alarms (these alarms are usually still present in historical datasets). Different approaches are present for the detection of the presence of a chattering alarm, for example the balance between the actions taken by the operator and occurrences of alarms [39], but the most well-known and commonly used approach is the application of the chatter index introduced by Kondaveeti *et al.* [40], [41]. Various studies applying the chatter index in alarm management are summarized in the following list:

- Kondaveeti *et al.* introduced the chatter index for the quantification of alarm chatter [40]
- Wang and Chen developed an online method for the detection and reduction of chattering alarms due to oscillations [42]
- The improvement of the alarm system of an industrial power plant case study [36]
- Plotting the Chatter Index over one week's period for top 50 alarms [43]
- Sun *et al.* reduces the number of chattering alarms via median filters [44]
- The application of the chatter index for the alarm system improvement in a Combined-Cycle Gas Turbine Power Plant [45]
- The design of alarm deadbands for the reduction of false and missed alarms and alarm chattering [46]
- In our previous work, we applied the chatter index to prefilter the alarm and event log database before sequence mining [28]
- Naghoosi *et al.* developed a method to estimate the chatter index based on statistical properties of the process variable as well as alarm parameters [47]

The calculation of the chatter index is discussed in Section 9.3.

### 2.3.4 Deployment metrics

The *deployment metrics* describe the progress of the implementation or modification tasks on an alarm system. Therefore, these metrics are only interesting during deployment and monitored and targeted by the alarm system owner and personnel responsible to measure the progress. A few examples for the most often used deployment metrics of alarm systems are:

- Percent of alarms rationalized
- Percent of alarms monitored
- Priority distribution of rationalized alarms.

### 2.3.5 Scaling metrics

The *scaling metrics* are usually not used for the direct measurement of the alarm system performance but to obtain a scaled and comparative measure by scaling other metrics to a unit base. These metrics are not directly used or reported, but are used in reporting other metrics. The unit base can be obtained by scaling to various units depending on the aim of comparison: time unit, control loops, number of operators, plant-based counts, *etc.*, can be derived.

### 2.3.6 Audit metrics

From a certain viewpoint, the *audit metrics* are not in line with alarm system performance metrics, as the scope of data audits, despite being present for decades, has been removed from the tasks of alarm management by the latest ANSI/ISA-18.2 standard in 2016 [14]. In order to respect the traditions and for the sake of completeness, I briefly recall the aim of these measures. The audit metrics simply describe the amount of data that has been (or missed to be) audited. Two simple examples are the "number and nature of unauthorized changes" and the "number and nature of unauthorized alarm suppression". However, as the process of authorization can be considered a technical or rather system administration question from the view point of alarm management, it was removed from the list of measures strictly focusing on the performance of alarm management tasks.

### 2.3.7 Design metrics

The *design metrics* are the measures aiming for the determination of the alarm thresholds optimal from a certain point of view. Although they serve a practical purpose, these measures are primarily utilized by the academia as an indicator of how well the designed alarm thresholds are appropriate for specific tasks providing a clear choice of the cost function of optimization tasks. They can be formulated for the characterization of a certain alarm level or the whole analysed alarm system as well.

The most often used design metrics of alarm systems are:

- False Alarm Rate (FAR)
- Missed Alarm Rate (MAR)
- Average Alarm Delay (AAD)
- Receiver operating characteristic (ROC) curve
- Fault detection rate (FDR)

In order to get a deeper insight into the task of alarm threshold design, the fundamental practical problems of the design of a perfect (univariate) alarm message is to be considered. First, as presented in Figure 2.4, industrial processes and the measurement (measurement noise) of process variables are subject to uncertainty. The process variables are scattered over the mean value of the specific state and these distributions usually overlap with each other. As a result, the determination of a perfect alarm threshold for the separation of normal and faulty operation may not be feasible in most cases. Even if we statistically improve the accuracy of the measurement, the faultless and the faulty operation may not be distinguishable by a single measurement. The more permissive the chosen threshold is the more missed alarms we get during the operation. On the contrary, the stricter thresholds generate an increased false alarm rate. This is indicated on the right side of Figure 2.4.

Considering the performance of not a single alarm message but the system of multiple alarm messages, the problem of connected/correlated alarm variables arises as depicted in Figure 2.5. As the variables monitored by the alarm messages

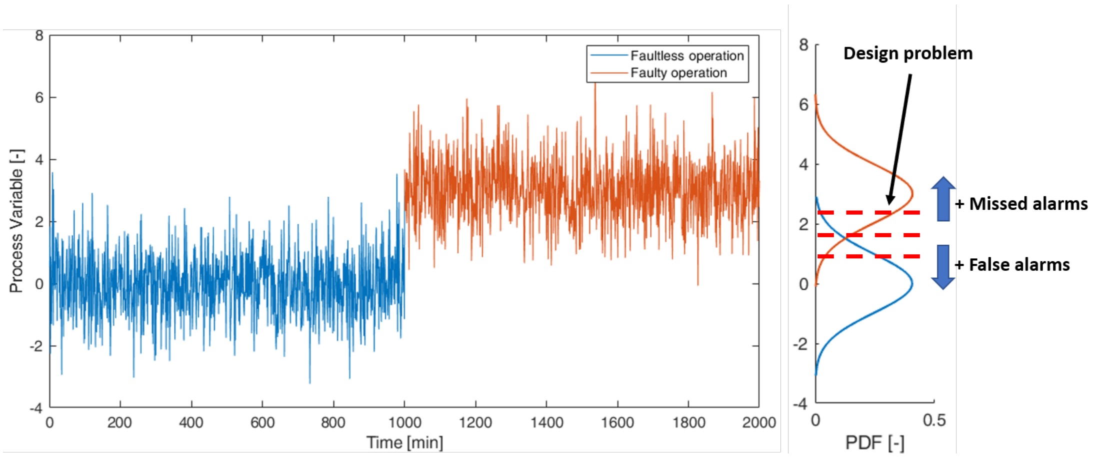


FIGURE 2.4: The problem of measurement and process uncertainty is to be considered during alarm design: the faultless and faulty operations may not be distinguishable by a single measurement.

are usually connected by the underlying physical properties of the system, the shape of the optimal operational area most likely cannot be delimited by univariate thresholds. Therefore, not just the uncertainty of process variables, but the inaccurate delimitation of the optimal operating range also poses a problem. The accuracy of the detection of process abnormalities is further complicated by the speed of determination.

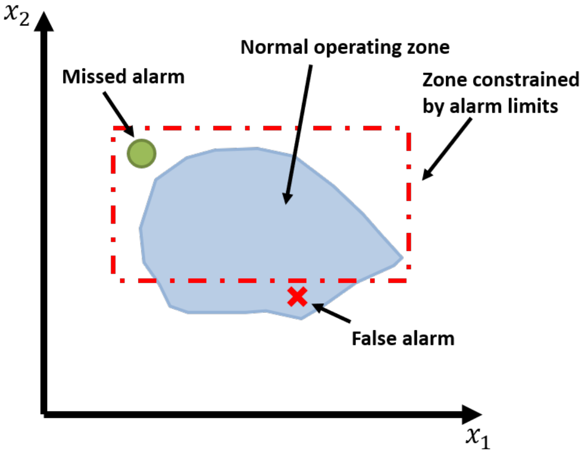


FIGURE 2.5: The problem of connected variables and the application of univariate alarm design.

Motivated by the above mentioned problems, a trade-off between the accuracy and speed of the determination of process malfunctions is drawn. The primary metrics used here are the false alarm rate (FAR) and missed alarm rate (MAR) and the average alarm delay (AAD) [48]. For the formal description of these metrics, consider the measurement of a process variable $x$. A false alarm is an alarm that is raised although $x$ behaves normally and a missed alarm is the case when no alarm is raised during the abnormal behaviour of $x$. The false alarms unreasonably overloads the operators and undermines their trust in the system leading to the cry-wolf effect [49], while the missed alarms are naturally against the functionality of indicating the presence of abnormalities in an alarm system. According to Figure 2.6 the probability density function (PDF) of variable $x$ during the normal operation is $q(x)$. The FAR, the probability of $x$ being above its trip point $x_{tp}$ during normal operation, is expressed as:

$$FAR = \int_{x_{tp}}^{+\infty} q(x)dx \tag{2.1}$$

The probability density function (PDF) of variable $x$ during the abnormal operation is $p(x)$. Then the probability of a missed alarm is given as:

$$MAR = \int_{-\infty}^{x_{tp}} p(x)dx \tag{2.2}$$

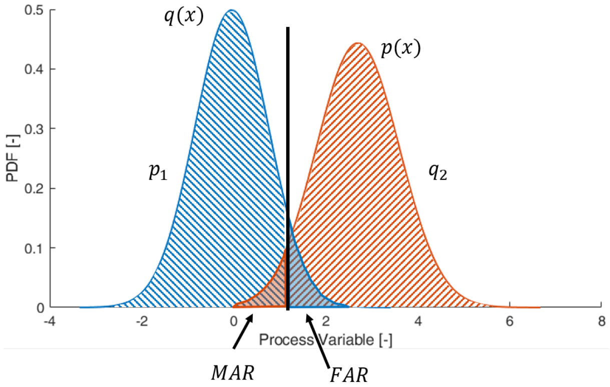


FIGURE 2.6: The graphical interpretation of the false alarm rate (FAR) and missed alarm rate (MAR) metrics.

The alarm delay of an alarm is calculated as the time difference of the occurrence of the abnormal condition and the appearance of the alarm message. The average alarm delay (AAD) is the expected value of the alarm delays.

As in most practice, the exact time of occurrence of a malfunction is unknown, and these metrics are usually not appropriate for the data-based measurement of alarm system performance, but are highly favoured by the academia for the development of new solutions for alarm threshold optimization tasks.

### 2.3.8 Operator's perspective

In previous categories, I approached the task of the assessment of alarm system performance from various, mainly technical, points of view. This work, however, would be far from complete if I left out the core actors and the driving forces of the improvement of alarm management systems, *i.e.*, the operators themselves. Human operators cannot be left out of this process as their performance is fundamentally determined by the performance of the alarm system and vice versa [18]. The overload of operators is disadvantageous for the success of alarm management as multi-tasking negatively impacts the performance and execution time of a task [50], in turn, increasing the possibility of errors [51]. There are numerous approaches to incorporate the operator performance to the overall performance assessment of alarm systems. The underlying philosophy thought has been well-summarized in the ANSI/ISA-18.2-2016 standard [13]. The operators' tasks when a disturbance or malfunction occurs can be divided into three stages: detect, diagnose and respond. This is discussed in the feedback model of the operator. During the detection, the operator becomes aware of the process deviation from the desired condition. Then, the operator diagnoses the problem, therefore, using their personal experience they determine the chosen corrective action. Finally, they respond to the situation by performing the corrective action. However, the operators' ability to carry out these tasks and hence, the overall performance of the alarm system is affected by a variety of circumstances: the workload of the operator, training, fatigue, the efficiency of Human-Machine Interface (HMI), short term or working memory limitations, motivation, *etc.* This interaction of operator-alarm system performance is illustrated in several studies, for example, the feedback model of operator process interaction in [14]. Figure 2.7 illustrates another representation given by Nochur *et al.* [18]. The three blocks represent the entire interaction cycle between the alarm system, the operator and the process itself. The process conditions generate the alarms in the alarm system, while the alarm system calls attention to the process conditions by sounding these alarms to the operators. The operators, after the acknowledgment of the situation, may interact with the process. The new process conditions effect the alarm system and the cycle continues on. The $\tau$ values between the blocks represent the dead time associated with the interactions, while the $P$ values represent the probability of interaction.

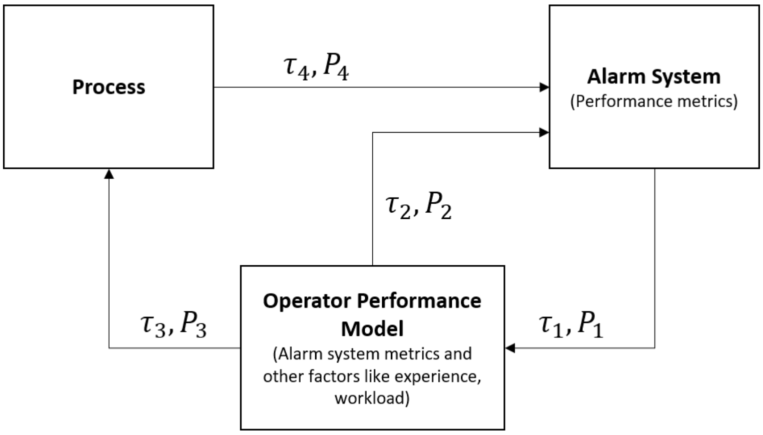

Process
$\tau_4, P_4$
Alarm System
(Performance metrics)
$\tau_2, P_2$
$\tau_3, P_3$
Operator Performance Model
(Alarm system metrics and other factors like experience, workload)
$\tau_1, P_1$

FIGURE 2.7: The causal effects of process, alarm system and operator performance according to Nochur *et al.* [18] (re-edited).

The process operators have been incorporated in the alarm management tasks for a long time in the form of operator surveys and usefulness questionnaires [7]. Such operator surveys and follow-up interviews provide a qualitative feedback from the users of the alarm system. The typical surveys usually include the following topics [52]:

- Operator experience
- Amount of DCS training
- Support provided by the alarm system during normal operation
- System performance during plant faults and trips
- Alarm system design
- Alarm management processes and procedures
- General questions requesting recommendations to improve the alarm management system

These surveys of the operators provide a good basis, however, they are not the explicit measures we are primarily interested in as they are not numeric and do not provide an online or on-demand picture of the state of the alarm system. Although several highly advanced alarm management solutions have been published mainly in the past decade, only a few incorporate the operator actions in the analysis. The work of Hu *et al.* [53] was the first, recognizing that it is also necessary to study

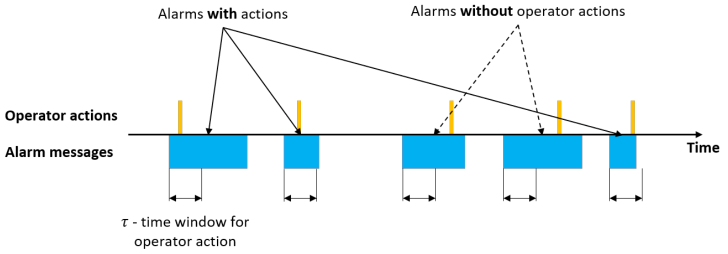

Alarms **with** actions
Alarms **without** operator actions
**Operator actions**
**Alarm messages**
**Time**
$\tau$ - time window for operator action

FIGURE 2.8: The timeline of alarm messages and their operator actions in response. The blue and yellow bars represent the alarms and operator actions, respectively. If the occurrence of an alarm message is not followed by an operator action in a system-specific time window, then that alarm is considered as "unactioned".

the connection of process alarms and the corresponding operator actions for the proper specification of operational procedures. This work was further elaborated to discover interesting workflow modes in the noisy Alarm & Event Log databases [54] and graphically represent them [55]. Building upon these findings, we previously highlighted that the sequences of industrial alarms cannot be handled by themselves since the interactions of the operator continuously change the underlying processes and thus the evolution path of the alarms [56] and formalised how the events of alarms and operator actions need to be represented for the extraction of useful knowledge from these databases [57]. Moreover, we have developed a frequent sequence mining and deep learning-based approach for the connection of alarm messages and the corresponding operator actions [58].

Besides performing operator actions as the answer of process alarms, a significant problem of the operators is the modification of alarm system settings in case they are not satisfied with the system. Therefore, by tracking the number of parameters different from the master alarm database of the alarm system, we can monitor how informative the alarms messages are to the operators or how well the operators accept the settings defined by the process engineers. The number of parameter deviations compared to the master alarm database of the alarm system are frequently monitored in daily *enforcement reports*.

A highly intuitive metric of the "actionability" of alarm messages is the calculation of the fraction of alarm messages which is followed by an operator action. For this, a $\tau$ time window is monitored after the occurrence of an alarm message, which can be considered the reaction time of the operators or the "deadtime" of the operator response. In case no action is performed by the operators in this $\tau$ time window, then the alarm message is unactioned. The schematic visualization of the analysis of the operator actions for alarm messages is illustrated in Figure 2.8. It is important to highlight that the fraction of actioned and unactioned alarm messages is not appropriate for the benchmarking of different processes as the operators can perform actions outside of the process control system as well, *e.g.* turn valves manually or communicate with field personnel outside of the control room. Therefore, this metric is rather applicable to provide a general picture and monitor the effects of alarm management projects on the same system. In the results section of the present chapter I will calculate the fraction of answered alarm messages.

## 2.4 How to point out the improvement of an alarm system? - A case study

The problem of the exclusive use of alarm load-based metrics instead of focusing on the quality of the alarm data is presented through the analysis of the alarm system of the hydrofluoric acid (HF) alkylation plant of the MOL Group Danube Refinery before and after the alarm rationalization process. The process flow diagram of the plant is depicted in Figure 2.9. The plant provides an ideal basis for the analysis of alarm management-related tasks, as its safety is of crucial importance due to the used strong acid. On the other hand, the technology is not so complex and therefore, even though the alarm system of the plant is important, its rationalization did not require as much human workforce as in the case of a bigger plant with more complex units and implemented sensors.

Prior to the present analysis, the operators and process engineers of the process reported a significant improvement in the performance of the alarm system, mainly due to the more informative and meaningful alarm messages, the results of the systematic alarm rationalization work. However, as we will see, due to the complex nature of such industrial systems, the numerical proof of this seemingly intuitive improvement is not trivial, as it is not reflected in a single, but in multiple metrics. Moreover, the improvement is not reflected in the metrics monitored by most industrial facilities, but deeper connections are to be captured. The basis of our analysis is two months (60 days) of operation data of the alarm & event-log database before and after the rationalization process, from the beginning of April until the end of May and from the beginning of November until the end of December 2020, respectively. In case of all the following figures, the ordinal number of alarm messages and the name of the alarm tags are removed due to confidentiality, but the problems and improvements, as well as how these metrics support the monitoring of alarm systems are well reflected in the trends and ratios depicted in the figures.

### 2.4.1 Load-based metrics

First of all, most industrial plants monitor the number of announced alarm messages or the number of active alarms in a specific time period. Figure 2.10 indicates the announced and the active alarms as the function of the ten-minute

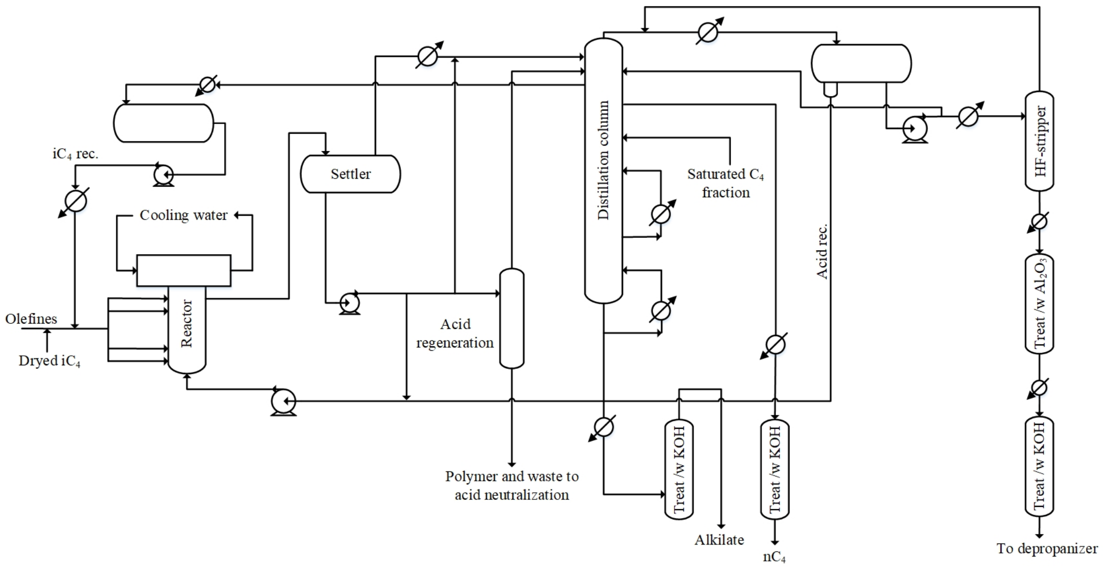


Figure 2.9: The process flow diagram of the industrial hydrofluoric acid (HF) alkylation plant.

time brackets. The solid blue and the dashed orange lines illustrate the number of alarms announced, while the dotted-dashed green and dotted red lines indicate the number of active alarms, before and after the rationalization process, respectively. While in the case of the number of announced alarms, just the number of activated alarm messages is considered, in the case of the active alarms, not just the alarms activated in the relevant 10-minute time bracket, but the ones activated formerly in the process and still active are considered during the calculation of the overall alarm load of the operator. However, the ordinal number of alarm messages is removed due to confidentiality, we cannot see a clear decrease in alarm load as the effect of the rationalization process.

While this trend may seem counter intuitive, as the alarm systems indicate the presence of process upsets, in the case of a more problematic operational period, even an increased alarm load is as acceptable. Since, according to the operators of the process, no significant plant upset has taken place in the analysed time periods, this is a good example of why the alarm load itself is not informative enough for the measurement of the performance of an alarm system.

As the alarm load alone is not sufficient for the examination of the performance of the alarm system, the alarm management standards recommend the application of multivariate metrics monitoring the maximum and the average alarm rate per operator in 10-minute periods of a day and averaging these daily values over a

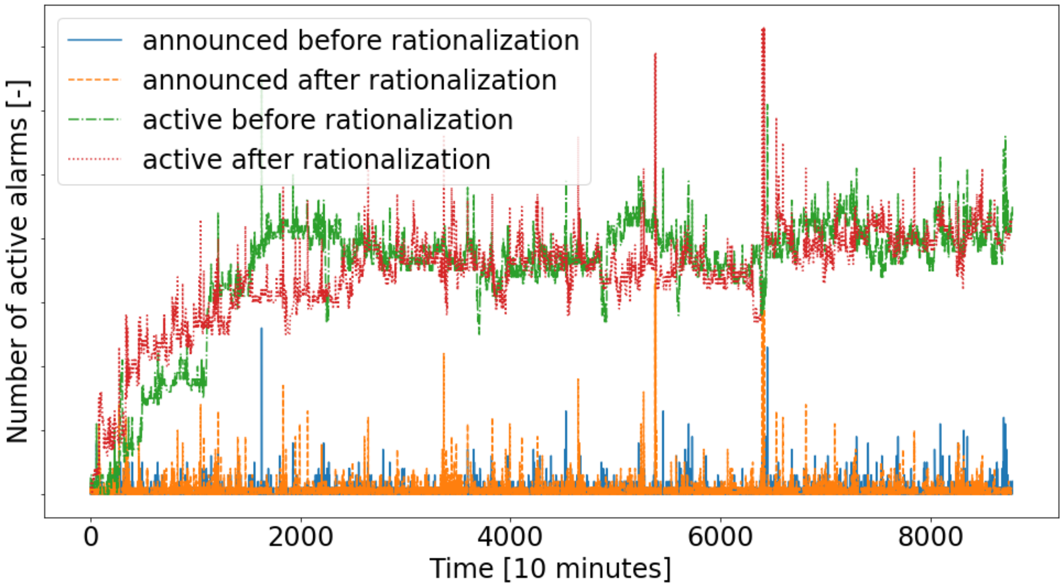
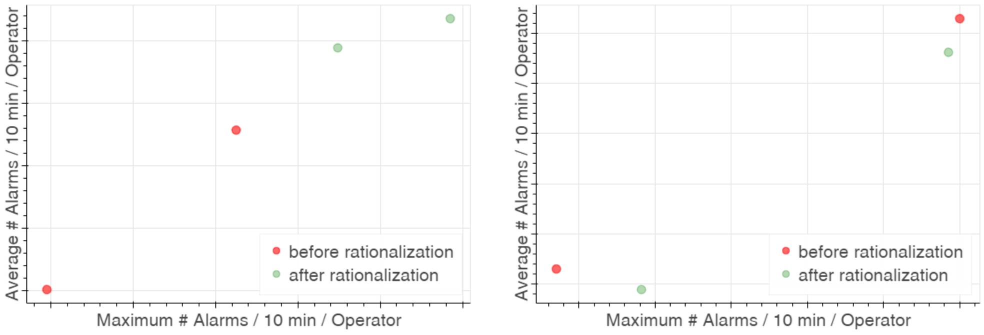


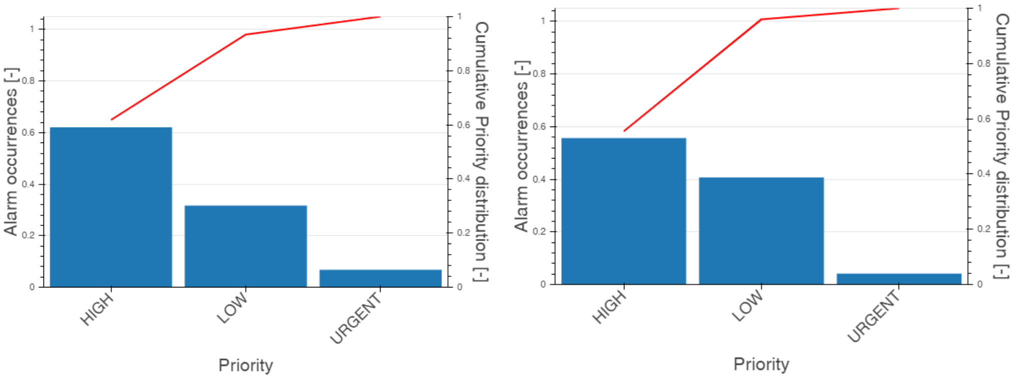
FIGURE 2.10: The number of announced and active alarms in 10 minutes before and after the rationalization of the alarm system. Despite preconceptions, the alarm load did not decrease after the alarm rationalization, but slightly increased.

month. The red and green dots in Figure 2.11 indicate the average alarm messages received by an operator in ten minutes as the function of the maximum number of alarms received by an operator in ten minutes before and after the rationalization process, respectively. However, these metrics are mainly applied for the analysis of the number of announced alarm messages as presented in part (a) of Figure 2.11, part (b) of Figure 2.11 presents this analysis for the number of active alarm messages. In this method, the average alarm rates indicate the constant load on the operators, while maximum rate represents a peak (most likely upset) operation, where the operators are under an increased pressure. As can be seen, in the case of the announced alarm messages the system seems to be worse as the result of the alarm rationalization process, as both the maximum and average number of announced alarm messages has increased. On the other hand, in the light of the number of active alarm messages, the rationalization process decreased the average alarm load on the operators. From this, we get the impression that the alarm rationalization process was not effective for the reduction of the number of alarm activations, but could improve the number of active alarms in a specific time period.



(a) Announced (b) Active

FIGURE 2.11: The multi-variate monitoring of alarm system performance in the HF alkylation unit for announced and active alarm numbers

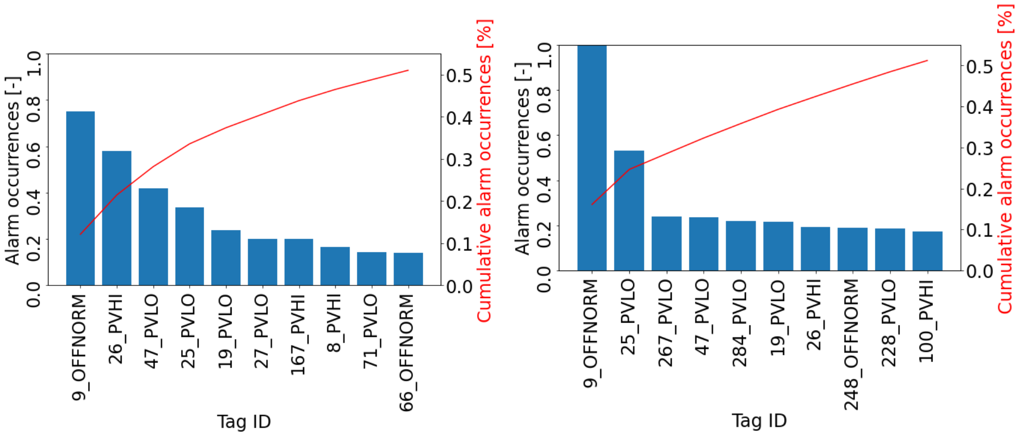


(a) Before alarm rationalization (b) After alarm rationalization

FIGURE 2.12: The distribution of the priority of announced alarms before (a) and after (b) the alarm rationalization process.

Figure 2.12 presents the distribution of the priority of the announced alarm messages before and after the alarm rationalization process. This indicator is significantly improved thanks to the alarm rationalization work, as the distribution of the alarm priorities, which was formerly dominated by the high signals in the vast majority, is shifted towards a more balanced direction. This more balanced distribution supports the operators with a clear indicator of which alarm is more urgent to handle, taking priority over less important ones.

As except for the distribution of alarm priority, we cannot state conclusively that the rationalization process caused a clear increase in quality, in the following, we shift our analysis towards the comparison of the type and consistency of alarm messages. By this we move from the general, load-based performance metrics of alarm systems (introduced in Section 2.3.2) to the diagnostic and operator-centric metrics, which indicate the presence of certain problems in the system and take



(a) Before alarm rationalization

(b) After alarm rationalization

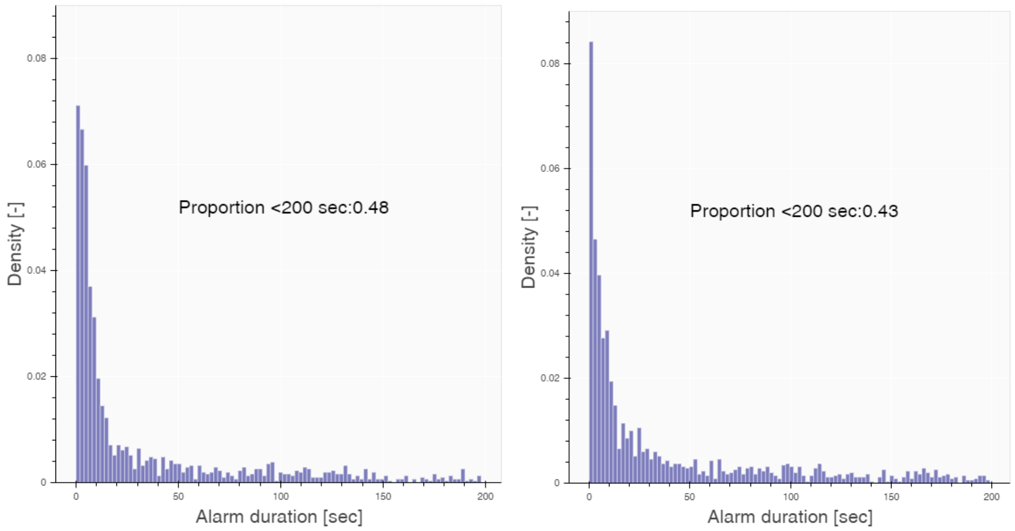
FIGURE 2.13: The top 10 bad actors before (a) and after (b) the alarm rationalization process. The (normalized) number of the individual alarms is indicated by the bars, while the cumulative contribution of the most frequent alarm messages to the overall alarm load is indicated by the red line.

into consideration the operators as agents interacting with the control system as presented in Sections 2.3.3 and 2.3.8.

### 2.4.2 Specific investigations

Figure 2.13 shows the trend of the Top 10 bad actors, a widely applied industrial monitoring tool of the alarm messages contributing to the overall alarm load. The ordinal numbers of alarm messages are not published due to confidentiality, but all of the numbers in both parts of the figure are normalized by the highest value, the "9_OFFNORM" alarm after the rationalization process. As can be seen, the values, the distribution, their contribution to the overall alarm load and the list of bad actors have not changed significantly due to the alarm rationalization process. However, the ratio of the Top 10 bad actors to the overall alarm number changed: it dropped from 51% to 41%. The applicability of the method is well reflected as well, the significant contributors can be analysed manually by the responsible personnel. For example, the most significant contributor to the alarm numbers, the "9_OFFNORM" tag indicates the problem of a sewage pump removing the used and contaminated liquid from the process. Another constant contributor, the "26_PVHI" is a high alarm on the HF stripper.

A highly intuitive metric of the informativeness of alarm messages is the analysis of their duration. The comparison of the distribution of alarm durations in seconds



(a) Before alarm rationalization

(b) After alarm rationalization

FIGURE 2.14: The distribution of alarm length

before and after the rationalization process is presented in Figure 2.14. The very short alarm messages are likely to be false indicators that require no action from the operators and hence, simply disturb their attention. On the other end, the very long alarms, stale or long-standing alarms can indicate the occurrence of a long-lasting problem, or in numerous cases, a bad alarm setting as well. Therefore, the optimal value is somewhere between these two extrema, with reasonable and actionable alarm durations. In the present setting, the duration of the alarm messages follows the same, highly skewed trend before and after the rationalization process as well. As can be seen in Figure 2.14, before the rationalization process almost half (48 %) of the alarm messages lasts less than 200 seconds. The fraction of these very short messages is reduced to 43 % as the result of the rationalization process.

Besides the duration of the alarm messages, the frequency of their occurrence is interesting as well. The chatter index, described in Section 2.3.3, describes the time spent between two occurrences of the same alarm tag. Figure 2.15 presents the chatter index as the function of the alarm tags. According to the industrial guidelines [13], an alarm with a chatter index above $\frac{3}{60} = 0.05$ alarms/s is considered to be chattering. Based on this, the chattering of the alarm messages has not decreased, but increased as the effect of the alarm rationalization.

A highly useful metric of the alarm system is to see how informative the alarm settings to the operators of the process or how satisfied they are with the settings

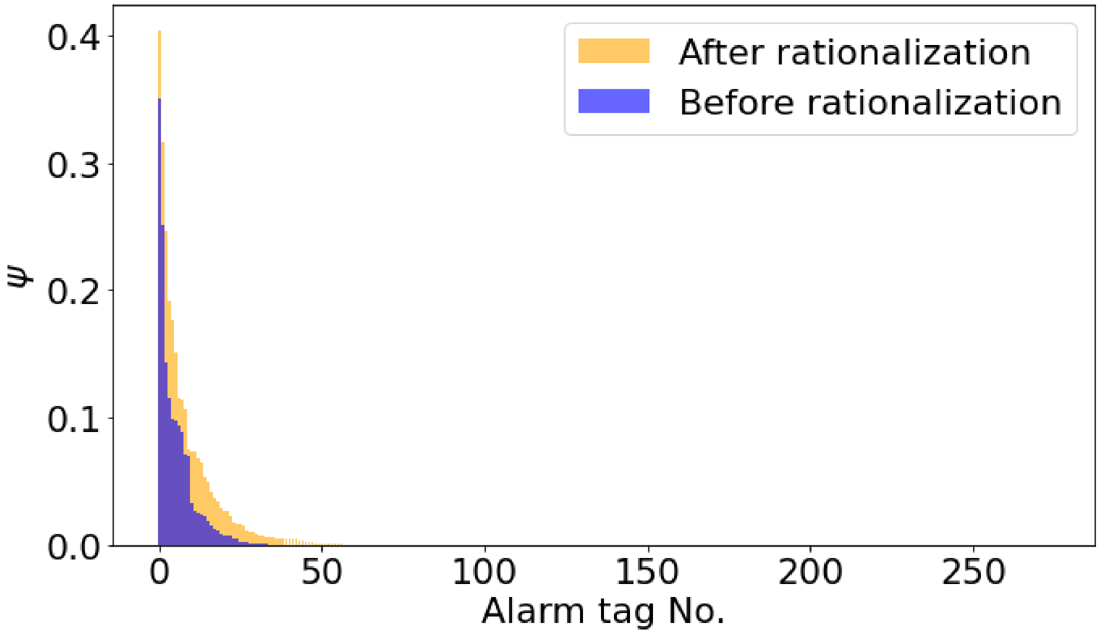


FIGURE 2.15: The chatter index of each alarm tag before and after the alarm rationalization process.

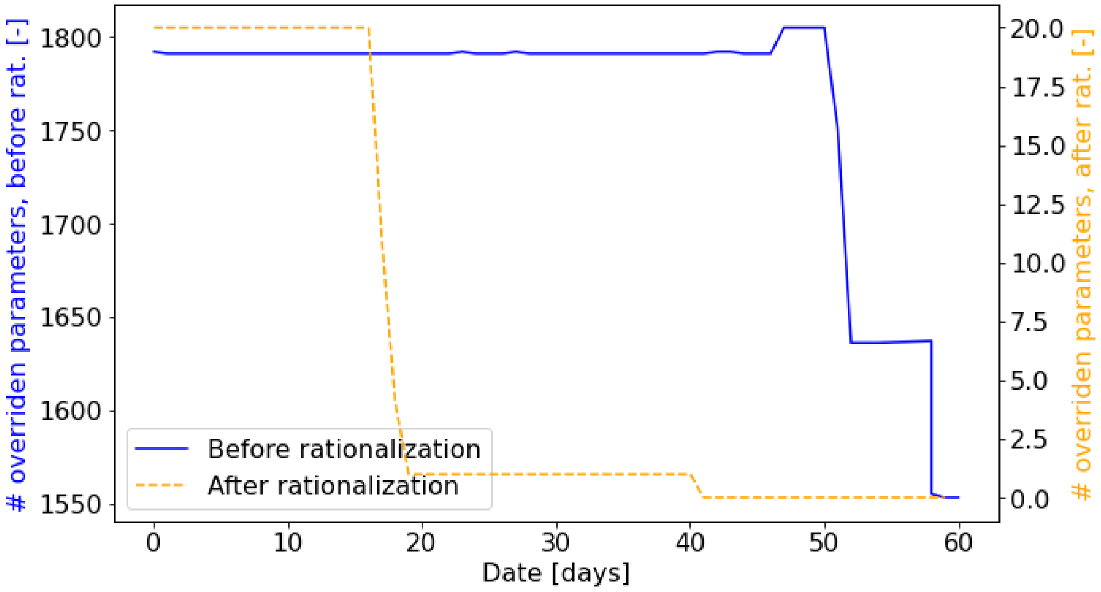


FIGURE 2.16: The number of daily deviating alarm parameters compared to the master alarm database, before and after alarm rationalization.

provided in the master alarm database. The system of the HFA plant compares the alarm parameters of the DCS to the ones provided in the master alarm database every night. Figure 2.16 illustrates the number of parameters deviating from the parameters of the master alarm database. As indicated by the solid blue line, the number of deviating parameters before the rationalization process was in the high 1800s daily (solid blue line, left-hand side vertical axis), while this number dropped to the order of tens (dashed yellow line, right-hand side vertical axis). This is a substantial improvement in the everyday operation of the plant, providing more informative metrics to the operators. The list of deviating parameters is also provided to the process engineers and if a parameter reappears in the list, they systematically investigate the reason and correctness of the setting used by the operators. This way, the process relevant knowledge of the operators is exploited during the improvement of the system.

Finally, I have investigated the fraction of answered alarm messages before and after the rationalization process. Based on the recommendation of process engineers, after the occurrence of an alarm message, a 1-minute time window was investigated and the fraction of alarm messages in which this time window an operator action was performed calculated. Even though the number of alarms in two months was more than 25% higher after the rationalization process, therefore, the operators worked with more alarms, the fraction of answered alarm messages is increased from 22% to 25%. This very slight improvement is however, not significant, supports the fact that not the number of alarm messages, but their informativeness is important for the operators.

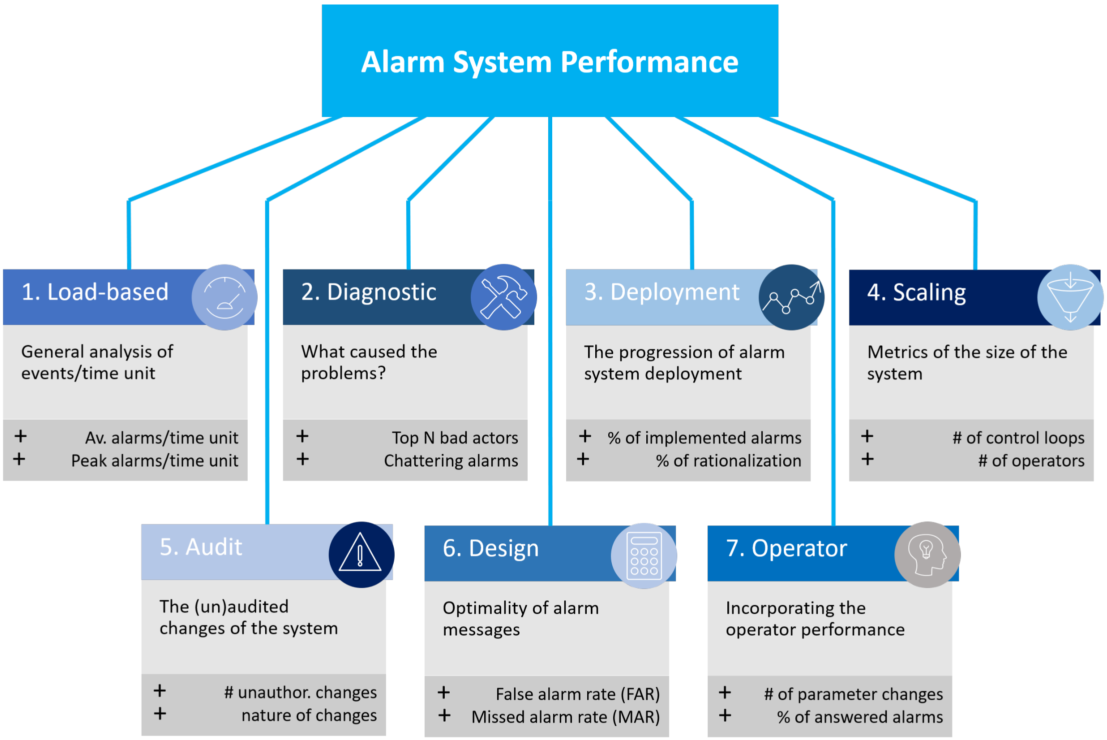


FIGURE 2.17: The seven categories of alarm performance metrics.

## 2.5 Discussion and open challenges

As a result of the systematic overview of the literature in the topic of alarm system performance measures, seven key categories were defined. These categories together with a short description and two examples for each, are depicted in Figure 2.17.

Enhanced monitoring of alarm system performance inevitably improves safety, reliability and profitability. As the chemical industry automatizes its operations, fueled by the Industry 4.0 revolution, the decreasing number of operators (and the increased workforce fluctuation) requires clear and actionable signals for the safe and efficient management of abnormal situations. Therefore, as it is evident to process experts, even though monetising the cost of individual alarm messages is hard, underperforming alarm systems pose a serious drawback for production sites.

We need to take special care of the Heisenberg-effect, meaning that the measurement of processes has a feedback effect on the process to be measured itself, similar to the world of quantum mechanics, where measurements cannot be taken without changing the system itself to some extent. The so-called Goodhart's law also states that if there is a financial interest in the development of an indicator, that indicator quickly loses its objective character; the actors do not focus on the goal behind the indicator, but on the indicator itself (When a measure becomes a target, it ceases to be a good measure).

The review of the alarm management performance metrics underscores the fact that industrial alarm systems are extremely complex and their measurement is not beneficially feasible with a single metric, but a multiobjective viewpoint is to be adopted. Even though several metrics are monitored by the industry, almost all consider the alarms as minimizable problems and from an optimization point of view, none of the monitored metrics tend to prevent the elimination of good quality alarm data. The measurement of the informativeness of alarm messages and the monitoring of operator actions must be incorporated to the key performance indicators of alarm management projects in order to ensure clear improvements in operational safety.

The analysis of the alarm management system of the industrial hydrofluoric acid alkylation unit before and after the alarm rationalization process underpinned how complex the task of the measurement of alarm system performance is. Prior to the analysis, the plant operators and process experts stated that the reliability and the effectiveness of the system increased significantly. However, this change was not traceable on the usual metrics monitored during an alarm management task but by monitoring the parameter changes performed by the operators. They relied much more on the alarm settings and this recursive monitoring and changing of the alarm limits ensures an up-to-date parameter setting for the alarm system.

## 2.6 Chapter summary

We should note that although the academia and the industry, as well as the different industries and industrial sites, monitor (or propose to monitor) different key performance indicators for the measurement of alarm system performance, no single solution has been found as a universal measure of efficiency. In the present study, a systematic overview and categorization of alarm system performance metrics were provided. Based on the aim and approach of the measurements, seven main categories were defined and discussed. A systematic difference in usage of the term of alarm system performance was revealed between the academia and industry: the academia mainly measures the optimality of alarm thresholds, while the industry, in the absence of the information of correct alarms, mainly measures the overall alarm load and its highest contributors. Moreover, the main metrics monitored by the industry proved to show an extremely unbalanced picture towards the quantitative measurements instead of qualitative ones. I presented how the informativeness and actionability of alarms should be taken into consideration. Finally, the analysis of the industrial hydrofluoric acid alkylation unit before and after the alarm rationalization process proved how extremely complex the rationalization process is and called attention to the problems of unilateral measurement of alarm systems.

The categorization of alarm system performance metrics provides the basis for the 1.3 subthesis of my first thesis finding, as these metrics support the maintenance of a well-functioning alarm system, which is a prerequisite for the application of advanced data-driven alarm management solutions. My thesis findings are summerized in Section 8.

# Chapter 3

# Decision trees for informative process alarm definition and alarm-based fault classification

Alarm messages in industrial processes are designed to draw attention to abnormalities that require timely assessment or intervention. However, in practice, alarms are arbitrarily and excessively defined by process operators resulting in numerous nuisance and chattering alarms that are simply a source of distraction. Numerous techniques are available for the retrospective filtering of alarm data, *e.g.*, adding time delays and deadbands to existing alarm settings. As an alternative, in the present chapter, instead of filtering or modifying existing alarms, a method for the design of alarm messages being informative for fault detection is proposed which takes into consideration that the occurring alarm messages originally should be optimal for fault detection and identification. This methodology utilizes a machine learning technique, the decision tree classifier, which provides linguistically well-interpretable models without the modification of the measured process variables. Furthermore, an online application of the defined alarm messages for fault identification is presented using a sliding window-based data preprocessing approach. The effectiveness of the proposed methodology is demonstrated in terms of the analysis of a well-known benchmark simulator of a vinyl-acetate production process, where the complexity of the simulator is considered to be sufficient for the testing of alarm systems.

**Note to practitioners:** Process-specific knowledge can be used to label historical process data to normal operating and fault-specific periods. Alarm generation should be designed to be able to detect and isolate faulty states. Using decision trees, optimal "cuts" or alarm limits for the purpose of fault classification can be defined utilizing a labelled dataset. The results apply to a variety of industries operating with online control systems, and especially timely in the chemical industry.

## 3.1 Introduction

Production costs, product quality and, most importantly, process safety are primarily affected by the efficiency of the monitoring and control of industrial processes. Consequently, the development of advanced fault detection and diagnosis methods is the focus of many research studies. Numerous well-advanced model- and process history-based methods are available for the purpose of monitoring and fault detection [59], however, still the simplest sensor-signal-based alarming technique is the most widespread and accepted state-of-the-art industrial practice due to its simplicity. This technique is not just considered too simple for complex fault detection and diagnosis tasks (a single alarm should be associated with each malfunction), but the poor design and maintenance of alarm systems make the situation even worse.

Another reason for poor alarm configurations is the simpilicity of their deployment. In distributed control systems (DCS) or supervisory control and data acquisition (SCADA) systems, the definition of alarm messages is determined purely computationally and entails no significant costs at all. As a result, alarm thresholds are arbitrarily and excessively defined by process operators. These poorly configured alarm limits are the direct cause of the numerous process alarms. Redundant (uninformative alarms for a single abnormality [9]) and chattering alarms [47] (chattering alarms are previously explained in Section 2.3.3) are quite common examples, but long-standing alarms are also well-known. In critical situations at plants, a flood of alarms overloads operators, significantly hindering or completely inhibiting the detection of root causes. The performance of alarm systems is degraded by not just the poor parametrization of alarms, but generally by the many alarmed variables as well.

The primary indicators for the operators of (chemical) technologies from faulty regions of operation are the alarm variables. In addition to the indication of quality issues and malfunctions associated with distinct process variables, complex fault detection and isolation efforts should be supported by the co-occurrence of alarm messages. Alarms are raised when a certain process variable exceeds its associated limits, more specifically, a low or high alarm sounds when the variable falls below its lower or exceeds its upper specification limit (alarm threshold), respectively. It should be noted that according to most industrial practices, multiple higher and lower thresholds can be defined as well - the respective alarm messages are referred to as, for example, high-high or low-low alarms, and although the present methodology can address this issue, it was decided to ignore these multiple upper and lower limits in most cases for the sake of simplicity. The purpose of these alarm messages, besides the most intuitive, univariate aim of drawing attention to the value of a single variable, is to indicate the presence of malfunctions (or their absence in the case of normal operating conditions). The philosophy of complex, multivariate and alarm-based fault diagnostic lies in the co-occurrence of multiple alarm messages, which form a characteristic fingerprint of certain malfunctions and indicate their presence. However, the analysis and interpretation of these alarm messages and, therefore, the way the operator extracts process-specific knowledge from them as well as their design approach are directly inter-dependent. Therefore, simply phrased, a two-way causality should describe the process of alarm design and the applied philosophy concerning fault identification: (1) the design of alarm messages determines how sounding alarms can be utilized for fault identification by the operators and, in turn, (2) the way the operators aim to utilize alarm messages for fault detection should determine the philosophy concerning the alarm design. Therefore, the key research question of the present work is how to facilitate the work of the operators of chemical technologies by alarm messages that indicate the faults of the technology.

## 3.2 Related work

Numerous approaches aim to improve the performance of alarm systems which vary with regard to the applied approach, the modified part of the alarm system, the required human workload, *etc.* Manual techniques, *e.g.*, the recursive investigation of bad actors, are time-consuming and highly labor-intensive [35].

Automatic methodologies usually aim to filter the incoming alarm messages using alarm deadbands [60] or time delays [61]. Another type of approach is to associate the existing alarm signals with additional information, rendering them more interpretable by the process operators. This is usually achieved by the grouping of the alarms into more informative classes by means of correlation analysis [62], frequent itemset [63] or sequence [64] mining, or classification techniques [65]. An important approach to enhance the usability of existing alarm systems is to build a root cause analysis solution upon them and track back the series of alarms to the original cause of abnormality [35], as it was carried out by a deep learning-based solution [66]. Moreover, an efficient approach is to concentrate on alarm floods, the periods containing the peak alarm rates, by for example sequence alignment supported pattern mining [67] and prediction techniques [68]. A nice overview of the recent advances in alarm data analysis is provided by Lucke *et al.* [69].

In contrast to studies that focus on the modification of alarm systems with additional features (delay timers, deadbands, *etc.*), this chapter will specifically describe how to set the alarm thresholds in order to formulate alarm messages, which are optimal for fault detection and identification. Both widely known industry standards, the EEMUA [3] and ISA [13], offer recommendations for the systematic steps and methods of the design of industrial alarms. Xu *et al.* proposed a design method for alarm systems in which, instead of the binary (alarm - no alarm) alarm limits, alarm limits are determined as fuzzy thresholds [70]. The performance of alarm systems is measured using three metrics, namely the false alarm rate, missed alarm rate and averaged alarm delay, moreover, these metrics are used to design alarm systems using a probability density function-based methodology as well [48]. Decision trees are applied for the design of new or the improvement of existing alarm systems, however, not for the design of alarm thresholds but rather for the identification of certain failures [71]. A four-step method was proposed for the design of plant alarm systems by Yan *et al.* [72] in which, following the cause and effect analysis of the assumed malfunctions, the variables for alarming are selected, the alarm limits set by adding a margin to the upper and lower bounds of the normal fluctuation range of variables and the rationalization of the alarm system carried out. Similarly, Takeda *et al.* proposed a three-step cause and effect-based methodology for the design of alarm systems [73] where first the set of fault origins to be distinguished is determined, followed by the selection of variables to alarm, and finally the alarm limits are set based on the effective rate, recall rate and timeliness measures according to [74]. Chang *et al.* proposed a methodology to improve the performance of existing alarm systems as a result of increasing their reliability by taking into account the issue of sensor redundancy and prioritizing the alarms based on the probability and impact of the potential hazard [75].

The main problem of the arbitrary and subjective definition of alarm messages is that this exercise can easily be counterproductive: the high number of alarm messages can lead to alarm fatigue, that is when the operators become desensitized to the alarm messages. Operators often define the alarms, not for fault detection but ulterior motives, *e.g.*, I have seen alarms defined to wake the operators up in certain situations at plants during night shifts.

Despite the high number of alarm messages and their inappropriate configuration, the important information is often still present in the data buried inside the avalanche of seemingly uninformative messages. Data mining techniques aim to reveal this information in the form of advanced alarm management techniques for fault detection. Chao and Liu combined timing constraint Petri nets with an alarm clustering mechanism in order to automatically identify faulty network elements [76]. Advisory information mainly based on heuristics from experienced plant engineers and the study of different alarms in the refinery was applied for fault detection and maintenance reports in an intelligent alarm management system at a petroleum refinery [77]. Dorgo *et al.* used the sequences of alarm messages and a long short-term memory (LSTM) unit-based deep learning model to determine the malfunctions that generate the alarm messages [66].

Although, such techniques (as with all of the advanced alarm management techniques) also require informative alarm messages, only a few techniques have been proposed where the alarm limit is explicitly determined in an advanced methodology. The root cause of the problem, namely the high false alarm rate is a result of the definition of control limits, has already been addressed [78]. Izadi *et al.* discussed that the philosophy of alarm design lies in a trade-off between false and missed alarm rates, moreover, a simple and systematic procedure is proposed for the design of alarm limits [79]. Yang *et al.* proposed a procedure for the determination of optimal alarm settings based on the similarities between correlation maps of physical process variables, their alarm history and the process connectivity information through causal maps [80]. A highly systematic design method was proposed by Tian *et al.* where the false and missed alarm rates as well as the average alarm delay are incorporated into an objective function and ant colony optimization is used to determine the optimal alarm limits [81]. Zhang *et al.* formulated the process of designing alarm limits as an optimization problem as well, taking into consideration the false and missed alarm rates in addition to the clustering weights of alarm variables [82]. Building upon these well-established and systematic techniques, my attention is drawn to a highly practical and yet often neglected objective concerning the design of alarm limits, namely that the alarms should indicate the presence of a malfunction. In this work, this issue is addressed by presenting a methodology that automatically and optimally assigns the alarm limits for fault detection, with a widespread technique of decision support, the decision trees.

## 3.3 Motivation & contributions

The motivation of this methodology is based on the assumption that the regimes of normal operation and different malfunctions should be distinguishable based purely on the alarm messages that have been raised, therefore, the problem is formulated as a data-based classification problem. However, unlike previous solutions, *e.g.* those proposed in [72] and [73], the selection of the variables to alarm and the determination of the limits of these process variables are carried out simultaneously in a single step. Moreover, these alarm limits are explicitly designed to support fault identification, unlike studies discussed in the Introduction. In order to obtain a model to determine the optimal alarm limits and describe the application of the generated alarm messages for fault classification, a decision tree-based technique is proposed. The decision trees are widespread for knowledge representation and decision support in the chemical industries [83] as they can be easily read and interpreted. For the generation of these decision trees, a data driven approach is used. Given that the well-known machine learning technique of decision tree classifiers is capable of determining the different operating regimes based on the measured process variables [84], in the present work, my aim is to utilize this technique for the data-based classification of different malfunctions as well as the normal operation and simultaneous determination of the associated alarm limits.

Based on this, the contributions of the present chapter are the following:

- The variables to alarm and the associated alarm limits are determined in a simultaneous step as discussed during the description of the decision tree

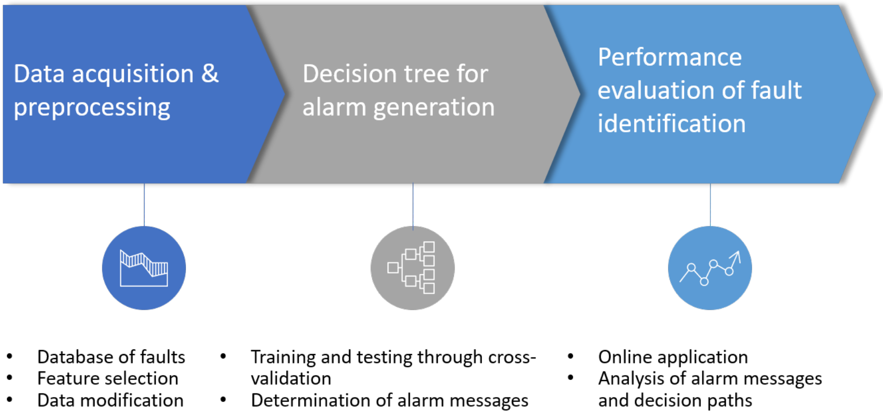

Data acquisition & preprocessing
Decision tree for alarm generation
Performance evaluation of fault identification
• Database of faults
• Feature selection
• Data modification
• Training and testing through cross-validation
• Determination of alarm messages
• Online application
• Analysis of alarm messages and decision paths

Figure 3.1: The framework of the proposed methodology starting from the process of data acquisition and preprocessing, leading on to the training of the decision tree and finally to the analysis of the resultant fault identification algorithm for online applications.

classifiers. This is the first study applying decision trees in alarm management for alarm-based fault classification.

- The presentation of how alarm-based, linguistically well-interpretable fault detection models are derived for the process operators by the utilization of decision trees.

The steps of the proposed methodology are presented in Figure 3.1. As in most of the data-based modelling problems, the core focus is on data acquisition and preprocessing, as these tasks primarily determine the performance of the trained decision tree. In order to avoid overfitting, the metaparameters of the trained decision trees are determined via tenfold cross-validation in the second step of the methodology. Finally, after the determination of the alarm messages, their performance for the identification of process faults is evaluated and their online applicability investigated.

The roadmap of the chapter is as follows. The alarm management background of alarm generation and the alarm-based classification of operating regimes are described and mathematically formulated in Section 3.4.1. Two different data preprocessing methods are presented and compared for the generation of alarm limits through the decision tree-based classifier (Section 3.4.2). The variables to alarm and the associated alarm limits are determined in a simultaneous step as discussed during the description of the decision tree classifiers (Section 3.4.3). The performance of the trained classifiers are measured by a two-step evaluation method as presented in Section 3.4.4. The results demonstrate how the generated decision trees and alarms can be used to distinguish between the different states of the process (normal or faulty) by a linguistically well-interpretable model, which is ready to be used by the operators or build advanced alarm management solutions (Section 3.5). In the present work, the performance of the methodology of alarm generation presented in Figure 3.1 is tested on the data generated by the simulator of a vinyl acetate production technology [85]. As an outlook of the present research, the results are followed by the discussion of the limitations and development opportunities (Section 3.6) and finally, some concluding remarks (Section 3.7).

## 3.4 Informative alarm messages for the detection of faulty operation states of complex processes

In this section, the alarm management background of fault detection and identification is discussed and merged with the mathematical formulation of classification-based alarm generation and fault identification. The section ends with the theoretical background of decision tree classifiers required to understand the methodology.

### 3.4.1 Decision tree classification of regions of operation

In the majority of cases, the operators of the technology look for the co-occurrence of certain alarm messages as a fingerprint-based primary symptom of process malfunctions. The present work follows this simple cognitive model of fault detection of the human mind.

Let us assume there are $n_C$ types of operations ($n_C - 1$ malfunctions and normal operating conditions) in the analysed system that should be identifiable. Although the normal region of operation and the different faulty conditions should be separable according to the alarm messages that occur, the definition of informative alarm thresholds for a complex, multivariate process is a highly labor-intensive process and the consideration of every possible scenario cannot be guaranteed even with the systematic work of the process experts. Therefore, as learning from examples is one of the most important features of machine learning techniques and the bottleneck of expert system development, such techniques are widely available for the identification of process models. The historical data of the process is required to learn the process model using the historical scenarios. In the present context, the considered information is the matrix of the measured process variables, which is logged in most of the modern DCS or SCADA process control systems and a logged list of historical fault types and occurrence time stamps.

Let us assume an $n \times m$ measurement matrix $\mathbf{X}$, where $n$ denotes the number of samples (sampled from the measurements from an arbitrary sampling time base) and $m$ represents the number of measured process variables. Therefore, $\mathbf{X}$ is composed of $\mathbf{X} = [\mathbf{x}^1 \dots \mathbf{x}^m]$ variable vectors with the dimensions $n \times 1$. Moreover, the vector $\mathbf{X}_i$ is the vector of the measured variables at sample point $i$ ($\mathbf{X}_i = [x_i^1 \dots x_i^m]$). The aim of fault detection and diagnosis is to classify the process into the $C_j$ class ($j = 1 \dots n_C$) based on the arbitrary feature vector $\mathbf{S}_i$ at time stamp $i$, and indicate the presence of either the normal state or $n_C - 1$ types of malfunctions. Mathematically, the function $f$ in Equation 3.1 is constructed as:

$$y_i = f(\mathbf{S}_i) \tag{3.1}$$

where $y_i \in \mathbf{C} = [C_1 \dots C_j \dots C_{n_C}]$ and $\mathbf{S}_i$ is the feature vector of arbitrary size, where the features are composed of the information available at time stamp $i$ in the measurement matrix $\mathbf{X}$. This information can be of various source of the technology utilized by the general classification-based fault detection function $f$. In the present chapter, the classification function $f$ is constructed using the binary decision tree classifier. The tool is not just suitable for the classification of the measurements into different classes describing the state of the process, but for forming informative alarm thresholds as well using the defined decision boundaries. Moreover, an advantage of the decision tree classifiers is that the $m_a$ variables to alarm and their respective alarm limits (even with multiple alarm limits for a single variable) can be chosen in a single step.

For a simple and didactic example of the proposed methodology, consider the explanatory case study presented in Figure 3.2. The operators of the technology monitor the occurrence of alarm messages to detect the presence of abnormalities. Hence, the aim of alarm messages is to distinguish normal operating conditions from two faults of the reactor: an increased temperature can be caused by the temporarily increased temperature of the reactor coolant (Fault 1), however, if it

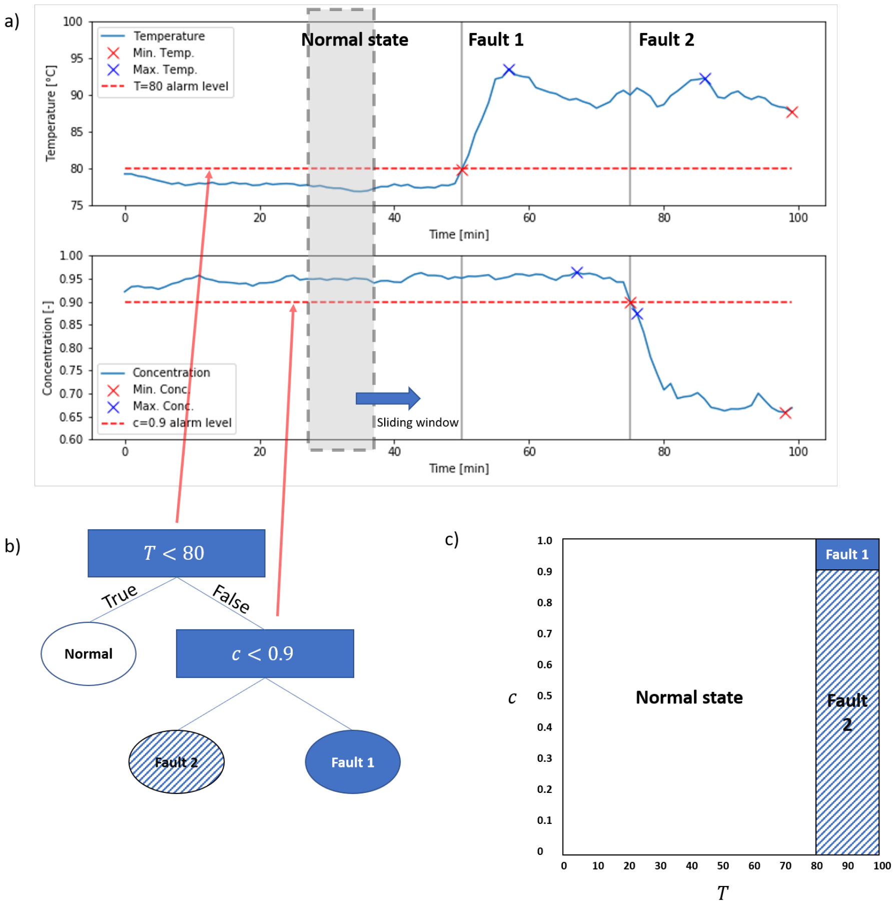


FIGURE 3.2: (a) An example of a time series data of process variables concerning temperature (top) and concentration (bottom). By applying the proposed methodology, the alarm limits can be determined, denoted by red dashed lines. The input data of the decision tree can be formed of the minimum and maximum values of the process variables during the malfunctions, denoted by red and blue crosses, respectively, and in this case, for the online application of the generated classification algorithm, the minimum and maximum values of process variables in a sliding window are used. (b) Example of a binary decision tree. (c) The decomposed feature space.

is not just temporary, *e.g.*, the cooling cycle underperforms due to the problems of the heat-exchanger network, then the product can be degraded as well which is reflected in the decreased concentration of the product (Fault 2). An arbitrary number of faults can be collected based on the expert knowledge of the process and once a historical (or simulated) database of the recorded process variables of each malfunction is available, like presented for the variables of temperature and concentration in part *a)* of Figure 3.2, a binary decision tree can be trained to classify the process variables as presented in part *b)* of Figure 3.2. As the decision trees determine the most informative features of every step following a greedy approach as well as the respective decision limits on these variables, these decision functions can be analogously transformed into alarm limits. Moreover, the decision tree for alarm-based fault identification is ready to be used by the process operators and the feature space of the multivariate decision-tree model can be visualized as well as presented in part *c)* of Figure 3.2.

The question is how to define the input dataset of the decision tree classifier. The application of the directly measured process data can be counteractive as the process may not contain enough information for the correct classification of every time sample. Another approach is the application of the characteristics with regard to the extrema of the variables in case of malfunctions as denoted by the red and blue crosses of Figure 3.2 for Faults 1 and 2. However, the classification may be more accurate since the online application of the trained classification algorithm requires a sliding window in which the extrema form the input of the decision tree classifier. The generation of the training data is discussed in detail in the following section.

### 3.4.2 Data generation & preprocessing

Once the methodology has been clarified, the next step is the training of the decision tree classifier to generate the informative alarm limits. To understand the value and information content of process data, it is necessary to place ourselves in the shoes of the operator of a chemical process who is searching for the occurrence of a fault $C_j$, the effect of which is identifiable for a well-defined temporal period $\tau_j$. Our aim is to detect the presence of abnormalities and identify the type of malfunction as early as possible. Depending on the complexity of the malfunction and our routine in the identification of the fault taking place, different data attributes can be taken into consideration: the simple extrema of process variables, and different symptoms of the malfunction that evolve over time after its occurrence, *i.e.*, the sequence of alarms or even the dynamics of process variables. One thing they have in common is that the human mind considers these symptoms as a fingerprint of the presence of a malfunction and connects them with "and" logical functions, in a similar way to decision tree classifiers. Therefore, the question is what data should form the input of a decision tree that aims to formulate our alarm limits and their subsequent classification? In the present work, two different types of input data are tested as depicted in Figure 3.3 and discussed below:

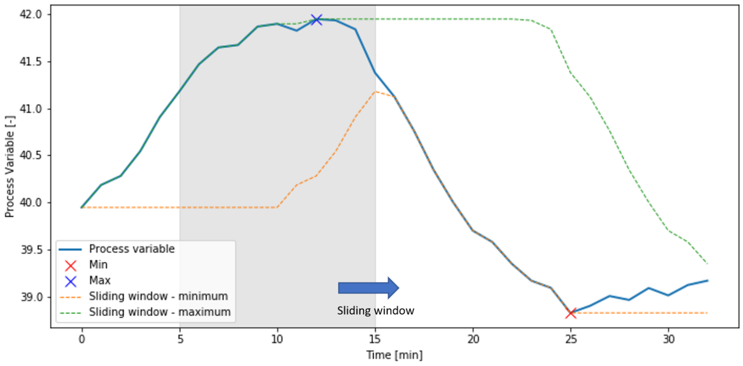


FIGURE 3.3: The time series data of a process variable and its extrema applied as the possible input data for the analysis. The sampled process variable (solid blue line) is used for the *Continuous data set* while the *"Max-min" data set* applies the minimum and maximum values of the process variable (red and blue crosses, respectively). The classification algorithm trained on the "Max-min" data set utilizes a sliding time window during the online application and classifies the process into the various states based on the minimum and maximum values of each process variable in the respective time window (yellow and green dashed lines, respectively).

**"Max-min" data set** If a well-known malfunction has occurred and/or is easily identifiable, first, only the extrema of specific process variables are checked to answer questions like: does the temperature exceed a certain value or does the concentration decrease beyond an acceptable limit? This is a very intuitive approach where only the extrema are considered, however, the plethora of information is not taken into account. Therefore, in this methodology, the alarm messages are designed based on the minimum and maximum values of process variables following malfunctions. Mathematically, the feature vector $\mathbf{S}$ for training the decision tree to capture the variable extrema of fault $C_j$ is constructed from the extrema of the individual measurements in a time window $\tau_j$ after the occurrence of the fault (assuming it is at time stamp $i$), $\mathbf{S} = [max(\mathbf{x}^1_{[i,i+\tau_j]}), \ldots, max(\mathbf{x}^m_{[i,i+\tau_j]}), min(\mathbf{x}^1_{[i,i+\tau_j]}), \ldots, min(\mathbf{x}^m_{[i,i+\tau_j]})]$, where the subscript $[i, i + \tau_j]$ denotes the time stamps between $i$ and $i+\tau_j$. From a technical point of view, the number of sample points is reduced to one per fault, however, the number of process variables has doubled when compared to the original due to the simultaneous handling of the minimum and maximum values of process variables. While a significant amount of information is lost along with the temporality of process variables, the critical sample points are preserved. As the temporality of the information is lost, the fault detection techniques that should be utilized in advanced alarm management are simplified as well: instead of frequent alarm sequences, the analysis of frequent itemsets is also sufficient (the chronological order can be neglected). The maximum and minimum values of a process variable are indicated by the red and blue crosses in Figure 3.3, respectively. As the application of data models should be similar to the training environment, in this case, for online fault identification, a sliding window is considered in which the extrema of each process variable are determined and used as the inputs of the decision tree trained for fault identification. This sliding window is indicated by the grey bar, while the sliding minimum and maximum values are represented by the yellow and green dashed lines in Figure 3.3, respectively.

**"Continuous" data set** In the case of a highly complex malfunction, the extrema may not be sufficient to identify the presence of a malfunction, and the dynamics have to be incorporated as well. One can simply input the process variables (sampled on the basis of an appropriate sample time by the process control unit) at every time instance of the operation and apply the trained classifier to determine the operating mode of the process. Therefore, mathematically, the feature vector at time stamp $i$ is simply composed of the measurement vector as $\mathbf{S}_i = \mathbf{X}_i$. Although this information would provide a timely and detailed picture of the present state of the process, some variables may be in informative states, while others not, which can easily overload the operator (and our model) with noisy and highly variable information due to the stochastic nature of production processes. The alarms designed based on this temporal information provide an instantaneous picture of the process, therefore, require the operator to have an exceptionally deep knowledge of the process as several highly varying scenarios can occur for a single type of malfunction. However, in this case, the temporality of the data is preserved and in advanced alarm-based fault detection techniques, this thorough information renders the designed alarms suitable for correlation and frequent sequence analysis. The continuous process variables are indicated by the solid blue line in Figure 3.3.

This idea is clarified and summarized in Table 3.1.

The key bottleneck of the presented methodology is the required measured or simulated *labelled* process data, therefore, a historical database of malfunctions that have occurred is necessary. Of course, after the occurrence of a malfunction, its effect can be prolonged and our aim is to determine the presence of the malfunction in this period. This time period can be determined by advanced anomaly detection techniques or using expert knowledge of the process.

| ID | Name | Information content | Advantage | Disadvantage | Fault detection logic | Advanced alarm management tools |
|---|---|---|---|---|---|---|
| a) | "Max-min" | The minimum and maximum values of process measurements after malfunctions | Highly informative and simple | Loss of temporality and information content | Simple rule of thumbs, logic rules for alarm co-occurrences | Frequent itemset analysis |
| b) | "Continuous" | The minute-based process measurements | Timely and detailed | Information overload, high degree of variability | Extensive knowledge of the process, scenario-based | Correlation analysis, frequent sequence analysis |

Table 3.1: The data utilized for alarm design and the characteristics of the resultant alarm system

### 3.4.3 Construction and interpretation of decision trees

Throughout the chapter, binary decision trees are applied to determine the informative (and in this sense, optimal) alarm rules. A binary decision tree consists of two types of nodes: (i) internal nodes with two children and (ii) terminal nodes without children, referred to as leaves. Each of the internal nodes is associated with a decision function to indicate which of the two outcome nodes is to be visited next. According to conventions, the true outcome of the decision is visualized on the left branch, while the right one indicates the answer "no" as presented in part *b)* of Figure 3.2. Each terminal node represents the output of a specific input that leads to this node throughout the decisions, therefore, in classification problems, each terminal node contains the label of the predicted class.

Therefore, the $n$ time samples in the measurement matrix $\mathbf{X}$ (and thus the composed feature matrix $\mathbf{S}$) are composed of a set of $D$ cases. The decision tree algorithms partition the data $D$ into subsets $D_1, \dots D_I, \dots D_N$ following tests $T$ of mutual outcomes $T_1, \dots T_I, \dots T_N$, where $D_I$ comprises the cases with outcome $T_I$. The applied general decision tree algorithm (we applied the Scikit-learn-based implementation [86]) generates binary decision trees (tests with binary outcomes). For numeric (not categorical) variables with regard to the $i^{th}$ sample point on the $k^{th}$ feature variable, the tests are written in the form $s_{k,i} \leq \kappa_k$ form, where $s_{k,i}$ is the feature variable $k$ in sample point $i$ and $\kappa$ denotes the threshold from the selected informative alarm limits and can be determined based on two different splitting criteria as described in the following. Here, it should be mentioned that the corresponding limits on the original process variables can be derived from the limits on the feature variables.

Decision trees aim to construct leaves (subsets) associated with a single class, *i.e.* most cases originate from a single class. Therefore, the splits and thresholds of the tree are determined to achieve as pure leaves as possible. In decision trees, the pureness of a set is usually determined using two measures, the *information entropy* and *Gini index* [87]. The information entropy is presented as follows:

$$Info_S(D) = -\sum_{I=1}^{N} p(D,I) \times \log_2(p(D,I)) \tag{3.2}$$

where the probability $p(D, I)$ denotes the proportion of cases in $D$ that belong to class $I$. The entropy of a set of elements belonging to the same class is 0, while the entropy of a set of elements uniquely distributed between all classes is maximized and equal to 1. However, the logarithmic function is computationally expensive, therefore, highly disadvantageous in the case of larger and more complex problems. In order to handle this issue, the Gini index was proposed:

$$Info_{Gini}(D) = \sum_{I=1}^{N} p(D, I)\,(1 - p(D, I)) \tag{3.3}$$

Similarly to information entropy, the Gini index is maximized if the classes are perfectly mixed in the set.

Once the pureness (or mixedness) of a set is measured, the decision tree algorithms split the original set of data by defining the decision threshold of the feature that results in the highest *information gain*, *i.e.*, achieves the highest increase in the pureness of the sets. The information gain on parent set $D_p$ by splitting it into sets $D_{left}$ and $D_{right}$ according to test $T$ is calculated using Equation 3.4:

$$Gain(D_p, T) = Info(D_p) - \frac{|D_{left}|}{|D_p|} Info(D_{left}) - \frac{|D_{right}|}{|D_p|} Info(D_{right}) \tag{3.4}$$

where $|D_{left}|$ and $|D_{right}|$ denote the cardinality (the number of individual elements) of sets $D_{left}$ and $D_{right}$, respectively, and the measure $Info$ can either denote the information entropy from Equation 3.2 or the Gini index from Equation 3.3. Naturally, this equation is only valid for binary decision trees, in the case of general decision trees with multiple outcomes, when a decision is to be made the weighted information of every subset must be subtracted from the information measure of the parent set. This splitting procedure is repeated iteratively at each child node until the samples at each leaf belong to the same class. However, this can easily result in a very deep and overfitted decision tree, thus, the construction of the tree is often limited by fixing the *maximum depth* of the fitted tree. (The depth of a decision tree is the length of the longest path, *i.e.* the number of nodes from the root of the tree to a leaf.) The overfitting of the decision tree can be avoided by validating the results, by measuring the accuracy of the model on the training set and the validation (as well as the test) set as well. During the validation, the hyperparameters of the decision tree can be set as the function of the complexity of the input data: more complex data generally requires a deeper, more complex tree as well, while a few decisions may be enough for dataset with a simple structure. Here it should be noted that several parameters and approaches are available to restrict the size of a decision tree. Finally, as in this case the resultant subsets $D_1, \dots D_I, \dots D_N$ generated by the tree may contain members from several classes, the predicted label in the case of each subset is defined as the class with the highest number of elements in the subset:

$$C_I = argmax_j \left(D_I\right) \tag{3.5}$$

As can be seen in the present section, during the training process, the decision tree classifiers determine what variable $i$ to alarm from the $m$ available variables and also generate the associated decision limit, $\kappa_i$. Here, it should be noted once more that the algorithm is able to define multiple "cuts" on a process variable, however, for the sake of simplicity, these are not indicated at every step. Therefore, the proposed classifier is ready to define the alarm limits, moreover, the resultant data model is ready to be used as an alarm message-based decision model by the operators.

### 3.4.4 Measuring the performance of the designed alarm system

The performance of the proposed method is derived back to the evaluation of the decision tree-based classifier. The resultant decision trees are evaluated over two steps. First, a tenfold cross-validation is applied and an accuracy measure calculated:

$$accuracy(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} 1\left(\hat{y}_i = y_i\right) \tag{3.6}$$

where $1(x)$ denotes the indicator function, which is equal to 1 if its argument is true and 0 otherwise, moreover, $y, \hat{y} \in \mathbf{C} = [C_1 \dots C_j \dots C_{n_C}]$ and indicates the true and predicted class of the input data at time instance $i$, respectively.

Secondly, a more practical evaluation of the classification performance is carried out to mimic the online application of alarm messages; a simulated data set was used for each fault as the input of the trained decision trees. The faults start at the beginning of the analysed data set ($0^{th}$ minute, except for Fault 0 when the operation was faultless) and their temporal period follows the logic of the training set. The algorithms should detect the fault as early as possible. The following question is raised: how should these algorithms be applied during production? In the case of the model trained on the "Continuous" data set, the answer is evident, the timely measurements form the input of the decision tree classifier. However, in the case of the "Max-min" data set, the sampled measurements do not follow the philosophy of the alarm design. Instead, a rolling time window is proposed, where the minimum and maximum values of the past time window-based time period always form the input of the algorithm. In this sense, the algorithm utilizes the maximum amount of information available in the time window. Then the type of the detected faults and when of the first fault was detected are measured.

## 3.5 The case study of the vinyl acetate process

In the present section, first, the data generation process on the vinyl acetate process simulator is introduced. Then, the described alarm design methodology created for efficient fault identification is tested on the simulated data set of a vinyl acetate production process.

### 3.5.1 The simulated data set

The description of the applied benchmark vinyl acetate process simulator, together with its modifications is presented in the Appendix in Chapter 9. To simulate the operation of a chemical production process, various malfunctions and operations are incorporated in the simulation applied for the generation of the analysed data. The list of the applied faults and their respective IDs can be seen in Table 3.2. During the simulation, the applied faults either occur following a lognormal distribution with a mean of 15 and variance of 5 minutes (Faults 1-3) or last for exactly 5 minutes (Faults 4-6). The type of the malfunction is chosen randomly every 120 minutes following a uniform distribution between the different types of malfunction. In order for a malfunction to occur multiple times, and for the noise on the process as well as how it changes over time to be taken into consideration (*e.g.*, as a result of catalyst deactivation), a fault is simulated multiple times and each fault occurred at least 100 times in the simulation. It is important to highlight that following this logic, a single occurrence of a fault is sufficient to determine the alarm levels based on a binary decision tree. However, to consider the different courses of the malfunctions, multiple simulations were applied. Moreover, the proposed simulation can be considered as the extraction of faulty periods over a longer operational period. A special case occurs if faults co-occur in the system. If some faults are connected and can frequently co-occur in the technology, then these faults should be implemented together in the simulation to track their cross-effect on the process.

In order to train the classification algorithm and evaluate its performance, the labelling of the operation with the fault IDs is required. Naturally, the fault should be identified as soon as possible following its occurrence, hence the beginning of the labelling starts with the occurrence of the fault. However, how long the effect of a fault can be detected for is harder to determine: in the present work, a 4-minute-long time window is applied after the end of the fault, except for Fault 4 where the process is exposed to the effect of the fault for a longer period of time, here, the time window is 20 minutes to be exact. These time periods were determined by the expert-based examination of the process. Moreover, only the 27 sensor variables extracted in the MATLAB code are used during the classification; these variables, together with their abbreviations and units, are listed in the Acronyms in Table 10.2. Then in the "Continuous" data sets, these minute-based measurements are considered, while their minimum and maximum values are used in the "Min-max" data set.

### 3.5.2 The alarm levels of the vinyl acetate process simulator

The decision tree classifier obtained by the training on the "Max-min" data set is depicted in Figure 3.4. The tree is considered sufficiently simple, yet accurate, however, only the maximum depth of the tree was set at five as the input parameter of the training algorithm.

The sampled continuous process variables form a significantly more complex training data set. Consequently, the obtained decision tree is highly complex, moreover, without any restricting input parameters, it could handle rare cases with just a

| Fault ID | Fault |
|---|---|
| 0 | no disturbance |
| 1 | setpoint of the outlet temperature of the reactor decreases by 8 °C (from 159 to 151 °C) |
| 2 | setpoint of the outlet temperature of the reactor increases by 6 °C (from 159 to 165 °C) |
| 3 | the vaporizer liquid inlet flowrate increases by 0.44 kmol/min (from 2.2 to 2.64 kmol/min) |
| 4 | HAc fresh feed stream lost for 5 minutes |
| 5 | $O_2$ fresh feed stream lost for 5 minutes |
| 6 | column feed stream lost for 5 minutes |

TABLE 3.2: The data utilized for alarm design and the characteristics of the resultant alarm system

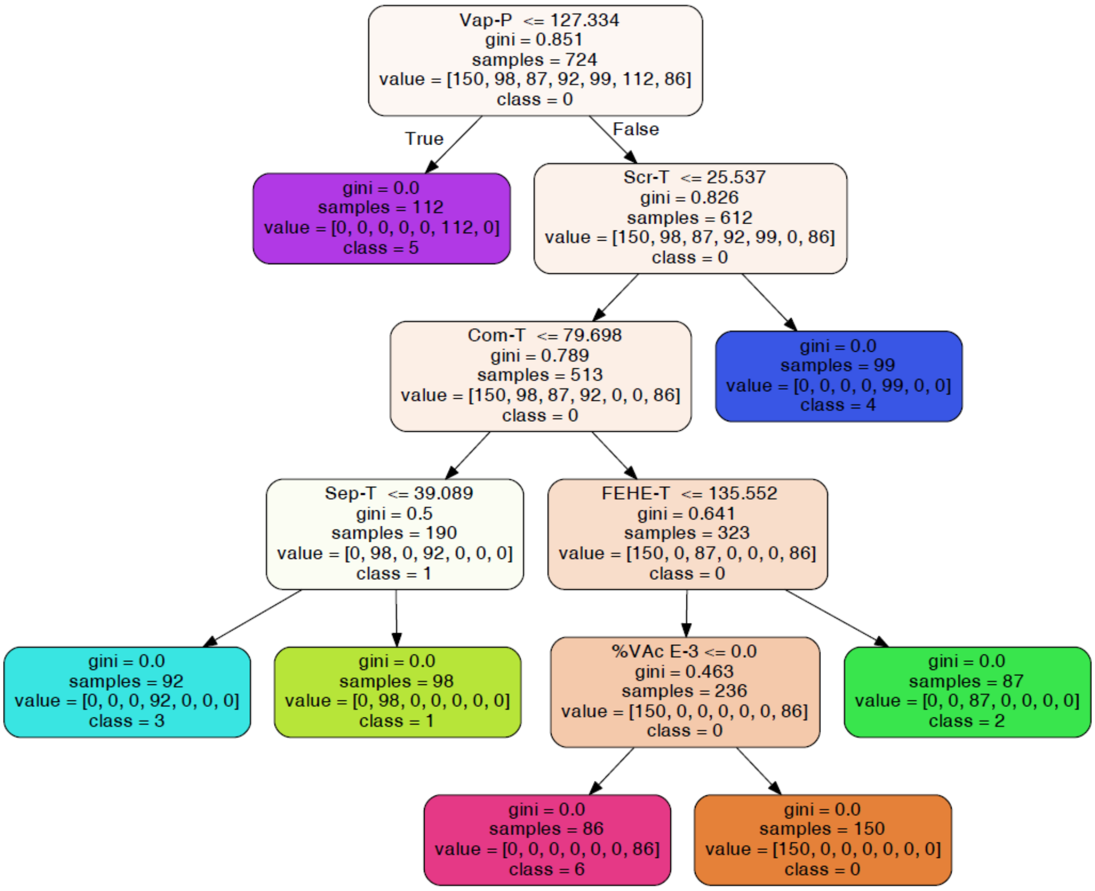


FIGURE 3.4: The decision tree classifier obtained by the training on the "Max-min" data set.

few sample points on each leaf. Therefore, after some trial-and-error tuning of the parameters, the maximum depth was set at seven, the maximum number of leaf nodes at 14 and to avoid the over-specific leaves, the minimum number of samples on a leaf was 50. Moreover, in this case, the entropy information measure was chosen over the Gini index. The final decision tree is depicted in Figure 3.5.

The resultant decision trees were evaluated over two steps. Firstly, a tenfold cross-validation was applied and the accuracy measure investigated according to Figure 3.6. Secondly, the online application of the algorithms was mimicked and the recorded data set of each simulated fault or operating state formed the input of the algorithms. The faults occur at the beginning of the analysed dataset ($0^{th}$ minute, except for Fault 0 where no fault is implemented) and their temporal period follows the logic of the training set. For the model trained on the "Continuous" data set, the continuous process variables were sampled on the same minute-basis as the training set, while for the one trained on the "Max-min" data set, the minimal and maximal values of the process variables within a sliding time window formed the inputs of the decision tree. The length of this sliding time window is a trade-off between accuracy and speed: choosing a long time window with regard to the effect of the fault will basically result in a very precise classification performance, however, this renders the classification algorithm exceptionally slow. I have intended to apply sensitivity analysis between a 1 and 20-minute-long time window, however, in this case the results were not significantly dependent on the length of the time window. Hence, after some expert-based investigation, the time window was set at 5 minutes. However, it is important to highlight that the choice of this parameter should always follow some expert-based guidelines or a detailed sensitivity analysis. In the case of expert-based investigations the rule-of-thumb is to set the time window compared to the average time constant of the process.

The detected operation states (or faults) and when a fault was first detected are summarized in Table 3.3. The detailed classification performance of the algorithms, *i.e.* their classification based on the measured process variables over time, can be seen in Figure 3.7.

According to the results, the decision tree trained on the "Continuous" data set provides rapid identification of the faulty states but tends to wrongly classify the operation in some cases. While the correct fault class is present in almost all of the cases, perfect classification cannot be expected from the algorithms as the identifiability of the faults is not investigated. Even in the case of misclassifications,

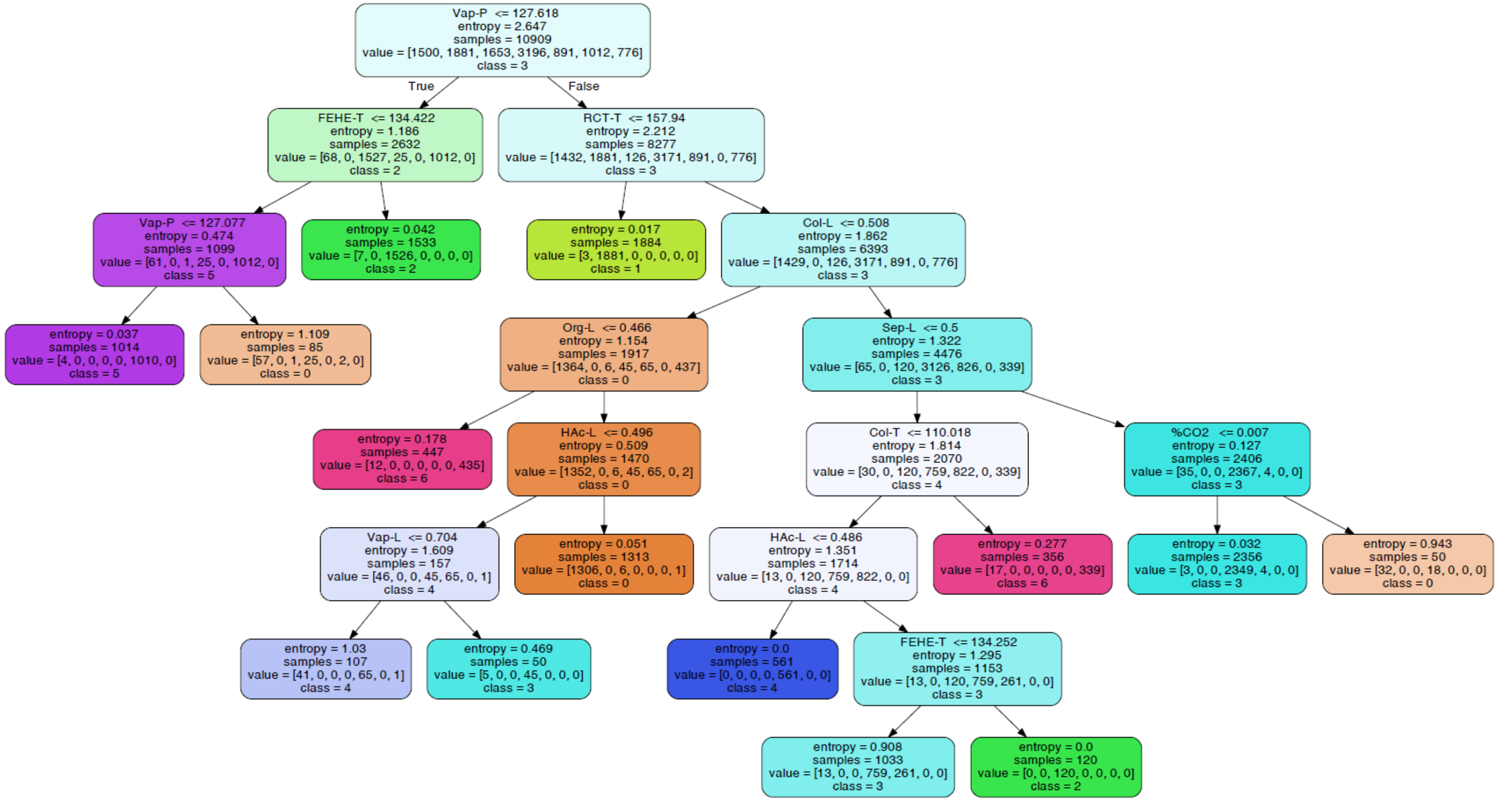


FIGURE 3.5: The decision tree classifier obtained by the training on the "Continuous" data set.

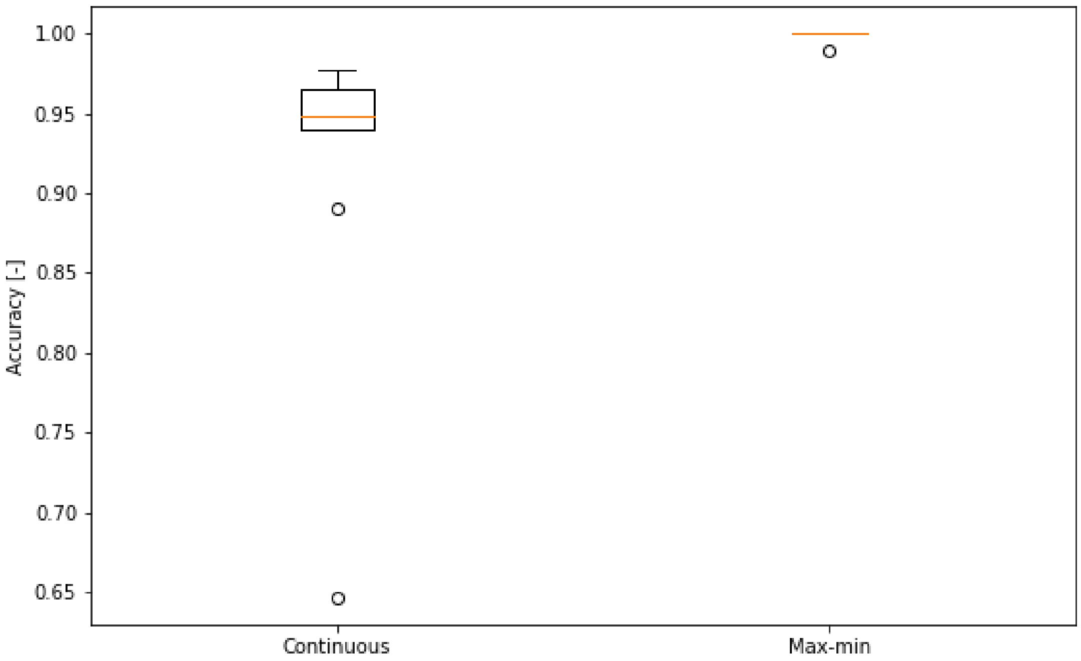


FIGURE 3.6: The accuracy measure calculated according to Equation 3.6 during the tenfold cross-validation.

| | "Continuous" | | "Max-min" (tw = 5 min) | |
|---|---|---|---|---|
| Fault ID | Detected fault | Time of first detection [min] | Detected fault | Time of first detection [min] |
| 0 | 0 | - | 0 | - |
| 1 | 0, 1, 2 | 0 | 0, 1 | 2 |
| 2 | 0, 2 | 1 | 0, 2, 5 | 4 |
| 3 | 0, 3, 6 | 0 | 0, 1, 3 | 18 |
| 4 | 0, 4 | 0 | 0, 4 | 2 |
| 5 | 0, 2, 5 | 0 | 0, 5 | 0 |
| 6 | 0, 3, 4, 6 | 1 | 0, 6 | 3 |

TABLE 3.3: The results of the classification algorithms. The type of the detected faults and time of the detection of the first fault. (tw - time window)

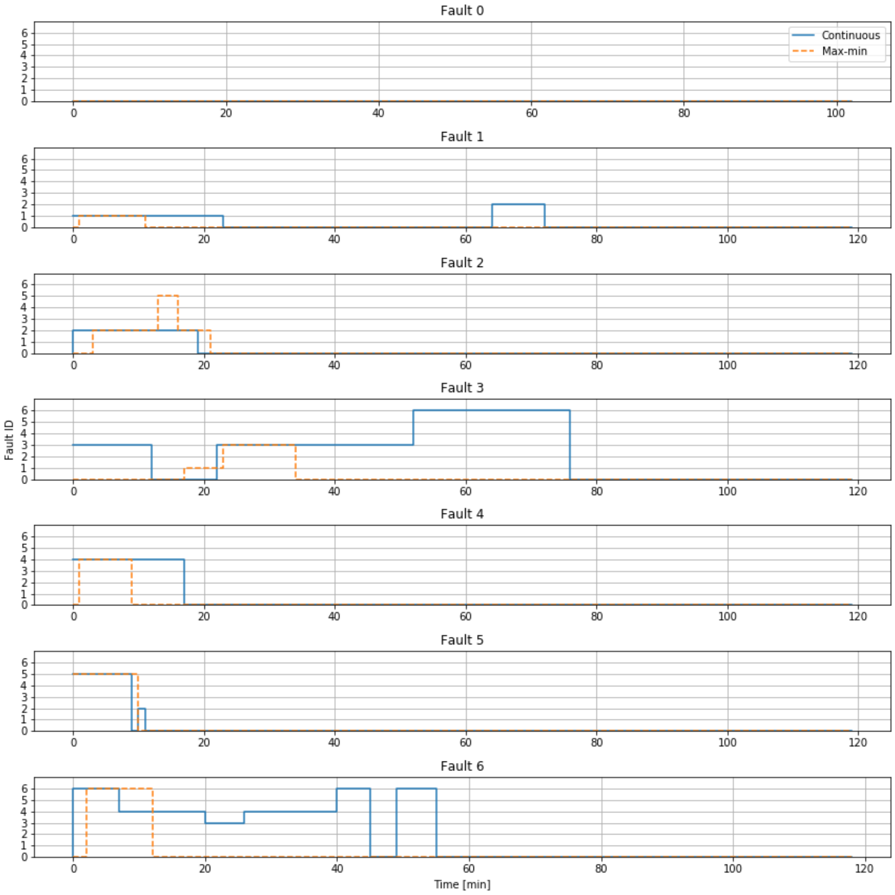


FIGURE 3.7: The trained classification algorithms tested on simulated faults. The minute-based process data formed the input of the decision tree trained on the "Continuous" data set, while the minimum and maximum values measured over a 5-minute-long rolling time window were the inputs of the decision tree trained on the "Max-min" data set.

the first fault to be identified is almost always from the correct class, except for Fault 6, which only occurs in the second minute of operation. In the case of the decision tree trained on the "Max-min" data set, the speed of fault identification is slower, however, the classification performance is much better, a misclassification only occurs in the case of Fault 2, which is mistakenly classified as Fault 5, while Fault 3 is mistakenly classified as Fault 1. It is important to notice the fact that despite the numerous alarm messages applied for fault detection and identification in complex systems, both approaches provided a solution for fault identification using just a very limited number of alarms (of course, however, the complexity of the simulator falls behind the complexity of a modern production plant with numerous failure modes). Yet, in the case of the decision tree classifier trained on the "Max-min" data set, only six alarms are sufficient, while in the case of the decision tree trained on the "Continuous" data set, 13 alarms are required to identify the six malfunctions and the normal operation.

To interpret the results, consider Fault 3 and how it is misclassified by the decision tree trained on the "Max-min" data set. Fault 3 is the increased flowrate of the inlet liquid of the vaporizer, which is captured by the alarms concerning the temperature of the separator and compressor, which are two closely connected units. The misclassification of this fault with Fault 1 is not a surprise as Fault 1 comprises a decrease in the temperature setpoint of the reactor, moreover, the increase in the flowrate of the inlet liquid of the vaporizer in Fault 3 most likely causes a drop in temperature in the reactor as well. Of course, this similarity is also captured in the data sets, as is presented in Figures 3.8 and 3.9 for Faults 3 and 1, respectively. As can be seen, the algorithm classified well the process based on the extrema: during Fault 3, the temperature of the separator was below its respective limit, while this was not the case when Fault 1 occured. However, when the temperature of the compressor is below its limit, and therefore, a decision with regard to the value of the separator temperature is reached, its value at first exceeds the separator temperature limit. As a result, Fault 1 is identified, which later "evolves" into Fault 3, when the temperature drops below this limit. This issue can be manually fine-tuned. Given the data, it can easily be observed why the detection of this fault also occurs as late as 18 minutes into the process: the temperature of the compressor is above its limit which prevents the decision model on a higher level from reaching these leaves of the tree.

## 3.6 Limitations and development opportunities

The proposed methodology has been proven to be applicable for the determination of which variables to alarm and identification of the most informative alarm limits on the chosen variables. Moreover, the resultant alarm-based data model is linguistically well-interpretable, providing a potentially efficient tool for the operators. However, some limitations and opportunities to enhance the process are raised by the authors. First of all, the methodology is based on process data labelled by the fault/operation occurring at the time, but the collection of this information can be challenging in some cases. Secondly, even though the tests proved that the methodology is appropriate for fault detection as well as diagnosis and provide the primary alarm management settings, expert knowledge of the processes

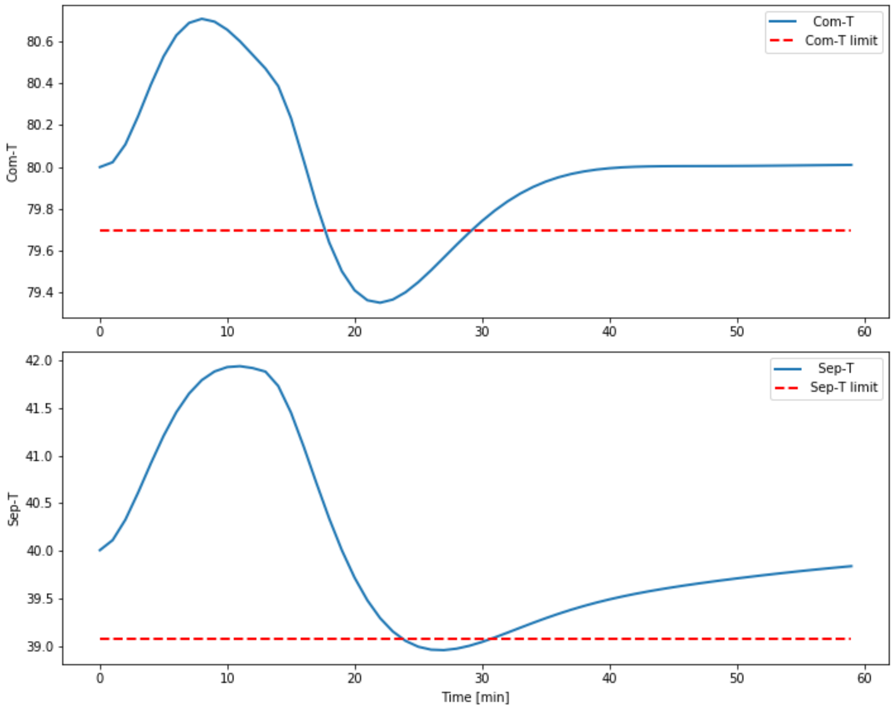


FIGURE 3.8: The temperature data of the compressor and separator during Fault 3.

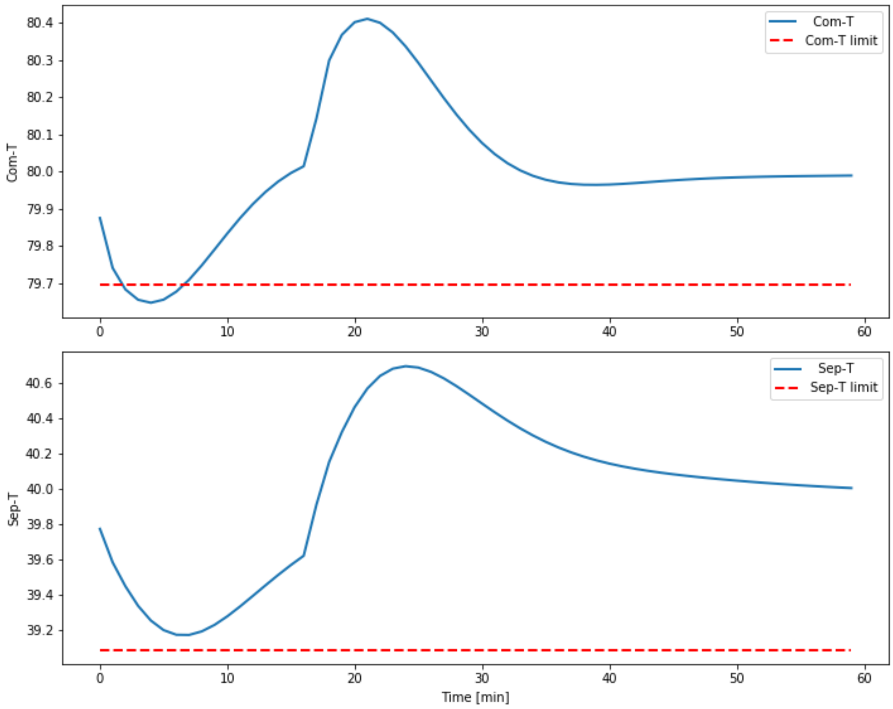


FIGURE 3.9: The temperature data of the compressor and separator during Fault 1.

in modern, safety-critical systems cannot be neglected. The proposed methodology should only provide suggestions for the design of alarm limits. Moreover, the optimal trade-off between false alarms and missed alarms varies between different industrial sectors: the malfunction of a plant in light industries may not cause as serious consequences as the fault of a nuclear power plant, therefore, stricter or wider thresholds and additional alarms should be defined to facilitate cheaper or safer production, respectively.

Finally, the trade-off between the "Continuous" and "Max-min" data set-based approaches in more complex systems does not only lie in the speed and accuracy of classification. In the case of the "Max-min" data set, the sequential information of the alarm messages is lost. In this case, the alarm messages should occur together, while the "Continuous" data set-based approach provides the opportunity for the symptoms to evolve towards the identification of the malfunction. This highlights the future direction of this research: the time value of the information can either be taken into consideration by applying modified and further enhanced decision trees, or the problem concerning the design of alarm limits can be formulated as a multi-objective optimization problem with objectives such as the accuracy and timeliness of classification, the number of alarm messages required for classification, *etc.*

## 3.7 Chapter summary

As a result of the presented research, some very intuitive and fundamental cornerstones concerning fault detection and alarm management are confirmed, namely that the definition of alarm limits should support the process of fault detection as well as identification and the goal of alarm-based fault identification can only be achieved by a relatively small number of alarm messages (in the case of the six faults of the VAC simulator, 6 and 13 alarms depending on the training methodology). Moreover, it is very important to note the fact that the applied advanced alarm management technique and therefore, the analysis of frequent sequences or itemsets should be chosen based on the connection between faults and alarms: the alarm limits designed for the extrema of the process variables support frequent itemset mining, while the alarm limits designed for the sampled process variables generate alarm messages that should be analysed as frequent sequences.

Regarding the efficiency of the proposed methodology, the designed alarm limits could classify the process to the right state with an accuracy in excess of 90 % according to the result of the tenfold cross-validation, moreover, this efficiency was also proven by the online application of the trained alarm-based fault identification algorithms.

The presented decision tree-based definition of informative alarm thresholds forms my 3.2 thesis finding as will be summerized in Section 8.

# Chapter 4

# Frequent sequence mining for the analysis of industrial processes

Despite the high-pace improvement of industrial process automation, the management of abnormal events still requires human actions. As the key role of an alarm management system is to ensure that only the currently significant alarms are annunciated, these systems are of crucial importance in providing situation-specific information to the decreasing number of operators. The design of alarm suppression rules requires the systematic analysis of the process and its control system. In the present chapter, an overview of the recently developed data-driven techniques is provided, moreover, it is highlighted that the widely applied correlation based methods utilize a static view of the system.

To provide more insight into the process dynamics and represent the temporal relationships among faults, control actions and process variables, a multi-temporal sequence mining algorithm is proposed. The methodology starts with the generation of frequent multi-temporal patterns of alarm signals, which are transformed into Bayes classifiers to form alarm suppression rules. To illustrate how the multi-temporal sequences are applicable for the description of operation patterns, the dataset of a laboratory-scale water treatment testbed is analyzed. For the demonstration of how alarm suppression rules are to be formed, the benchmark simulator of a vinyl acetate production process is extended providing the basis for easily reproducible results and stimulating the development of alarm management algorithms. The results of detailed sensitivity analyses confirm the benefits of the application of temporal alarm suppression rules.

## 4.1 Introduction

Optimizing economic performance within environment and safety related constraints became the primary challenge of control systems [88]. Alarm management systems aim to minimize physical and economic loss through operator interventions in response to an abnormal situation [89]. According to the Engineering Equipment and Materials Users' Association (EEMUA) [3] the purpose of an alarm system is to redirect the operator's attention towards plant conditions requiring timely assessment or action. Therefore, properly designed and operated alarms help the operator to keep the processes in the normal operation range by indicating the presence of abnormal situations. Accordingly, alarm management means the efficient design, implementation, operation, and maintenance of industrial process alarms.

According to the guidelines of EEMUA, 300 alarms per day (one alarm in every five minutes) are manageable for an operator. In most of the European refineries, the number of alarms significantly exceeds this recommended level [34] as it was previously discussed in Chapter 2. The typical alarm management performance metrics presented in Table 4.1 also confirm that industrial metrics are far away from the specified goals [9].

Table 4.1: Typical alarm performance metrics and targets according to [9]

| Metric | Target | Action limit |
|---|---|---|
| Average alarm rate per operator (alarms per day) | <288 | >432 |
| Average alarm rate per operator (alarms per hour) | <12 | >18 |
| Average alarm rate per operator (alarms per 10 minutes) | 1-2 | >3 |
| Percent of 10-minute periods containing >10 alarms | <1% | >5% |
| Maximum number of alarms in a 10 minute period | ≤10 | >10 |
| Percent contribution of top 10 most frequent alarms | <1% to ∼5 % | >20% |
| Percent of time the system is in flood | <1% | >5% |

There can be several reasons for poor alarm management performances. The main driving force is that the definition of alarm thresholds has technically negligible cost in most of the process control systems, resulting in the configuration of far more alarms with often meaningless alarm levels for alarm variables that should not be alarmed at all. Therefore, a high number of nuisance alarms occur that do not require any corrective actions from the operators, as no real abnormalities appeared. This can cause the so-called *cry wolf effect* [49], when the alarms lose their awareness-raising effect and operators are more likely to miss the relevant alarms that are obscured by irrelevant ones. Bliss *et al.* substantiated the existence of the cry-wolf effect for alarm responses [90], highlighting the problem of informativeness of alarm systems. Sorkin and Woods showed that the system performance can be highly improved in light of the operators' workload characteristics [91]. Meyer *et al.* have performed an experimental study and suggested that the occurrence of hazard warnings should depend on the operator's character and that the diagnostic value of a warning system decreases for better operators [92]. It is interesting to note that the issue of informativeness and alarm response has been also discussed in healthcare applications [93].

### 4.1.1 The methodologies of alarm design

As it was discussed in Chapter 3, the design of alarm thresholds is the top priority to reduce the number of false or missed alarms (alarms without the presence of abnormal situation or alarms, which do not appear in case of abnormality, respectively). However, the full functionality of alarm management systems is not guaranteed by relying purely on the proper definition of alarm thresholds as there are numerous process variables that cannot be associated with a static alarm configuration.

In some cases, alarm limits should be changed according to the appropriate operating mode, therefore, a mode based alarm system [94] or dynamic alarm management [95] might be needed. Other main contributors of high alarm numbers are the *chattering alarms*. According to ANSI/ISA-18.2 (2009), any alarm occurring more than three times in a one-minute-period can be considered as a chattering alarm. In a more informal context, any alarm that appears with a disturbingly high frequency can be regarded as a chattering alarm [13]. To avoid chattering, deadbands [60, 96], delay-timers [61], or filtering [97] can be used. A summary of methods for the improvement of alarm systems is given by Izadi *et al.* [35].

### 4.1.2 Advanced alarm reduction techniques

Although guidelines suggest having only one alarm for a single abnormal event, the number of interacting components makes it almost infeasible to avoid redundant alarms. The main idea of the approach discussed in the present chapter is the grouping of overlapping and redundant alarms and, thanks to their increased informativeness, the application of these groups for fault detection, event prediction and alarm suppression (see Figure 4.1) [89]. In the following, a brief overview of the methods developed in support of this concept is provided.

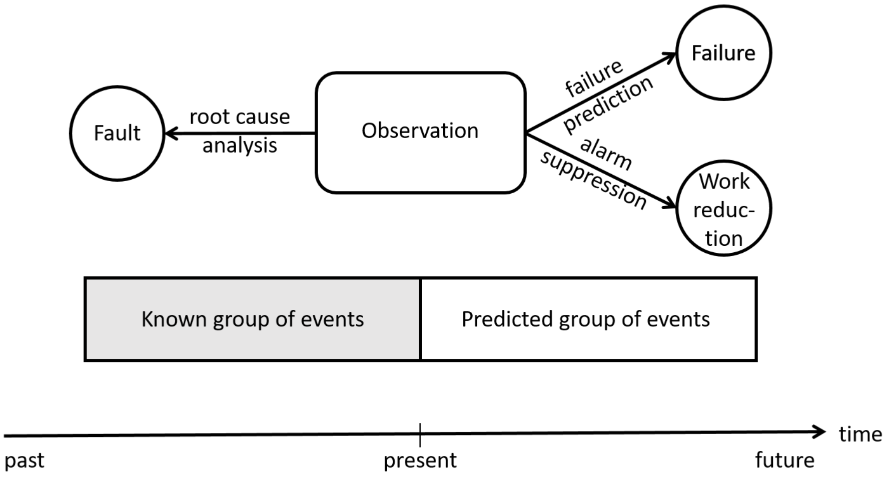


Figure 4.1: Grouping of events can be useful in fault detection, root cause analysis and alarm suppression

#### Correlation analysis based techniques

The main idea of correlation analysis is the exploration of alarms frequently co-occurring within a short period of time [80]. The results of this analysis can be used for the reduction of operator workload as has been reported in a Japanese ethylene plant [98]. Kondaveeti *et al.* used binary and Gaussian kernel based representations for the visualization of clustered alarms [99, 20]. According to Yang *et al.* [100], among the 22 similarity measures tested for the analysis of binary represented alarm signals, the Sorgenfrei coefficient found to be the most effective and the correlation delays were also found to be indispensable for the detection of correlated alarm tags. As the estimated Sorgenfrei and Jaccard coefficients were found to be too small and the distribution of correlation delays required a large database, the proposed approach was further improved by Hu *et al.* [101]. Wang *et al.* [102] combined the similarity analysis and the detection of causal relationships for the identification of consequential alarms and their evolution path. Grouping of massive alarm data in distributed parallel alarm management systems also utilizes alarm similarity measures [103]. More recently, alarm prioritization and clustering methods (correlation analysis, cluster analysis, and principal component analysis) were applied for the reduction of the alarm rate of a natural gas processing plant by Soares *et al.* [104].

### Pattern analysis based techniques

The key idea of pattern analysis based techniques is that advanced data mining algorithms can extract useful patterns that can be used to formalize alarm suppression rules. A context-based segmentation approach was applied to find a pattern of correlated alarms by Kordic *et al.* [105]. The proposed method was further improved to avoid the disturbing effect of alarms triggering each other [106]. As the triggering alarm may return before the end of the pattern sequence, a major limitation of the method is the requirement of a designated target tag in advance to set the starting point of the segmentation. Folmer and Vogel-Heuser proposed an automatic alarm sequence generating algorithm to find the possibilities to redesign the alarm management systems. However, the authors admit the deficiency of the algorithm, as it is not able to compare different alarm sequences even if they differ in only a single alarm in the sequence [107]. This problem can be solved by a modified Smith-Waterman algorithm which utilizes the time stamp information of alarm tags to calculate the similarity index of alarm floods [108]. The algorithm uses these similarity indexes to classify the alarm floods to similar patterns instead of identifying exactly the same sequences. This work was extended with new scoring functions, dynamic programming, backtracking and alignment generation procedures in order to align multiple alarm flood sequences [109]. At the application of dynamic time warping for the pairwise alignment of the sequences, it can be also assumed that the alarm sequences are generated from first-order Markov chains [110]. Recent works also focus on the wider scope and general applicability of the alarm management concepts. A novel framework together with the developed toolbox for smart data analytics of alarm, process data and causality analysis were published by Hu *et al.* [111], while an online pattern matching approach for reduction of incoming alarm floods was reported by Lai *et al.* [68].

### 4.1.3 Motivation and outline of the chapter

The previously presented detailed overview highlighted that the current state of alarm management is static and lacks the detailed probability analysis of extracted suppression rules. In the present chapter we would like to introduce a methodology resolving these issues.

To provide more insight into the process dynamics, pattern-based techniques are widely applied in the literature. Taking a step further, a method to handle the temporal relationships among the alarms, control actions and process variables is proposed.

The main contributions discussed in the present chapter are:

- The mining of frequent temporal sequences requires a special algorithm. I extend the previous work of [112] with the probabilistic interpretation-based design of alarm suppression rules.
- The multi-temporal sequences are transformed into Bayes classifiers that can be directly used to define alarm suppression rules.
- The real-life applicability is demonstrated via the analysis of an openly available dataset obtained from a laboratory-scale water treatment testbed.
- The benchmark simulator of a vinyl acetate production technology is extended to generate easily reproducible results and further stimulate the development of alarm management algorithms.
- A detailed sensitivity analysis to demonstrate the effects of the parameters of the sequence mining algorithm to the number of the identified alarm suppression rules is provided.

The roadmap of the chapter is as follows. The sequence mining problem, the suggested reliability measures, and the pattern mining algorithm are presented in Sections 4.2.1, 4.2.2 and 4.2.3, respectively. The workflow of the proposed methodology is summarized in Section 4.2.4. The real life applicability and effectiveness of the proposed sequence mining algorithm are demonstrated through the description of the process dynamics of a scaled-down version of an industrial water treatment process (Section 4.3.1). The detailed analysis of the alarm suppression methodology is based on simulated data. The studied vinyl acetate production process, its simulator and its extension to make it suitable for the study of alarm management problems are described in the Appendix of the Thesis in Sections 9.1 and 9.2, respectively. The structure of the generated temporal database applied in this analysis is described in Section 4.3.3. The causal connections between alarms are illustrated in Section 4.3.4. As problematic faults have similar consequences, the statistical measures, the complexity and similarity analysis of the faults are evaluated in Section 4.3.5, followed by the sensitivity analysis-based illustration of the applicability of the extracted alarm suppression rules in Section 4.3.6. Based on the results, the applicability and the limitations of the methodology is discussed in Section 4.3.7, which is followed by some concluding remarks (Section 4.4).

## 4.2 Methodology

### 4.2.1 Multi-temporal representation of alarm sequences

Alarm and warning signals can be treated as the *states* of the technology. Each state (denoted by $s$) is represented by $< pv, a >$ data couples, where $pv$ is the index of the process variable and $a$ is the attribute showing the value of the process variable related to the alarm and warning limits, such as $a \epsilon \{Low\ A, Low\ W, High\ W, High\ A\}$, where $A$ stands for Alarm and $W$ stands for Warning. For example the description of a state can be represented as follows: $s_e\ := < Column\ Top\ Temperature, High\ A >$.

An *event* is the time interval in which the defined state occurs, denoted by $e$. Therefore an event can be represented by a triplet, such as $< s,\ st,\ et >$, where $s$ is the state, which is taking place in a time interval between $st$ start time and $et$ end time. Note that each state can correspond to one or more events as well. An event is related to a state $s_e$ as follows: $< s_e,\ 12,\ 14 >$.

As the prerequisite of the application of the discussed methodology is the existence of an event log database, in case it is not available, it is necessary to create one by converting the time series data to an event log. In the present context, we use the limits of the individual variables to define events. The definition of such thresholds is not trivial, in most of the cases, process-relevant knowledge and sensitivity tests are used, but for the conventional approaches see Section 4.1.1 or refer to Chapter 3 for a wider scope. After the definition of alarm and warning thresholds, the periods of faulty operations can be defined by checking whether the process variables exceed their associated limits. Therefore in Figure 4.2, two events can be defined, $e_1 \;=\; < Temperature_{LowA},\; st_{e1},\; et_{e1} >$ and $e_2 \;=\; < Temperature_{HighA},\; st_{e2},\; et_{e2} >$, where $Temperature_{LowA}$ and $Temperature_{HighA}$ are the states of the specified object.

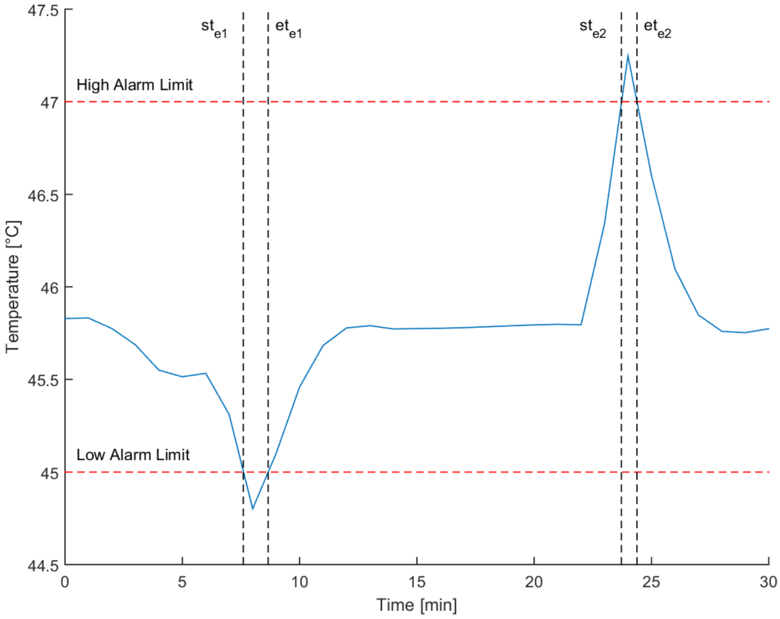


FIGURE 4.2: Example for the time series of a process variable with the lower and higher thresholds indicated

With the obtained event start and end points, an event log database is to be formed by sorting these events into ascending order based on their start times. The $D_T$ database in Table 4.2 shows an example for an event log database and the events are visualised in Figure 4.3.

For the determination of event correlation it is important to define a *time window*. In the case of a $(st_1, et_1)$ time interval of event $e_1$, the time lag (window) defined as $C_1 = (st_1, et_1{+}window)$ is the window constraint of $e_1$. If we have an event $e_2$ with the starting point $st_2$, such that $st_2 \,\epsilon\, C_1$, than $e_2$ satisfies the window constraint of $e_1$. The value of the window parameter sets the time distance in which two event is considered correlated. Given the $D_T$ database with a time window of 4, then $e_6$ satisfies the window constraint of $e_4$ and a temporal co-occurrence or some kind of correlation can be assumed. Given two events, $e_1$, $e_2$, with the time intervals of

TABLE 4.2: The exemplary $D_T$ event log database

| Event id | State id | Starting time | Ending time |
|---|---|---|---|
| $e_1$ | $s_1$ | 1 | 4 |
| $e_2$ | $s_2$ | 2 | 6 |
| $e_3$ | $s_3$ | 2 | 6 |
| $e_4$ | $s_1$ | 5 | 8 |
| $e_5$ | $s_4$ | 5 | 15 |
| $e_6$ | $s_2$ | 10 | 15 |
| $e_7$ | $s_5$ | 11 | 14 |
| $e_8$ | $s_2$ | 16 | 18 |

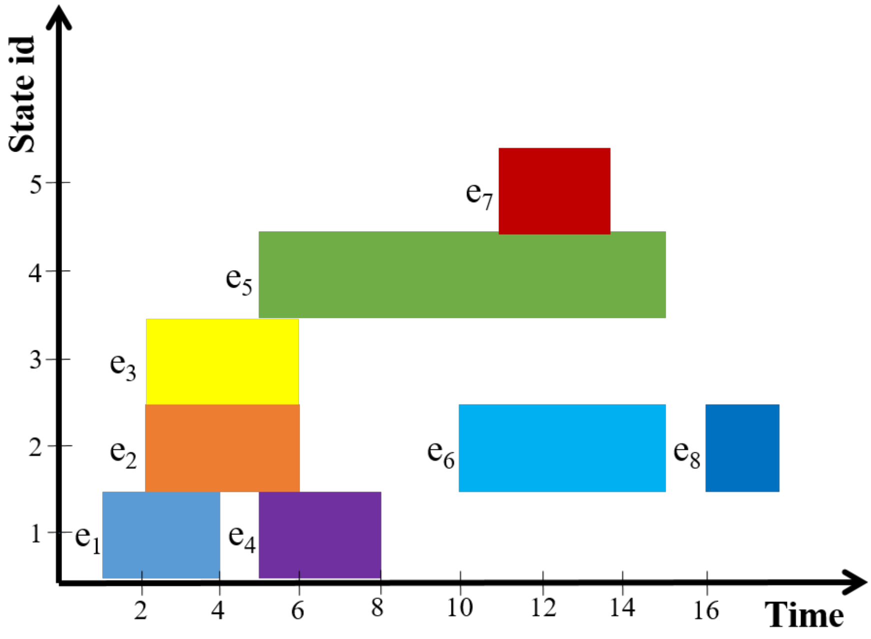


FIGURE 4.3: A visual representation of the events of different (process) states in time. The horizontal axis represents the time and the rows of the vertical axis represent the states. The vertical length of the bars with different colours indicates the length in time of the given event.

$(st_1, et_1)$ and $(st_2, et_2)$, respectively, a temporal connection can be defined between them, if the events are connected by one of the following four temporal predicates:

- If $st_1 = st_2$ and $et_1 = et_2$, then $e_1$ *equal* $e_2$,
- If $0 \leq st_2 - et_1 \leq$ *window*, then $e_1$ *before* $e_2$
- If $st_2 < st_1 < et_1 \leq et_2$, then $e_1$ *during* $e_2$
- If $st_1 \leq st_2 < et_1 < et_2$ or $st_1 < st_2 < et_1 \leq et_2$, then $e_1$ *overlap* $e_2$

Hereinafter, the notations $E$, $B$, $D$ and $O$ are used for $equal$, $before$, $during$ and $overlap$, respectively.

Using this approach, temporal instances as $\phi \ := \ e_1 \overset{R}{\Rightarrow} \ e_2$ are defined, where $R \ \epsilon \ \{ \ E, \ B, \ D, \ O\}$ is a temporal predicate. For example, in the case of the $D_T$ temporal database: $e_1 \overset{O}{\Rightarrow} e_2$, $e_2 \overset{E}{\Rightarrow} e_3$, $e_7 \overset{D}{\Rightarrow} e_6$, $e_6 \overset{B}{\Rightarrow} e_8$. The set of $n$ states is denoted by $S = (s_1, s_2, ..., s_n)$ and the set of temporal predicates is marked as $\{E, \ B, \ D, \ O\}$.

A pattern of $k + 1$ states is connected by $k$ temporal predicates and called a $k$-length temporal pattern (the number of predicates is $k$), or a $k+1$-state temporal pattern (the number of states is $k + 1$) and is denoted as $\Phi_k$. Consequently, the following patterns are defined:

- $k$ = 0-length patterns, where $\Phi_0 := s_0, \ s_0 \ \epsilon \ S$ (a trivial pattern with only one sate, e.g. "low temperature", is called a degenerated temporal pattern)
- $k$ = 1-length pattern is formulated as $\Phi_1 := (\Phi_0 \overset{R_1}{\Rightarrow} s_1) := (s_0 \overset{R_1}{\Rightarrow} s_1)$, $R_1 \ \epsilon \ \{E, \ B, \ D, \ O\}$
- In general, a $k \geq 2$-length pattern is formulated as $\Phi_k := (\Phi_{k-1} \overset{R_k}{\Rightarrow} s_k) := (s_0 \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} s_2 \overset{R_3}{\Rightarrow} ... \overset{R_k}{\Rightarrow} s_k)$, $R_j \ \epsilon \ \{E, \ B, \ D, \ O\}$, where $j = 1, 2, ..., k$.

If a temporal pattern e.g. $\Phi := s_i \overset{R}{\Rightarrow} s_j$ and a temporal instance $\phi := e_{ip} \overset{R}{\Rightarrow} e_{jq}$ are given where $e_{ip}$ and $e_{jq}$ are the events related to $s_i$ and $s_j$, respectively, then we call the temporal pattern $\Phi$ supported by the temporal instance $\phi$.

The patterns with 3 or more states (or 2 predicates) are called multi-temporal patterns. Based on this, given a multi-temporal pattern $\Phi_k := (\Phi_{k-1} \overset{R_k}{\Rightarrow} s_k) := (s_0 \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} s_2 \overset{R_3}{\Rightarrow} ... \overset{R_k}{\Rightarrow} s_k)$, $R_j \ \epsilon \ \{E, \ B, \ D, \ O\}$, $j = 1, 2, ..., k$, then a pattern $\Phi' := s_{j1} \overset{R_{j2}}{\Rightarrow} s_{j2} \overset{R_{j3}}{\Rightarrow} s_{j3} ... \overset{R_{jm}}{\Rightarrow} s_{jm}$, where $0 \leq j_1 \leq j_2 \leq j_3 \leq ... \leq j_m \leq k$, and $j_1, j_2, ..., j_m$ is a series of sequential natural numbers, then $\Phi'$ is called a sequential sub-pattern of $\Phi$. In the $D_T$ temporal database, the $\Phi_1 := s_1 \overset{O}{\Rightarrow} s_2$ is a sequential sub-pattern of the $\Phi_2 := s_1 \overset{O}{\Rightarrow} s_2 \overset{B}{\Rightarrow} s_5$ pattern (however, it is supported by only one temporal instance).

### 4.2.2 Probabilistic interpretation of temporal patterns

The estimation of the number of annunciated alarms and the risk associated with the suppression of a warning signal requires the probability based interpretation of the temporal patterns. The key idea is that the probability of an alarm is proportional to the support of the $k = 0$-length sequence as can be seen in Equation 4.1, where $supp(s_i)$ denotes the number of supporting events of each $s_i$ state.

$$P(s_i) \backsimeq supp(s_i) \tag{4.1}$$

The number of the supporting events of a state is easily determined by counting the number of instances under each state in a transformed database, as presented in the $D'_T$ transformed temporal database in Table 4.3 (the transformed form of the $D_T$ temporal database from Table 4.2). The columns of this converted database are the individual states, while the rows below each state represent the time intervals when that the state occurs.

Table 4.3: The $D_T$ example event log database transformed into $D'_T$. The columns of this converted database are the individual states, while the rows below each state are the time intervals in that the state occurs.

| $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ |
|---|---|---|---|---|
| $[1,4]$ | $[2,6]$ | $[2,6]$ | $[5,15]$ | $[11,14]$ |
| $[5,8]$ | $[10,15]$ | | | |
| | $[16,18]$ | | | |

In the present approach, to characterize the frequency of the given event, the number of temporal instances is normalized by the number of temporal instances of the most frequent state according to Equation 4.2.

$$support(\Phi) = supp(\Phi)/|E| \tag{4.2}$$

where $|E| = \max\limits_{j=1,2,\ldots,N}(|E_j|)$, $N$ is the number of states in the $D_T$ temporal database, $E_j$ is the set of events supporting state $s_j$, $|E_j|$ is the cardinality of $E_j$ (the number of events in $E_j$). According to Table 4.3, the degenerated pattern, $s_1$, is supported by two temporal instances, therefore $supp(s_1) = 2$, and $|E| = max(2,3,1,1,1) = 3$, resulting in $support(s_1) = 2/3$. Therefore, the support value is a measure between 0 and 1 and is used to measure the frequency of each pattern, an essential characteristic measure of frequent patterns. Assuming a defined support threshold ($minSupp$), determined as an input parameter of the mining algorithm, the $\Phi$ sequential pattern is called frequent pattern if $supp(\Phi) \geq minSupp$. The main goal of sequential pattern mining is to find all the frequent patterns for a specific $minSupp$.

A consequence of this is that the $k > 0$-length patterns are proportional as well to the support of the degenerated pattern ($k = 0$-length sequences). The probability of a transition between two states is consequent to the support value of the appropriate $k = 1$-length sequence.

Alarm suppression rules can be defined when the temporal instances of the states are not independent, so the probability of a $P(\Phi_k) := P(s_0 \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} s_2 \overset{R_3}{\Rightarrow} ... \overset{R_k}{\Rightarrow} s_k)$ sequence can be calculated by the chain rule:

$$\begin{aligned} P(\Phi_k) = P(s_1|s_0) \times P(s_2|s_0 \overset{R_1}{\Rightarrow} s_1) \times \dots \\ \dots \times P(s_k|s_0 \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} s_2 \overset{R_3}{\Rightarrow} ... \overset{R_{k-1}}{\Rightarrow} s_{k-1}) \end{aligned} \tag{4.3}$$

Equation 4.3 highlights that the probability of the occurrence of a $k$-length sequence is calculated from the probability of occurrence of the $k-1$-length sequence ($\Phi_{k-1}$), and the conditional probability of the occurrence of the state $s_k$ after the $\Phi_{k-1}$ sequence, denoted by $P(s_k|\Phi_{k-1})$, where $\Phi_{k-1}$ is the $k-1$-length sub-pattern of $\Phi_k$ (therefore, based on the definition, $\Phi$ can be also written as $\Phi_{k-1} \overset{R_{k-1}}{\Rightarrow} s_k$):

$$P(s_k|\Phi_{k-1}) = \frac{P(\Phi_k)}{P(\Phi_{k-1})} = \frac{supp(\Phi_k)}{supp(\Phi_{k-1})} \tag{4.4}$$

As a result, a measure of probability, the confidence of the sequence $\Phi_k$ is obtained ($conf(\Phi_k)$), describing the confidence of the transition rule defined between the states:

$$conf(\Phi_k) = \begin{cases} \frac{supp(\Phi_k)}{supp(\Phi_{k-1})} \times conf(\Phi_{k-1}) & R_k \neq D \\ \frac{supp(\Phi_k)}{supp(s_k)} & R_k = D \\ 1 & \Phi_{k=0} \end{cases} \tag{4.5}$$

Both the support and the confidence measures are scaled between 0 and 1 and used for the characterization of frequent sequences. The support gives a measure of the relevance of the given sequence by characterising how frequent it is, while the confidence measures its reliability. As the confidence measure is calculated as the product of conditional probability values, its result can be interpreted similarly to a conditional probability value: if a sequence has started, what is the probability of the occurrence of a specific chain of events. Therefore, if a state is not frequent by itself, but after its occurrence, a well-defined sequence follows it, then the support of the sequence will be low while its confidence is high. It is important to mention, that $conf(\Phi_k)$ therefore gives the confidence of the whole $\Phi_k$ sequence, while the confidence of the transition between states is a simple conditional probability value denoted as $conf(\Phi_{k-1} \overset{R_k}{\Rightarrow} s_k) = P(s_k|\Phi_{k-1})$ as described before, thus the two quantity is not equal.

Assuming that information on the type of fault is available as well, then the sequences of events can be defined as the consequence of a $c_j$ type of fault. Therefore the probability of this event sequence can be defined by the $P(c_j \overset{R_0}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} ... \overset{R_k}{\Rightarrow} s_k)$ probability. According to the philosophy of the alarm suppression rules, the occurrence of a failure and the sequence of the occurring events are not independent. Therefore, the probability of the whole sequence is described by the chain rule:

$$\begin{aligned} P(c_j \overset{R_0}{\Rightarrow} \Phi_k) = P(c_j) \times P(s_1|c_j) \times P(s_2|c_j \overset{R_1}{\Rightarrow} s_1) \times ... \\ ... \times P(s_k|c_j \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} ... \overset{R_{k-1}}{\Rightarrow} s_{k-1}) \end{aligned} \tag{4.6}$$

Equation 4.6 statistically handles the case when multiple $c_j$ faults are followed by similar alarm sequences or when a single fault can be followed by multiple alarm sequences as well. In the case of similar alarm sequences with different root causes, however, the probability metrics of the different root causes can be similar, the slight differences can still indicate the presence of the one with the higher probability. On the other hand, if different alarm sequences can follow a malfunction, these alarm sequences are similarly characterized by their probability of occurrence.

It is also measured how dependent a consequent part of a sequence on the antecedent part. To evaluate this dependence, a measure called the *improvement* is defined, calculated as the fraction of the confidence of the transition between states and the support of the added state:

$$\begin{aligned} Improvement(\Phi_k) &= \frac{P(s_k|c_j \overset{R_0}{\Rightarrow} \Phi_{k-1})}{P(s_k|c_j)} = \\ &= \frac{conf((c_j \overset{R_0}{\Rightarrow} \Phi_{k-1}) \overset{R_k}{\Rightarrow} s_k)}{support(s_k)} = \\ &= \frac{support(c_j \overset{R_0}{\Rightarrow} \Phi_{k-1} \overset{R_k}{\Rightarrow} s_k)}{support(c_j \overset{R_0}{\Rightarrow} \Phi_{k-1}) \times support(s_k)} \end{aligned} \tag{4.7}$$

An improvement value less than 1 means that $s_k$ occurs independently of the $c_j \overset{R_0}{\Rightarrow} \Phi_{k-1}$ sequence and the rule is not really characteristic or interesting.

### 4.2.3 The multi-temporal mining algorithm

To generate informative patterns from temporal databases, the algorithm published in [112] is utilized, with a fine-tuned implementation to make it more convenient and applicable for alarm management purposes. In the original implementation, the generated patterns are only filtered based on a minimal support (*minSupp*) value. Since we are interested in patterns that are useful as alarm suppression rules, the patterns generated by the algorithm are also filtered based on a minimal confidence value ($minConf$). The pseudo-code of the described algorithm can be seen in Algorithm 1.

The $k+1$-length patterns are generated step by step by combining all the $k$-length patterns (degenerated patterns at the beginning of the mining algorithm) and the events of the $D_T$ temporal database, while checking their relationship to the defined time window. The appearing sequences are stored until no more $k+1$-length sequences can be created. Then, the sequences are examined against the specified *minSupp* and $minConf$ conditions, as only the sequences meeting these criteria are defined as frequent and reliable patterns. Patterns violating these conditions are discarded. This process is repeated until no more sequences are created with $k+1$ element.

**Algorithm 1 Multi-Temporal Mining** method

```
Require: $D_T$: Temporal database
    $minSupp$: Support threshold
    $minConf$: Confidence threshold
$F_0 = 0$
for every state $s$ in $D_T$ do
    if $supp(s) \geq minSupp$ then
        $F_0 = F_0 \cup \{s\}$
    end if
end for
$C_1 = \{s_i \overset{R}{\Rightarrow} s_j | s_i \in D_T, s_j \in F_0\}$
i=1
while $F_{i-1} \neq 0$ do
    $F_i = 0$
    for every sequence $\Phi \in C_i$ do
        if $supp(\Phi) \geq minSupp$ and
        $conf(\Phi) \geq minConf$ then
            $F_i = F_i \cup \{\Phi\}$
        end if
    end for
    $C_{i+1} = \{s_i \overset{R}{\Rightarrow} s_j | s_i \in F_i, s_j \in F_0\}$
    i=i+1
end while=0
```

### 4.2.4 Workflow of the methodology

The workflow of the methodology is depicted in Figure 4.4. First, the examined events are collected or generated in the form of process log files of alarms or the discrete events. For the construction of these events, time series data can be utilized as well. There are cases when the log files of previous operation periods are not provided, however, nowadays many of the industrial plants have a detailed dynamic simulator either for modeling or operator training purposes. In the case of the analysis of faults that formerly have not happened or in the lack of process log files, malfunctions in these simulators can be implemented and the events occurring after them can be recorded.

With the use of this database, the described multi-temporal mining algorithm is applied for the mining of the frequent patterns of events. Here, two options arise. If the root cause of the events is known, therefore the database contains the malfunctions as well, the faults are characterized by their following sequences and the spillover effect of the malfunctions can be investigated. On the other hand, if the root cause of the events is unknown, as in the case of general industrial alarm log files, the frequently co-occurring alarm pairs and longer patterns can still be analyzed through their statistical metrics. Moreover, in these cases the alarm suppression rules can be derived as well, defining of a confidence threshold as the requirement for the correctness of the prediction.

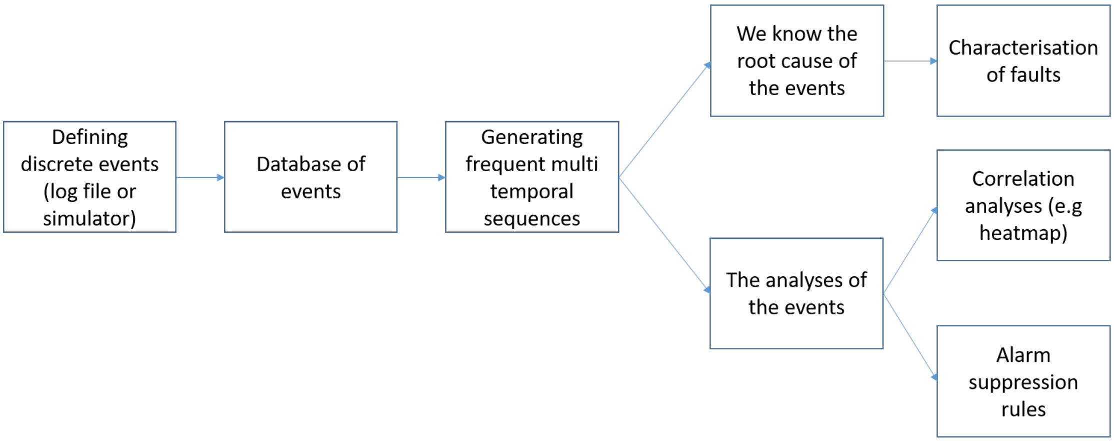

Defining discrete events (log file or simulator)
Database of events
Generating frequent multi temporal sequences
We know the root cause of the events
Characterisation of faults
The analyses of the events
Correlation analyses (e.g heatmap)
Alarm suppression rules

FIGURE 4.4: The schematic workflow of the proposed methodology.

## 4.3 Results and discussion

### 4.3.1 The multi-temporal representation of a dynamic process

The proposed multi-temporal sequences are a reduced-information-content representation of the process dynamics compared to the sensor measurements. This representation is highly useful if a deeper understanding of the process dynamics is aimed and only alarm & event log databases are available of the longer process history.

In the present section, the effectiveness of this representation is illustrated using an openly available dataset, the Secure Water Treatment testbed (SWaT) dataset, made by the iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design [113]. The testbed, the scaled-down version of an industrial water treatment plant, was designed to support the research of cyber-physical systems. The data collection process lasted for 11 days. In the first seven days, the testbed operated without malfunctions, so this part of the dataset is available for the determination of normal operating patterns. In the remaining four days, the SWaT was under attack, this part of the dataset is considered as the faulty operation. In these attacks, mainly the functioning of different parts of the control system is deteriorated: a manipulated variable is altered (*e.g.*, a valve is opened or closed), or a process variable or setpoint is changed to constantly decrease or increase something (*e.g.*, a level variable). In total, 946,722 records comprising of 51 attributes were collected over the 11 days of the data collection process including signals from sensors (tank levels, flow meter's, water properties, pressure) and actuators (water pumps, motorized valves and ultraviolet dechlorinators).

The applicability of the multi-temporal sequences is demonstrated on a simple example related to the raw water storage tank located at the first part of the testbed. The tank has two sensors measuring the inflow into the tank and the water level inside the tank, as well as two binary control inputs switching on or off the inflow and the outflow. In case of the measured sensor variables, events are recorded when the associated variable drops below the lower or increases above the higher threshold of the variable. The thresholds are the margins of the lower and upper 5 % of operation range. In case of the actuators, the events were defined based on the binary input control signals. Figure 4.5 shows a normal operation period with the indication of the occurring discrete events and their time intervals.

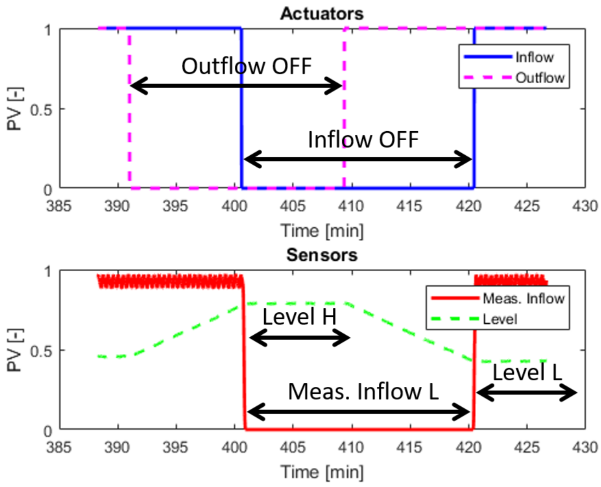


FIGURE 4.5: The actuator (inflow and outflow) and sensor (measured inflow and tank level) values of the raw water storage tank of Secure Water Treatment testbed provided by iTrust. The time intervals of the recorded discrete events are indicated by black arrows with the name of the recorded event above it (L for low and H for high): Outflow off, Inflow off, Level High, Measured Inflow Low, Level Low. The horizontal axis shows the time, while the vertical axis shows the scaled value of associated process variable (PV).

The proposed multi-temporal mining algorithm is capable of the determination of the frequently occurring operation patterns, as with values of 0.2 of support and confidence thresholds and 2-minute-long time window, the algorithm finds the pattern of operation: $Level\ High \overset{D}{\Rightarrow} Outflow\ OFF \overset{O}{\Rightarrow} Inflow\ OFF \overset{O}{\Rightarrow} Measured\ Inflow\ Low \overset{B}{\Rightarrow} Level\ Low$. It is important to highlight that regardless of the chosen parameters, the algorithm is capable of the characterization of processes as reasonable patterns are found with other input values as well, however, parameters produced a single sequence that nicely represents the operation of the analyzed tank. The sequences found by other parameters are equally valid for the characterization of the processes by the occurring events, but this approach of seeking the sequence of connected events can facilitate the tuning of the parameters of the frequent sequence mining algorithm. In the case of the sequences of rare events, the frequent sequence mining algorithm can be supported by the goal-oriented filtering of the log file to the periods containing the analyzed rare events and/or by the filtering of the resultant sequences containing the relevant events.

The detected sequence describes well the normal operation pattern of the storage tank. While the outflow from the tank is off, the level will reach the high state (notice that the during temporal predicate starts with the shorter event, which is incorporated into the other event). Thanks to the high liquid level, the outflow from the tank is turned on (this state is the complementer of the turned-off state, therefore it is not defined as an event) and the inflow to the tank is turned off, which resulted in a low measured inflow as well. Finally, we will reach the low-level state.

The application example demonstrates the effectiveness of the multi-temporal sequences in the description of dynamic processes. If the alarm and warning signals of a process are considered as the indicators of the process state, the frequently occurring operational patterns are derived with the use of the proposed methodology and the related algorithm. However, to demonstrate the applicability of the proposed sequence-based representation in the field of alarm suppression, a more detailed simulation example is constructed.

### 4.3.2 Defining and implementing faults into the vinyl acetate process simulator

The description of the applied benchmark vinyl acetate process simulator, together with its modifications is presented in the Appendix in Chapter 9.

For the efficient and precise work with the VAc simulator, the simulator was revised and additional faults were inserted related to the controllers, as the manipulated value of the process variable remained at a constant value for a specified time. The duration of these faults follows a lognormal distribution. The values of the manipulators in the case of faults can be seen in Table 9.1 in the Appendix.

In the case of injected malfunctions an Error Event Matrix (EEM) is created randomly with the following content:

- Fault occurrence time (min)
- Fault end time (min)
- Fault ID for the identification of fault

Each malfunction type is chosen based on a uniform distribution and the duration is determined using the lognormal distribution described before. The occurrence of each fault is generated between the $10th$ and $T_{simend} - 100th$ minute of the simulation (to provide enough time for the process to go back to normal operations). The OSM and EEM record the input data of the randomly generated experiments.

### 4.3.3 The generation of the log files of process alarms

The alarm limits were determined based on a simulation containing 150 different process states (according to the Operator State Matrix, OSM), during these simulations no faults were implemented. The minimal and maximal values of each process variable recorded in the simulation were used as the alarm limits during faulty operations.

First of all, with the utilization of the VAc simulator, time series were created for each fault in separate runs. All of the faults were implemented in 200 different, 1-hour-long operating states with lognormal time distribution in length as it was described before. The time series were then transformed to an event-log database using the defined alarm thresholds. If the improvement of HAZOP analysis and the characterization of the consequences of faults are targeted, than the known faults are also included as events in the log file. This log file forms the input database of the multi-temporal mining algorithm. The obtained sequences were filtered, as we were interested only in the sequences containing the implemented faults at the beginning of the characteristic sequences. If the reduction of incoming alarms is targeted, the faults are not included in the event database, as in this case only in the frequent operating patterns are to be determined, and we are not interested in their root cause. In this case, the frequent event sequences are generated from this database.

The effect of the size of the window parameter on the number of the generated sequences was tested, as this reflects the temporal causality of the implemented faults. The results were calculated with the use of the database containing the faults as well, using 0.1 and 0.2 of support and confidence values, respectively. Figure 4.6 illustrates that the number of generated sequences show a saturation as the function of the time window. This is reasonable as in optimal cases, the effect of a fault decays over time after the end of the fault.

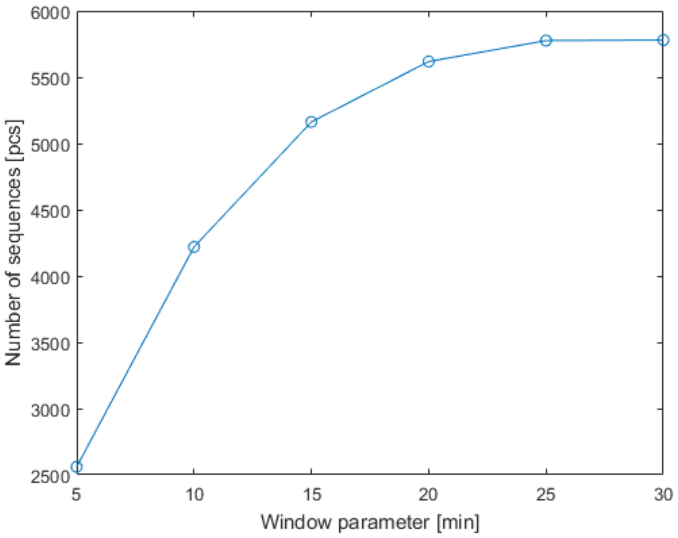


FIGURE 4.6: The number of frequent sequences shows a saturation with increasing time window

The number of generated sequences highly depends on the chosen support and confidence threshold values as well. The horizontal axis of Figure 4.7 shows the value of the investigated threshold, the other threshold is set to 0.2 during the analysis and the window parameter is set to 20 minutes. The first vertical axis on the left shows the number of found sequences during the modification of the support threshold, while the second vertical axis on the right shows the number of generated sequences during analysis of the confidence threshold. According to Figure 4.7 the number of frequent sequences decreases significantly as the function of the increasing threshold values.

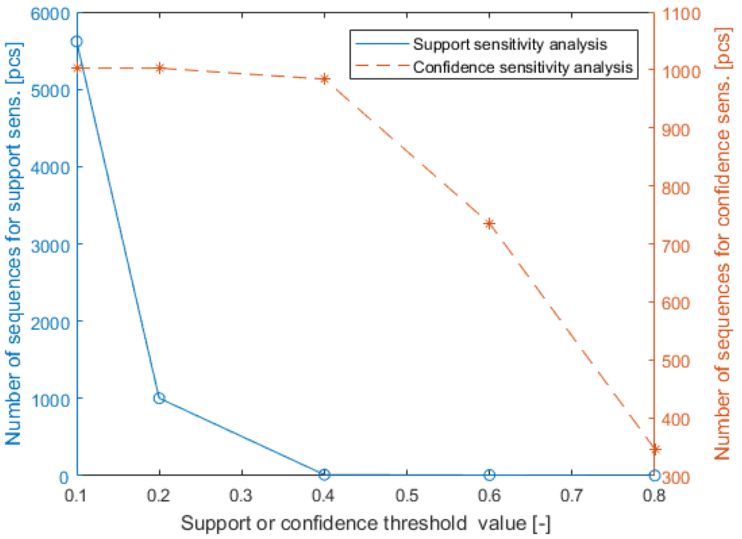


FIGURE 4.7: The number of generated sequences in case of different support and sensitivity threshold values. The investigated threshold value is indicated on the horizontal axis, the other threshold is set to a value of 0.2. The first vertical axis on the left shows the number of sequences for the analysis of the support threshold, while the second vertical axis on the right shows the results of the confidence threshold analysis.

It is important to mention that the confidence of transition between states is the most critical measure that determines the applicability of an alarm suppression rule. Taking into consideration the results of sensitivity analysis, 20-minute time window, 0.1 and 0.2 support and confidence threshold values were used, respectively.

### 4.3.4 Correlation of occurring alarms

Advanced alarm management algorithms are often based on correlated alarms, as correlation analysis is capable of exploring the hidden connections between alarms. This causal connection is illustrated by the heatmap in Figure 4.8, visualizing the intensity of the alarm tags occurring together in the defined order. Figure 4.8

shows the support value of the $k = 2$-length sequences. The support values are calculated ignoring the temporal relationships, therefore, the support values of the same alarm pairs connected with different temporal predicates are summarized. The database not containing the faults was used for the analysis.

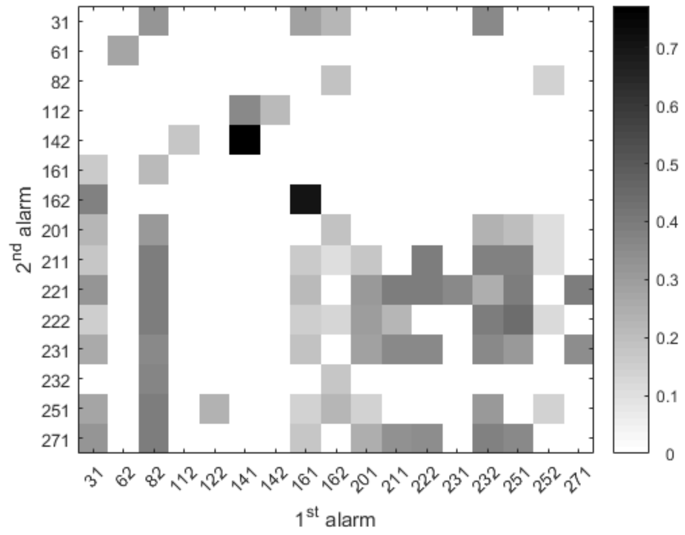


FIGURE 4.8: The heatmap shows the support values for the $k = 2$-length sequences. The support values are calculated ignoring the temporal relationships, therefore, the support values of the same alarm pairs connected with different temporal predicates are summarised. Axis x shows the first, while axis y shows the second alarm tag of the sequence.

Figure 4.8 highlights that some alarms can be followed by multiple alarm tags. Due to this uncertainty, the prediction of the next occurring alarm messages is not possible based on just the knowledge of the previous alarm message. The analysis of longer sequences is needed, as the knowledge of the history of states and the transition between them can increase the probability of the accurate prediction of the next event (see the improvement measure in Equation 4.7).

### 4.3.5 The characterization of faults based on their complexity

The sequence database holds an enormous amount of information on the dynamics of the process. Based on the stored information, more problematic or complicated faults can be revealed, which are hard to identify based on their particular sequences, or faults that cause a spillover effect on the process can be identified.

This can support the improvement of safety-related analysis methods, e.g., hazard and operability analysis. Creating such studies is usually highly labor-intensive, as a multidisciplinary team creates scenarios for possible malfunctions in brainstorming meetings. The team tries to identify all the potential process faults with their consequences, developing recommendations and action-scenarios for handling them each by each. However, this technique requires long hours of human work and relies entirely on the expert knowledge of the team members. With the use of the proposed methodology, the time and workforce requirement of such techniques can be reduced.

To analyze the consequences of specific faults, the sequences of the event database containing the faults as well were filtered to the ones containing only one fault at the beginning of the sequence. The statistical evaluation of the obtained sequences was carried out as can be seen in Table 4.4 (the No. of faults follows the order presented in Table 9.1). The higher number of characteristic sequences reflects that it is harder to identify the root cause of the sequence of events, while longer sequences imply that the given fault causes a spillover effect on the process triggering a series of alarms. The support and confidence values show the measure of how relevant (frequent) and reliable the obtained information is, respectively.

According to Table 4.4, most of the faults have only a few types of frequent alarm patterns following them. The only exception is Fault 3, the fault of the separator liquid exit flow rate.

Faults with fewer types and shorter length of characteristic sequences are easier to identify, and the information held by these sequences are usually treated as trivial by the process experts. Moreover, using expert knowledge, faults with longer characteristic sequences are much harder to interpret and predict their consequence. The characterization of faults based on the number of characteristic sequences following them and maximal length of these sequences is visualized in Figure 4.9. The number of sequences following the given malfunction is illustrated in the horizontal axis, while the maximal length of these sequences is indicated on the vertical axis. Faults not generating any frequent alarm sequences are not illustrated in the figure, as well as Fault 3, as the number of possible sequences after this malfunction has a higher order of magnitude. The faults, considered more complicated based on the variety and length of the sequences can be seen in the upper right quadrant of the graph, while the ones at bottom left corner can be considered as a more easily identifiable malfunction.

Table 4.4: The statistical evaluation of the alarms and sequences

| Tag of Fault | Number of Characteristic Seq's | Max. Length of Seq's | Max. Conf. of Seq's | Max. Support of Seq. with Max. Conf. | 1st Event after Failure in Seq. with Max. Conf. | Length of Following Seq. after Fault |
|---|---|---|---|---|---|---|
| 1 | 7 | 3 | 0.975 | 0.377 | 162 | 1 |
| 2 | 3 | 2 | 1 | 0.387 | 61 | 1 |
| 3 | 1357 | 8 | 1 | 0.387 | 82 | 1 |
| 4 | 9 | 3 | 1 | 0.387 | 112 | 1 |
| 5 | 5 | 2 | 0.96 | 0.371 | 122 | 1 |
| 6 | 3 | 2 | 1 | 0.387 | 141 | 1 |
| 7 | 3 | 2 | 1 | 0.387 | 161 | 1 |
| 8 | 0 | 0 | - | - | - | - |
| 9 | 0 | 0 | - | - | - | - |
| 10 | 1 | 1 | 1 | 0.387 | 232 | 1 |
| 11 | 1 | 1 | 0.445 | 0.172 | 242 | 1 |

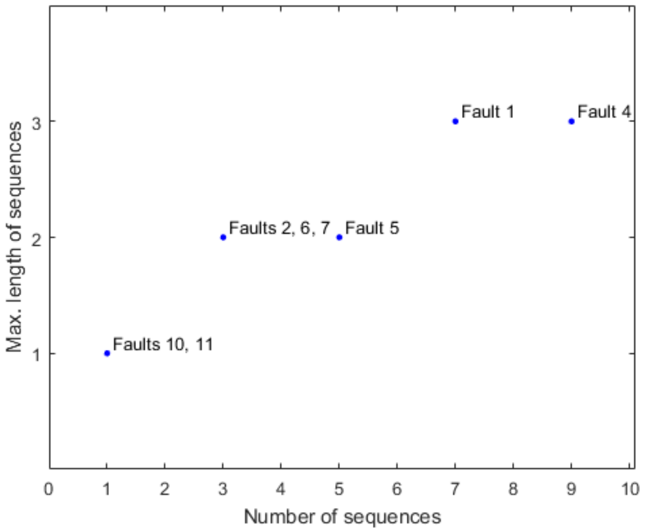


FIGURE 4.9: The illustration of the complexity of faults based on the variety and length of their following sequences (the No. of faults follows the order presented in Table 9.1). The more complicated faults are at the upper right corner of the figure, as these faults can be followed by different and often long series of events. The easily identifiable alarms are at the bottom left corner consequently. The figure does not contain faults that do not generate frequent alarm sequences and Fault 3, as the number of possible sequences after this malfunction has a higher order of magnitude.

The complexity of the analysis of the alarm sequences can be well illustrated with the examination of the time distribution of the sequences, according to Figure 4.10. If only the time distribution of the single alarms is considered, as can be seen in the upper two graphs of Figure 4.10, it is well apparent that the lognormal time distribution of the implemented faults is not followed by the alarm time distribution as it varies in time. In the case of longer sequences (sequences containing more events), as it is presented in the bottom four graphs of the figure, the experience shows longer time periods with the increasing sequence lengths, which is probably caused by the spillover effect of the processes.

It is also interesting to examine how hard it is to distinguish different faults based on the alarm types caused by them. The difficulty arises when the consequences of these faults are similar. To investigate the problem, a similarity measure was proposed according to Equation 4.8 to compare the announciated alarm tags caused by each fault.

$$Sim_{i,j} = \frac{|E_i \cap E_j|}{|E_i \cup E_j|} \tag{4.8}$$

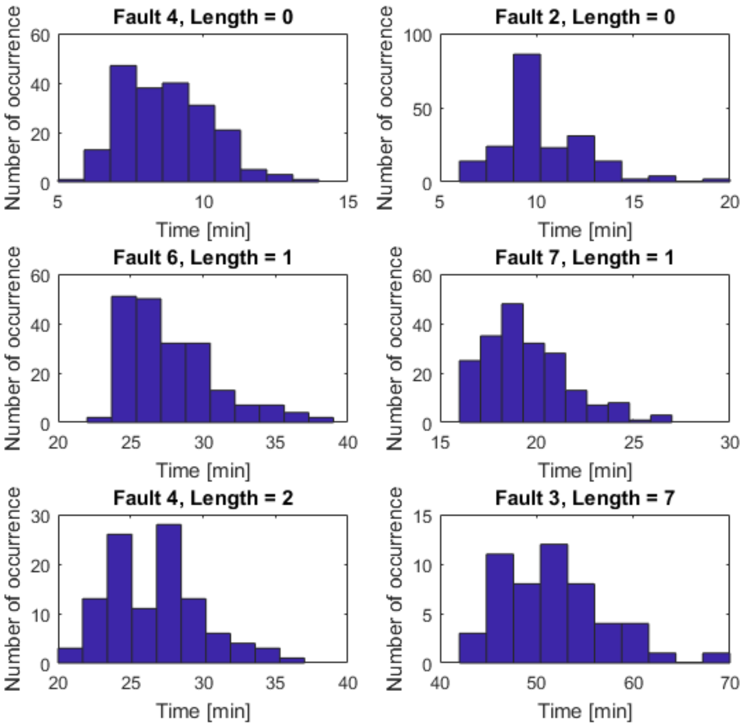


FIGURE 4.10: The time distribution of selected sequences. Even the time distribution of shorter sequences presented at the upper graphs can vary significantly in time, but the spillover effect of the processes is well illustrated by the longer sequences in the bottom graphs. $k = 0$-length patterns are degenerated patterns with only one state without any temporal predicates, as a pattern of $k+1$ states is connected with $k$ temporal predicates, is called a $k$-length temporal pattern.

The counted (nonzero) similarity ($Sim_{i,j}$) indexes are presented in Table 4.5. According to this, the fault pairs 1-7 (the fault of HAc fresh feed flow rate and the fault of circulation cooler heat flow) and 4-6 (the fault of the compressor heater heat flow and the fault of the absorber scrub heat flow) are considered problematic or similar during root cause analysis.

TABLE 4.5: The counted (nonzero) similarity indexes of each fault pair. According to the results, the 1-7 (the fault of HAc fresh feed flow rate and the fault of the circulation cooler heat flow) and 4-6 (the fault of the compressor heater heat flow and the fault of the absorber scrub heat flow) fault pairs are hard to distinguish based on the occurring alarm messages.

| Fault number | Fault number | Similarity measure |
|---|---|---|
| 1 | 3 | 0.231 |
| 1 | 7 | 0.667 |
| 3 | 5 | 0.071 |
| 3 | 7 | 0.154 |
| 3 | 10 | 0.077 |
| 4 | 6 | 0.667 |

The distinction of these similar faults can be carried out with the inspection of the appearing alarms (events) as it is presented in Table 4.6. In both fault pairs, one of the faults (Fault 1 and 4) causes one more alarm type, which can be used for the identification of the root cause of the events.

TABLE 4.6: The alarms (events) appearing in the case of similar faults (the name of alarm tags can be seen in Table 10.2)

| Fault Tag | Alarms |
|---|---|
| 1 | 31, 161, 162 |
| 4 | 112, 141, 142 |
| 6 | 141, 142 |
| 7 | 161, 162 |

The causal relationship of Faults 4 and 6 can be considered obvious taking into consideration their physical meaning: Fault 4 is the manipulation of the compressor heater heat flow, while Fault 6 is the manipulation of the absorber scrub heat flow. Both of them controls the temperature of the given equipment, and their location is close to each other. Both of them triggers the high and low alarm of the circulation stream temperature, but Fault 4 also causes a high alarm for the compressor exit temperature. Fault 1 is the manipulation of the HAc fresh feed flow rate, and Fault 7 is the fault of the circulation cooler heat flow. Similarly to the former, the lack of the HAc storage can cause problems with the scrub stream temperature as well, as in this case, the flow rate of the scrub stream decreases and the applied heat flow can over or under heat it. The low HAc feed can cause low alarm on the HAc tank level as well.

In case of similar processes, the first appearing alarm can be suggestive of the root cause as well. Table 4.7 shows the number of the type of first alarms appearing after the occurrence of the process faults. Based on this, fault types 10 and 11 can be identified relatively obviously, but Fault 1, 4 and especially 3 is hard to determine as different alarms appear right after the malfunction.

### 4.3.6 Development of alarm suppression rules

The fundamental idea of alarm suppression is that incoming alarms with a high confidence should not be raised, and this reduction in the number of alarm messages can significantly reduce the workload of the operators. Sequence-based alarm suppression rules could be more beneficial than simple correlation-based methods

Table 4.7: The number of the type of the first alarms appearing after each fault

| Fault Tag | First alarm types |
|---|---|
| 1 | 3 |
| 2 | 2 |
| 3 | 10 |
| 4 | 3 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 0 |
| 9 | 0 |
| 10 | 1 |
| 11 | 1 |

since the information about the past events contained in the sequences can increase the probability the correct prediction of the next event. The transition between the states of the process can be predicted well with the use of the conditional probability described in Equation 4.4.

Aiming the definition of alarm suppression rules, the knowledge of the fault causing the alarm sequence is not needed, the exploration of the alarms occurring together and their sequence of occurrence are sufficient. Using the described sequence mining algorithm, the conditional probability of the transition of a $\Phi_{k-1}$ sequence to the $s_k$ state (or in this case alarm) is known, therefore, an intuitive mode to define alarm suppression rules is to define a threshold for this last transition between alarms at the end of the sequences. Following this approach, the process history described at the beginning of the sequence is utilized. To assure a high prediction accuracy, a considerably high threshold with a value of 0.8 was set for the confidence of transition between the last two states of the sequences during the definition of alarm suppression rules. The sequences meeting this rule end in 10 different alarm messages (with tags 31, 82, 141, 162, 211, 222, 231, 232, 251, 271). The distribution of the length of these sequences is illustrated in Figure 4.11.

Alarm 82 has a significantly higher number of applicable sequences, while for example the alarm 141 and 231 have only a few.

The ratio of the suppressed and nonsupressed alarms can be seen in Figure 4.12. It is interesting to see that alarm 141 had only four applicable sequences, but all of the alarms could be suppressed with the application of these few alarm suppression rules.

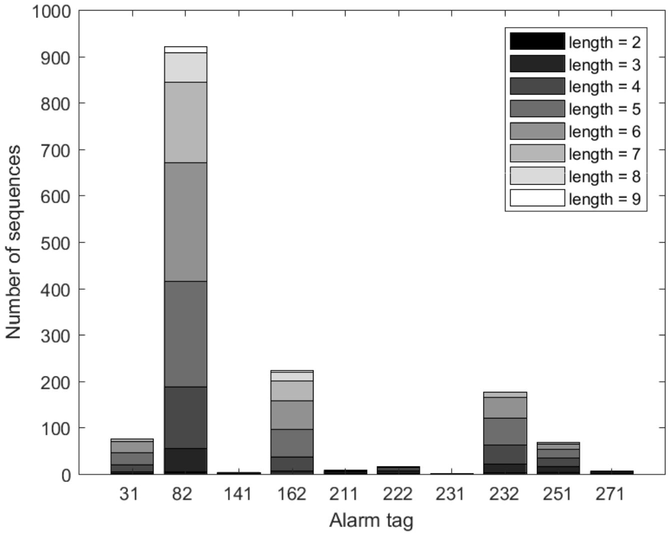


FIGURE 4.11: The number and length of sequences applicable for the suppression of the given alarms

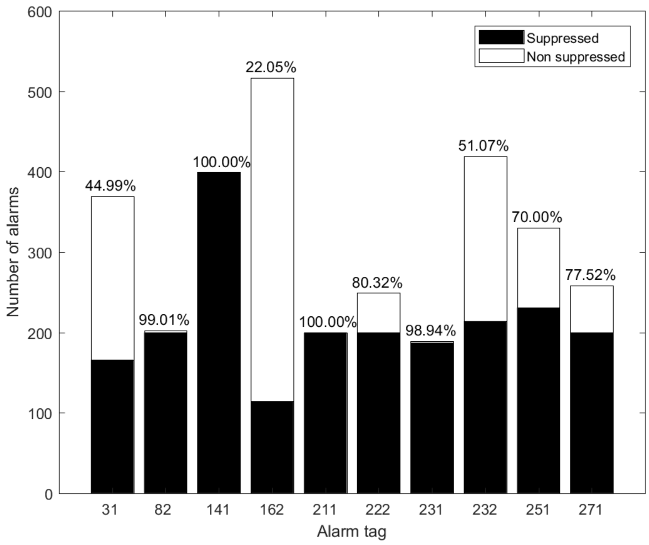


FIGURE 4.12: The ratio of the suppressed and nonsuppressed alarms, the percent of the suppressible alarms as the ratio of all occurring alarms of each alarm tag is provided at the top of each bar.

### 4.3.7 Limitations

Although in most of the cases the effectiveness of alarm management systems is measured based on the alarm rates, many studies have shown that the real problem is related to the informativeness of alarms [90, 91, 92, 93]. The concept of our method is that predictable alarms in most of the cases do not bring new information to the operators and the predictability can be evaluated based on the confidence of the alarm suppression rules. Although the number of properly suppressed alarms can be estimated based on the support and the confidence of the rules, these variables should be tuned by making a tradeoff between suppression power and fault diagnostic sensitivity.

The critical question of the present methodology is the sensitivity of the alarm suppression for the length of the window parameter and the support and confidence thresholds. The number of generated sequences is investigated, the results are presented in Section 4.3.3. However, it is important to mention that these parameters influence the amount of generated sequences, but the predictive accuracy of the methodology is just the question of the confidence of the transition between the last two alarms in these sequences. This gives the probability of the occurrence of the last alarm, which can be suppressed. If a confidence threshold of this last transition of 0.8 is chosen to suppress alarms, ostensibly 20 % of times that the alarms is missed. For the handling of this issue, during the online application in the future, the times when not the predicted alarm occurs the system should display it and only the correct predictions are to be suppressed. According to these limitations, the generated alarm suppression rules must be revised by a safety expert who supervises the transformation of the extracted sequences into alarm suppression rules, since evaluating the risk related to a superficially suppressed alarm requires detailed process relevant knowledge.

Finally, the risk of mistakenly confusing the alarm sequence of a previously known malfunction with negligible consequences with a more severe, formerly unknown fault with similar alarm messages will never be fully reduced. Moreover, this underpins the direction that the work of the operators can be supported by intelligent solutions, however, in the chemical industry, there is a very long road to replacing a human operator as new, previously unknown situations can occur any time. That is why the application of expert knowledge during the development of data-driven alarm management solutions is crucial.

## 4.4 Chapter summary

Alarm management is a crucial task of improving process safety and reduction of operator workload. The log files generated by process control systems can provide valuable information for the exploration of causal relationships between alarm signals, which can be used to form alarm suppression rules and support root cause analysis. To incorporate the dynamics of the process into the identification of the alarm suppression rules, a multi-temporal sequential pattern mining-based algorithm is proposed. The analysis of a laboratory-scale water treatment system showed that the algorithm could efficiently map the operational pattern of the raw water storage tank from the set of discrete events. Since a more sophisticated and detailed analysis of the applicability of the method was aimed, the simulator of a vinyl-acetate production technology was extended to support and ensure the reproducibility of our results and motivate future research of alarm management and fault diagnosis techniques.

Based on well-documented scenarios a detailed sensitivity analysis was elaborated to evaluate the effects of the parameters of the algorithm (time window, support and confidence threshold values) on the number of generated sequences. The results of the time distribution analysis of sequences highlighted that the malfunctions start a spillover effect on the process, which cannot be neglected during the definition of alarm thresholds and general hazard analysis principles. The proposed probabilistic interpretation of the model helps the evaluation of how suppressible the alarms are. The tuning of the parameters describes the trade-off between the number of suppressed alarms and the confidence of the prediction accuracy. In the studied dataset ten alarm tags met the chosen 0.8 value of state transition threshold and using the identified alarm suppression rules, approximately 67.4 % of these alarms could be suppressed.

The results confirmed that the proposed methodology is applicable in evaluating similarities of the faults and the significance and accuracy of the alarm suppression rules extracted from the frequent temporal patterns.

The application of multi-temporal frequent sequence mining provides the basis of my 1.1 and 1.2 thesis findings, as will be summerized in Section 8.

# Chapter 5

# Hierarchical Frequent Sequence Mining for the Analysis of Production Processes

In case of complex chemical production systems faults and malfunctions generate alarm cascades, hindering the work of the operators and making the fault diagnosis a complex and challenging task. The core concept of my work is the incorporation of the hierarchical structure of the technology in a multi-temporal sequence mining algorithm to group the high number of variables. The spreading of the effect of malfunctions over the plant is well-traceable on the higher levels of hierarchy, while the critical elements of the spillover effect can be detected on the lower levels. I have proposed confidence-based goal-oriented measures to describe the orientation of fault propagation giving a good insight of causality on a local level of the process, while the network-based representation gives a global view of causal connections. The effectiveness of the proposed methodology is presented on the analysis of the alarm and event log database of an industrial delayed coker plant, where the complexity of the problem and the size of the event log database requires a hierarchical constraint-based representation.

## 5.1 Introduction

The problems of the management of alarms provided to the operators of chemical production systems are closely interconnected with the fault detection of chemical processes. Alarm management is the efficient design, implementation, operation and maintenance of industrial process alarms. According to the guidelines of the Engineering Equipment and Materials Users' Association (EEMUA), the purpose of an alarm system is to help the operators to keep the processes in the normal operating range by redirecting the operator's attention towards critical plant conditions requiring timely assessment or action [3]. In chemical production plants, as part of the alarm management system, discrete events, e.g. alarms, warnings, operator actions and system messages are recorded in alarm management databases. However, alarm management guidelines suggest having only one alarm for a single malfunction of the process, in distributed control systems (DCS) the definition of alarms has no significant cost, leading to the configuration of more and more alarms with often meaningless alarm levels resulting in the overload of the operators. With the analysis of the recorded discrete events of the process we are able to build data model-based solutions to determine the root cause of the event series [66], to predict the occurring future events [64], determine the frequently occurring operation strategies for advanced automation solutions [53] and in general to help and reduce the work of the operators.

For the processing of the high amount of historical data recorded in chemical plants different approaches are in common, a thorough collection and classification of process history-based fault detection and diagnosis techniques are given by Venkatasubramanian *et al.* [59]. Among the tools of chemometrics, we can find several alarm management solutions as well. The monitoring of PCA-based $T^2$ and Q statistic is recommended by Kondaveeti *et al.* [114] and Gupta *et al.* [115]. A Gaussian mixture model is applied by Wang *et al.* to monitor multimode processes [116]. Chenaru *et al.* developed a real-time decision tree analysis methodology based on the multivariate analysis of measured sensor datasets [117].

However, alarm management databases are mainly formed of discrete events raising the problem of discrete event-based solutions. Just to simply reduce the operator workload and the triggered spurious alarm signals, conventional techniques like alarm limit deadbands [60], delay-timers [61] or filtering [97] are widely applied. In the case of more advanced techniques, the aim is to find overlapping and redundant alarms that should be grouped, which groups can form association rules and can be eventually applied for the detection of failures, the prediction of future events or the suppression of predictable future alarms. To achieve this, there are two different approaches in common. First, we can apply correlation analysis based techniques, where we would like to explore frequently co-occurring alarms within a short time period [101]. Second, with the use of advanced data mining algorithms, we can look for frequently occurring operational patterns to identify consequential alarms and their evolution path [109].

The data-driven analysis of chemical processes can be significantly improved by capturing the fundamental connectivity relationships of the analysed process [118]. Following this approach, Yim *et al.* developed a software for the simultaneous consideration of process topological information and historical process data [119, 120]. This software has been improved to a cause and effect analysing tool [121] and to be able to derive plant connectivity information from process drawings [122] making it ready to merge process models and plant topology. Realising the importance of process hierarchy to find more informative alarm connections, I would like to improve a data-mining approach that can effectively handle hierarchical constraints of process topology and is able to reveal more informative alarm co-occurrences and at the same time reduce spurious connections.

Mining association rules with hierarchical constraints of the process means the formulation of multiple-level association rules. To achieve this, data is needed at multiple levels of abstraction and efficient algorithms for multiple-level sequence mining [123]. Viewing the problem as a bottom-up approach, assuming an imaginary chemical technology, this means that besides showing that in 30 % of the cases the *high temperature in the top of the first column* is followed by a *high pressure in the reboiler of the second column* within a defined time window, but process experts can be interested in that in 90 % of the event series any alarm in the *first column* is followed by the alarms of the *second column*, therefore, the spillover effect of the malfunction can be revealed. Moreover, this approach can help to neglect alarm patterns that are not supported by the connectivity information of the process. Thus, the connection of alarms of distant units of a process can be avoided. The topic of mining multiple-level association rules is thoroughly studied in the past decades. Han and Fu proposed a group of algorithms based on a top-down deepening method and the a priori principle [123]. Following this approach, the theme of multi-dimensional sequential pattern mining was proposed, integrating the topics of multidimensional analysis and sequential data mining [124]. The algorithms developed by Eavis and Zheng exploit an existing frequent pattern tree (FP-tree) structure to reveal conventional and cross-level frequent patterns as well [125]. Realising the problem of mining association rules in very large databases, Tseng proposed the methodology of hierarchical partitioning using a novel data structure, the frequent pattern list, for mining frequent itemsets [126]. Melo and Völker investigated the problem of mining large knowledge bases such as DBpedia using the hierarchical background knowledge of the database [127]. Examination of the interestingness of the mined association rules was combined with the mining of hierarchical market basket analysis in the work of Zekić-Sušac and Has [128]. Dewitt *et al.* analysed network alarm data with the incorporation of network topology constraints to avoid implausible sequences [129].

Noticing the importance of alarm management, more and more modern chemical plants record the occurring alarms, warnings and operator actions in response to these situations to monitor the performance of the alarm system. However, in complex and large production plants (especially in refineries) the size of these databases and the diversity of the stored events can easily exceed the processing capacity of process engineers and traditional data mining algorithms. Noting that each event can be clearly classified on every hierarchical level of the production plant, the occurring discrete events have their independent interpretation on these levels of hierarchy. Completing the event log database with the classification of the recorded tag name of the process variable, we can give the specific measuring circuit, asset, operating unit, operating division, etc. where the specific event happened. This more detailed database helps us to reveal the connections between the operation subunits and illustrate how the effect of a malfunction will spillover the examined system.

In the present chapter, realising that the mining of frequently occurring alarm sequences in large alarm management databases is almost computationally infeasible due to the generation of spurious sequences of the alarms of distant elements of the process, I would like to describe a methodology for the generation of more informative sequences. The contribution of the present chapter is twofold: first, I would like to merge the fields of hierarchical sequence mining and frequent pattern-based alarm management solutions. Since the present algorithms in the literature of hierarchical sequence mining are in the absence of temporality, I improve the formerly introduced multi-temporal sequence mining algorithm [64] with the introduction of the hierarchical constraints of the analysed process. Applying the extended algorithm, I demonstrate how the hierarchical topology of the process helps the exploration of more informative causal relationships between the alarms and process units, and the generation of longer alarm sequences, therefore giving an insight on the spillover effect of malfunctions. Secondly, applying the results of the field of heuristic process mining [130], I have proposed measures to quantify the orientation of fault propagation over the process. These measures give a good insight of the direction of the connection of process elements on a local level and give the opportunity to define the network of these connections. Using the network-theory-based metrics, like PageRank and betweenness centrality measures [131], I present how the central and ending elements of alarm cascades can be detected. Moreover, during the presentation of the results, novel visualization solutions are proposed, like the sunburst plot for the illustration of alarm load on different hierarchical levels, the parallel coordinates for the visualization of fault propagation over the process or the network-based representation of the direction of connections during the spillover.

The roadmap of the chapter is as follows. The sequence mining problem, the probabilistic interpretation of multi-temporal sequences and the proposed algorithm are described in Sections 5.2.1, 5.2.2 and 5.2.3, respectively. Section 5.2.4 describes the measures proposed for the quantification of alarm spreading and the network-based representation of spillover effect. The description of the analysed industrial dataset and its preprocessing in Section 5.3.1 is followed by the evaluation of the hierarchical sequences of alarms in Section 5.3.2. The effect and extent of the spillover effect of malfunctions are analysed in Section 5.3.3 using the proposed probability measures and network-based representation. Finally, the discussion of the results is followed by concluding remarks in the Conclusions section.

## 5.2 Hierarchical mining of multi-temporal alarm sequences

In this section, first, the formulation of hierarchical multi-temporal alarm sequences is described, then the probabilistic interpretation of these temporal sequences is proposed and the proposed sequence mining algorithm is discussed.

Finally, how the resulted sequences can be used to detect the critical elements of alarm cascades is illustrated.

### 5.2.1 Hierarchical multi-temporal alarm sequences

The structure of the alarm & event-log database, the states and events of a technology, moreover, their mathematical formulation was already described in the previous section (Section 4). The concept of temporally connected events are also explained in the previous section. Therefore, in the present section, I focus on the description of the extension of this methodology with for the hierarchical analysis of processes.

Supposing a complex production system, with $n_h$ hierarchical levels, we can determine the place of origin of the $pv$ process value from the bottom level of the tag names to the top level of plant hierarchy as well. In the present work, I follow the guidelines of the ISA-95 standard [29], which recommends applying the following hierarchical levels (enumerated in top-down order): enterprise, site, area, production unit, unit and the level of sensors and actuators (although the standard focuses on the first four levels). Figure 5.1 illustrates a didactic example for easier understanding, supposing a refinery site of an enterprise, with the delayed coker and the distillation areas illustrated. These areas can be further divided into production units, such us columns, reactors, separators, etc. Production units are composed of units, the furnaces, pumps and any further parts of the production units. At the bottom level of the presented hierarchy, we have the sources of signals, i.e. the sensors and actuators. Following this concept, the $pv$ process indicator can represent the origin of the discrete event on the given hierarchical level ($h$). Therefore it is a function of the hierarchical level ($pv(h)$).

In this context, the previous example of high inlet temperature can be expanded to the higher hierarchical levels, as it happened in the *Furnace #1* unit, which is the unit of the *Coker Column #1* production unit, in the *Delayed Coker* area of the *Refinery*, etc., on any arbitrary levels. The set of states on the first (and highest) hierarchical level is denoted by $\mathcal{S}^1$, the states of this hierarchy level are easily denoted by their subscripts as $\mathcal{S}^1 = s^1, s^2, ...s^i, ...s^{n_1}$. Each of these states is the set of the underlying states of the second hierarchical level ($\mathcal{S}^2$), therefore the $s^i$ state can be unwrapped to $s^{i,1}, s^{i,2}...s^{i,j}...s^{i,n_i}$ states, similarly as the *Delayed*

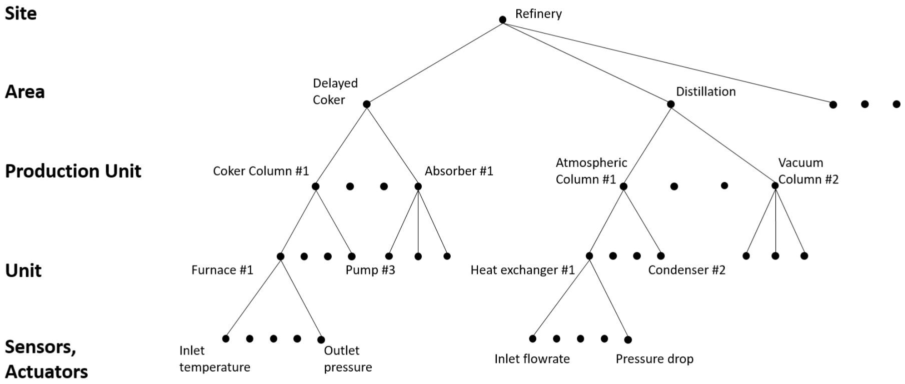


Figure 5.1: The hierarchy of the process according to the guidelines of the ISA-95 standard [29]

*Coker* area can be unwrapped to its production units, where $n_i$ is the number of elements belonging to $s^i$ on the lower hierarchical level.

For easier understanding, Figure 5.2 gives an illustrative example of the connection between the hierarchy of the process and the occurring temporal events. The three production units are marked with 1, 2 and 3 in brackets and the states of the events occurring in these production units are $s^1$, $s^2$, $s^3$, respectively. The units of the production units are marked with an additional, comma-separated upper script number, therefore we have states $s^{1,1}$, $s^{1,2}$, $s^{2,1}$, etc. On the bottom level of sensors and actuators we use the notation of $s^{1,1,1}$, $s^{1,1,2}$, etc., therefore the notation of the state of an event on the lowest hierarchical level clearly indicates which process variable (sensor/actuator) alarmed in the corresponding unit and production unit. The left side of Figure 5.2 illustrates the schematic plant layout of the presented example, while the right side of the figure shows the occurring temporal events. The horizontal axis represents the time, and the rows of the vertical axis indicate the different states. Since we have periods of faultless production in our system, we can define an *event trace* of events between two periods of faultless production, assuming some kind of causal connection between the events of the same event trace. The definition of event traces is explained in details in Section 5.3. In the presented example the occurring events are assumed to be in the same, first event trace. Table 5.1 shows the *DB* alarm management database of the occurring events completed with the hierarchical states of the process.

Table 5.1: Example $DB$ event log database representing an event with its event trace, event id, the occurring state in the specific event on three different levels of process hierarchy and the starting and ending time of the event.

| Trace id | Event id | $\mathcal{S}^1$ | $\mathcal{S}^2$ | $\mathcal{S}^3$ | Starting time | Ending Time |
|---|---|---|---|---|---|---|
| 1 | $e_1$ | $s^1$ | $s^{1,1}$ | $s^{1,1,2}$ | 0 | 4 |
| 1 | $e_2$ | $s^1$ | $s^{1,2}$ | $s^{1,2,1}$ | 0 | 4 |
| 1 | $e_3$ | $s^1$ | $s^{1,1}$ | $s^{1,1,1}$ | 2 | 8 |
| 1 | $e_4$ | $s^3$ | $s^{3,1}$ | $s^{3,1,1}$ | 2 | 6 |
| 1 | $e_5$ | $s^3$ | $s^{3,2}$ | $s^{3,2,1}$ | 4 | 12 |
| 1 | $e_6$ | $s^3$ | $s^{3,2}$ | $s^{3,2,2}$ | 10 | 11 |
| 1 | $e_7$ | $s^2$ | $s^{2,1}$ | $s^{2,1,2}$ | 12 | 18 |
| 1 | $e_8$ | $s^2$ | $s^{2,1}$ | $s^{2,1,1}$ | 16 | 20 |

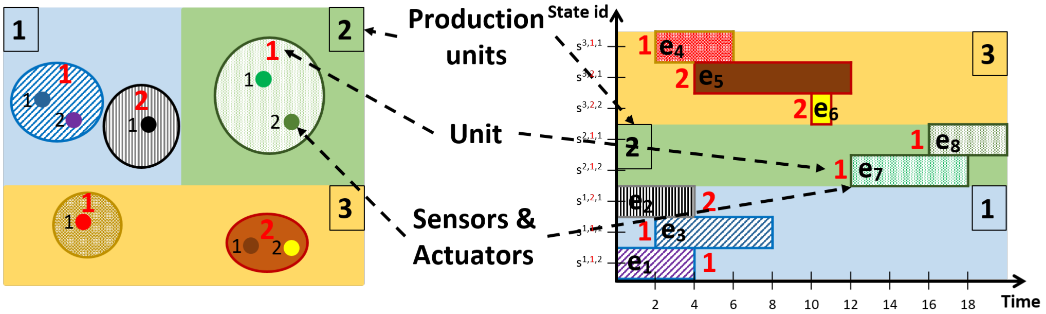


Figure 5.2: The visual illustration of the connections between the process or plant hierarchy and the temporal sequences. The figure on the left illustrates the schematic representation of the process layout, while the figure on the right shows the visual representation of the recorded events. The horizontal axis represents the time, and the rows of the vertical axis represent the states. The numbers in brackets 1, 2, 3 indicate the production unit of the occurring discrete event, while the units inside of them are represented by the red numbers 1, 2, 3 in the circles. On the lowest hierarchical level we can see the place of origin of the recorded event, therefore the signals of the sensors and actuators (e.g. alarms, warnings or operator interventions) marked with numbers as well. The vertical length of the bars indicates the length of the time of the given event.

### 5.2.2 Probabilistic interpretation of hierarchical sequences

The aim of the present study is to give an insight on how the different events are correlated in a chemical plant. The characterization of these co-occurrences requires the probabilistic-based interpretation of the occurring events. However, the basic principles of the applied probability-based interpretation were already described in Section 4. For the definition of the number of supporting events (Equation 4.1), the conditional probability of the occurrence of the $s_k$ state after a $\Phi_{k-1}$ sequence (Equation 4.4), the probability of the occurrence of a longer sequence (Equation 4.3) and the definition of the confidence measure (Equation 4.5) please refer to Section 4. Here, I focus on the difference compared to these equations, as the probability of occurrence of a state on a higher level of the hierarchy and the definition of the support value can be fitted to multiple levels of hierarchy.

Applying the a priori principle of frequent item mining for multiple levels of hierarchy, we can easily conclude that the probability of occurrence of a state on a higher level of the hierarchy is the sum of the probability of occurrence of the incorporated states on a lower level of the hierarchy. This hierarchical relationship is described in Equation 5.1.

$$P(s_i) = \sum_{j=1}^{n_i} P(s_{i,j}) \tag{5.1}$$

To give a measure of the frequency of occurrences of a given state, the number of instances is normalized by the number of instances of the most frequent state in the given database as presented in Equation 5.2.

$$support(\Phi) = supp(\Phi)/|E(h)| \tag{5.2}$$

where $|E(h)| = \underset{j=1,2,\ldots,N}{max}(|E_j|)$, $N$ denotes the number of states in the $DB$ temporal database, $E_j$ is the set of events supporting state $s_j$, $|E_j|$ is the cardinality of $E_j$ (the number of events in $E_j$) on the $h$ level of hierarchy. This support value measures the frequency of occurrences of each pattern, therefore gives the basis of the determination of frequent sequences. Frequent pattern mining algorithms usually define frequent patterns based on the $support(\Phi) \geq minSupp$ equation, where $minSupp$ is the support threshold, an input parameter of the algorithm. The goal of frequent pattern mining is to find all the frequent patterns for a given support threshold. In the case of multiple levels of hierarchy, all the incorporating higher level states of a frequent state are consequently frequent. Reversing this concept, using the same support threshold on every level, we can use the frequent states on higher hierarchical levels to neglect infrequent ones on lower levels of hierarchy.

### 5.2.3 The hierarchical multi-temporal mining algorithm

We have previously fine-tuned the algorithm published in [112] to make it more convenient and applicable for alarm management purposes [64]. In the present chapter, I further expand the algorithm by implementing the handling of multiple levels of hierarchy.

The pseudo-code of the Hierarchical Multi-Temporal Mining method is described in Algorithm 2. The mining process starts at the highest level of hierarchy. First, the frequent states of the hierarchical level are determined and stored in the $F_{1,0}$ variable, denoting the frequent $k = 0$-length patterns of the first hierarchical level. The longer $k + 1$-length frequent pattern candidates are generated step by step by combining the found $k$-length frequent patterns and the events of the frequent states stored in $F_{1,0}$. The candidates are tested against the support and confidence thresholds ($minSupp$ and $minConf$) and the frequent patterns are added to the $F_{1,k+1}$ variable containing the $k+1$-length sequences of the first hierarchical level. This procedure continues until no more $k+1$-length patterns are found to expand. Once the frequent temporal patterns of the highest level of hierarchy are generated the algorithm goes on to the lower levels of $\mathcal{S}list$ indicating the levels of hierarchy which we intend to mine in an ascending order. The **Hierlevelminer** function requires the frequent patterns of the higher hierarchical level, the No. of the level of hierarchy for mining and the states of the hierarchical level. The function revisits every temporal instance of the sequences of the next higher level of hierarchy and examines the temporal pattern of states on the defined hierarchical level of the given events if they meet the support and confidence thresholds. The found frequent temporal patterns are stored in $F_{h,k}$, where $h$ and $k$ indicate the examined level of hierarchy and sequence length.

### 5.2.4 Measuring the orientation of the propagation of the effect of process malfunctions

In the present investigations, I would like to reveal the orientation of the propagation of malfunctions over the process. In order to quantify the orientation of the revealed causal connections, I have used the *dependency* measure originally applied in the field of process mining [130] as presented in Equation 5.3. This frequency based metric indicates how certain we are in the dependency between $s_0$ and $s_1$.

**Algorithm 2 Hierarchical Multi-Temporal Mining** method

```
Require: $DB$: Temporal database
    $minSupp$: Support threshold
    $minConf$: Confidence threshold
    $\mathcal{S}list$: Levels of hierarchy for mining in ascending order
$F_{1,0} = 0$
for every $\mathcal{S}^{\infty}$-level state $s$ in $DB$ do
    if $supp(s) \geq minSupp$ then
        $F_{1,0} = F_{1,0} \cup \{s\}$
    end if
end for
$C_1 = \{s_i \overset{R}{\Rightarrow} s_j | s_i \in F_{1,0}, s_j \in F_{1,0}\}$
i=1
while $F_{1,i-1} \neq 0$ do
    $F_{1,i} = 0$
    for every sequence $\Phi \in C_i$ do
        if $supp(\Phi) \geq minSupp$ and $conf(\Phi) \geq minConf$ then
            $F_{1,i} = F_{1,i} \cup \{\Phi\}$
        end if
    end for
    $C_{i+1} = \{s_i \overset{R}{\Rightarrow} s_j | s_i \in F_{1,i}, s_j \in F_{1,0}\}$
    i=i+1
end while
for every level of hierarchy $h \in \mathcal{S}list$ do
    Hierlevelminer$(F, h, \mathcal{S}^h)$
end for
Function Hierlevelminer($F$:Frequent sequences of the higher hierarchical level, $h$: No. of examined level of hierarchy, $\mathcal{S}^h$: states of the examined level of hierarchy )
begin
$F_{h,0} = 0$
for every sequence $\Phi \in F_{h-1}$ do
    for every temporal instance $\phi$ supporting $\Phi$ do
        $dum = 0$
        for every event $e \in \phi$ do
            $dum = dum \cup \{pv(h)\}$
        end for
        if $dum \notin F_h$ and $supp(dum) \geq minSupp$ and $conf(dum) \geq minConf$ then
            $F_{h,i} = F_{h,i} \cup \{dum\}$
        end if
    end for
end for
EndFunction =0
```

The support values of sequences $s_0 \overset{R_1}{\Rightarrow} s_1$ are summed for every type of temporal predicate.

$$dependency(s_0 \overset{R_1}{\Rightarrow} s_1) =$$
$$= \frac{\sum\limits_{R_1 \epsilon\ \{E,\ B,\ D,\ O\}} supp(s_0 \overset{R_1}{\Rightarrow} s_1) - \sum\limits_{R_1 \epsilon\ \{E,\ B,\ D,\ O\}} supp(s_1 \overset{R_1}{\Rightarrow} s_0)}{\sum\limits_{R_1 \epsilon\ \{E,\ B,\ D,\ O\}} supp(s_0 \overset{R_1}{\Rightarrow} s_1) + \sum\limits_{R_1 \epsilon\ \{E,\ B,\ D,\ O\}} supp(s_1 \overset{R_1}{\Rightarrow} s_0) + 1} \tag{5.3}$$

The dependency measure is always between -1 and 1. To illustrate its applicability, imagine a dataset, where the generated 1-length frequent sequences of $s_0 \overset{R_1}{\Rightarrow} s_1$ have 4 temporal occurrences and we have no occurrence for $s_1 \overset{R_1}{\Rightarrow} s_0$. In this case, the dependency is $dependency(s_0 \overset{R_1}{\Rightarrow} s_1) = 4/5 = 0.8$ meaning that we are not sure of the dependency relation. However, if we have 100 occurrences of this sequence, dependency is $dependency(s_0 \overset{R_1}{\Rightarrow} s_1) = 100/101 = 0.9901$ indicating that we are almost completely sure in this dependency relationship. If not just the 100 occurrences of $s_0 \overset{R_1}{\Rightarrow} s_1$, but the reverse of this sequence ($s_1 \overset{R_1}{\Rightarrow} s_0$) is present for 5 times as well, the value of $dependency(s_0 \overset{R_1}{\Rightarrow} s_1)$ drops down to $(100 - 5)/(100 + 5 + 1) \approx 0.90$ indicating that $s_1$ is probably highly dependent of $s_0$.

The dependency measure is calculated from the number of occurrences of the analysed sequence. However, the confidence measure gives an insight into the probability of the pattern evaluates across the states of the revealed pattern. Applying this reliability measure, I have defined a metric called *directionality* which is calculated from the conditional probability of transition from the anticipating state to the following one as presented in Equation 5.4. The confidence values of the sequences are summarised for all types of temporal predicates. We should note that in the case of a 1-length temporal pattern containing two states, the confidence measure and the conditional probability of state transition are exactly the same. The directionality shows a measure that how confident we are in that once $s_0$ has occurred it will evolve to $s_1$ related to the confidence of the $s_1$ state will evolve to $s_0$ over time. The directionality ranges between -1 and 1 similarly to the dependency and a value higher than 0 indicates that we are more confident in the $s_0 \overset{R_1}{\Rightarrow} s_1$ direction of their temporal relationship.

$$directionality(s_0 \overset{R_1}{\Rightarrow} s_1) =$$
$$= \sum_{R_1 \epsilon \ \{E,\ B,\ D,\ O\}} conf(s_0 \overset{R_1}{\Rightarrow} s_1) - \sum_{R_1 \epsilon \ \{E,\ B,\ D,\ O\}} conf(s_1 \overset{R_1}{\Rightarrow} s_0) \quad (5.4)$$

In the following, I have applied the dependency, and the directionality measures for 1-length temporal sequences containing 2 states, but both measures are applicable for longer sequences as well.

The confidence of transition between states can be applied for the definition of a weighted directed graph to give a global view of the propagation of process malfunctions. The nodes of the network are the process elements on the examined level of hierarchy, while the weights are the confidence of the propagation of the effect of malfunctions over the related nodes. This network-based representation is not just advantageous for its illustrative visualization, but it gives the opportunity of the calculation of centrality measures of network-theory. In the present work the PageRank [132] and betweenness measures [133] are applied. The applicability of the defined measures is presented in Section 5.3.3.

## 5.3 Industrial case study

In this section, the historical alarm data of an industrial delayed coker plant is analysed. First, the original dataset and its preprocessing are described and then the effectiveness of the proposed methodology is demonstrated. All the tag names and identifiers are masked due to confidentiality.

### 5.3.1 Description and preprocessing of the analysed dataset

The methodology is demonstrated through the analysis of the delayed coker plant at the Danube Refinery of the MOL Group. An approximately 4-month-long operational period was analysed with more than 2000 process tag names on the level of sensors and actuators recorded in almost 400 units, which are located in 19 production units so that our example of application can be considered as a realistic and challenging case study. The process flow diagram of the analysed technology can be seen in Figure 5.3.

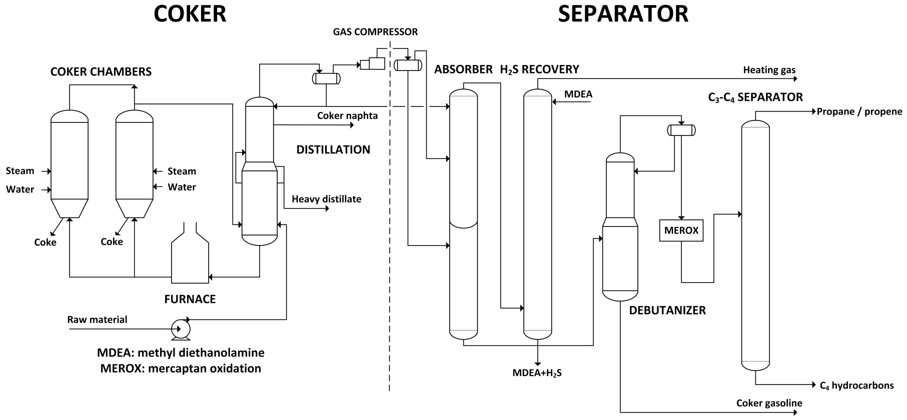


Figure 5.3: The process flow diagram of the analysed industrial delay coker plant. The plant is divided into two main parts: one for the production of coke and one responsible for its separation.

An insight into how the intensity of the alarms varies with time in the 19 process units in the preprocessed database is given in Figure 5.4 in the form of a *high density alarm plot* [20]. The horizontal axis of the high-density alarm plot usually indicates the examined time domain divided into temporal bins with a certain length, while the vertical axis shows top bad actors of the analysed alarm management system. The bins of this graph are colour coded indicating the number of instances of the corresponding alarm in the related temporal period. As a reference for easier interpretation of the illustration, it should be noted that 300 alarms per day (one alarm in every five minutes) is considered manageable by an operator according to the guidelines of EEMUA [3], although we cannot publish the number of alarms in the related process units due to confidentiality.

To identify chattering alarms,the modified chatter indices of each alarm tag are calculated as presented in the Appendix. Since we know that nuisance alarms following an actual alarm will not last more than a specified time period, the run lengths longer than 1200 seconds ($\tau = 1200$) are neglected and the modified chatter indices ($\psi_{\tau=1200}$) are calculated. During the calculations only alarm tags with a minimum of 10 annunciations are considered due to practical reasons. The *run length distribution histogram* illustrates the alarm counts with the related run lengths. The horizontal axis indicates the run lengths, while the vertical axis represents the alarm count. An example of the run length distribution of alarm tags can be seen in Figure 5.5. The left plot of the figure illustrates a highly

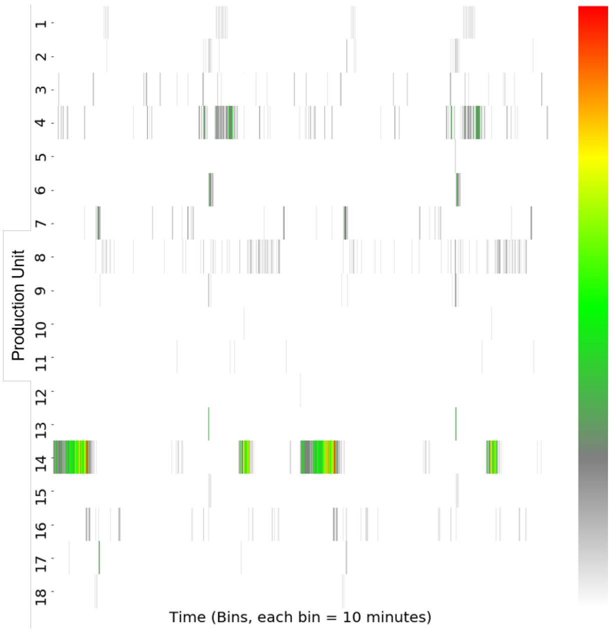


FIGURE 5.4: High density alarm plot of the production units.

chattering alarm; it is well-visible from the high number of alarms with short run lengths and from the high value of chatter index. In contrast, the right side of the figure shows an alarm tag that more likely to have informative temporal instances, as suggested by the smaller chatter index and by the small number of very short run lengths.

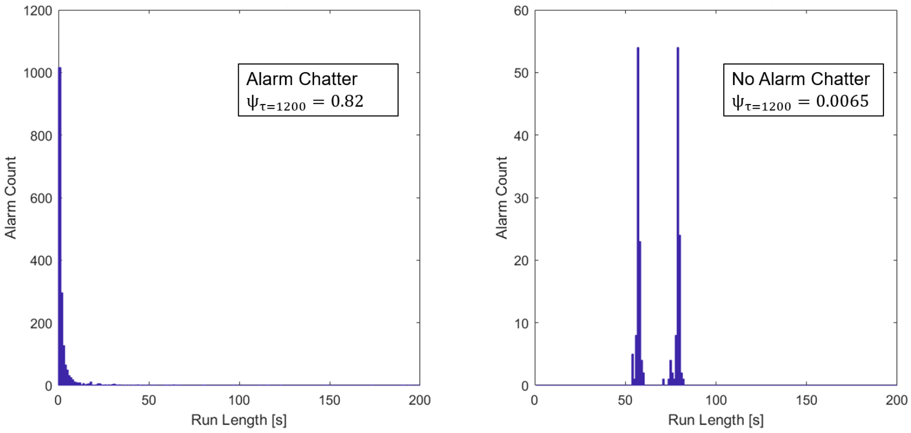


FIGURE 5.5: An example of the run length distribution of a chattering alarm (left) and a non-chattering one (right). The horizontal axis shows the different run lengths of the alarm tag (up to 200 seconds), while the vertical axis shows the alarm count with the given run length.

Figure 5.6 shows the calculated chatter indices in ascending order of the alarm tags with the highest chatter indices. Based on the calculated chatter indices we defined a run length limit below which the temporal instances are ignored, this is indicated on the right-side vertical axis of Figure 5.6, for the detailed description of the calculations see the Appendix. I have also plotted the recalculated modified chatter indices after I have cleared the first temporal instances of the run lengths below the calculated threshold. As a result of clearing, the chatter index of the problematic alarm tags has significantly decreased, although in some cases it is still above the defined cutoff value (0.05 alarms/s). This could be surprising, but after the clearing of certain temporal instances, alarms with short run lengths can still be present in the case of highly chattering tags.

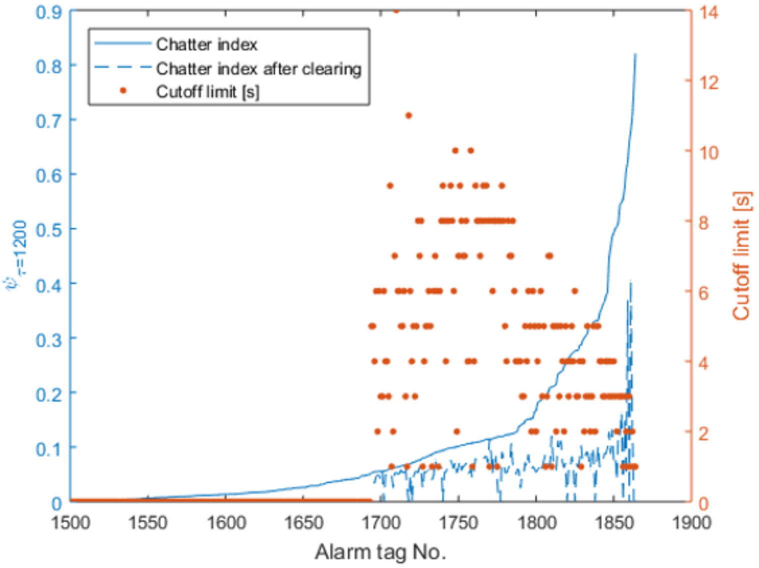


FIGURE 5.6: The calculated modified chatter indices ($\psi_{\tau=1200}$), the run length threshold under we neglect the temporal instances of these alarms and the recalculated modified chatter indices. The horizontal axis shows the type of alarm tags in ascending order based on their modified chatter index value.

For practical reasons, too short or too long alarms are also neglected. A reaction time of approximately 20 seconds for chemical process operators was suggested by Buddaraju [134]. Therefore, alarm instances shorter than 10 seconds are cleared from the database, as an alarm which occurs and vanishes within 10 seconds assumed to be highly uninformative for the operators. To avoid long-standing-alarms, alarms longer than two hours are also cleared, as these alarms considered to be either neglected by the operators or shelved or suppressed in the DCS system (but they are still visible in the historical database). After preprocessing the analysed dataset contains 18 production units, 139 units and 358 sensors/actuators.

In order to analyse and detect the extent to which individual events occur at each level of hierarchy, we have developed a sunburst plot visualization presented in Figure 5.7. The *sunburst plot* is a multilevel pie chart to visualize hierarchical data, depicted by concentric circles. The levels of hierarchy move outwards from the center, and a segment of the inner circles holds a hierarchical relationship to the segments of the outer circles which lie within the angular sweep of the parent segment. The inner small, the middle and the outer circles symbolize the ratio of alarms between the related production units, units and sensors/actuators, respectively. The labels in the circular sectors indicate the tag of the production units, units and process variables (sensors/actuators). Therefore, it is highly conspicuous that the process variable No. 1688 in the unit No. 467 located in the production unit No. 14 contributes significantly to the number of alarm counts. The sunburst plot is highly applicable for the illustration of the distribution of the recorded alarms on different levels of hierarchy and how critical the operation of production units and units. It is important to note that this is a very useful picture of the frequency of alarms and the expected loads of process operators.

The purpose of this work is, however, to go beyond this simple and useful data analysis to reveal the evolution of events in a temporal manner, the pattern of the spillover effect of process malfunctions. These results are presented in the next subsection.

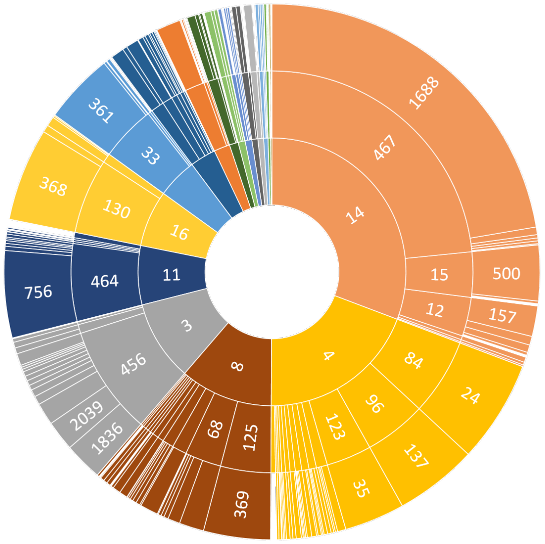


Figure 5.7: Sunburst plot of the alarm database to visualize the alarm count ratio of the levels of hierarchy.

Based on the results of former temporal analysis and by the application of the expert knowledge of process engineers, a time window beyond which no causal connection between the occurring events is presumed was defined. If the event occurrences are divided by a longer time period than the defined time window than we assume that these events are the part of a different *event trace*. Therefore we have segmented the occurring events into event traces, which we considered the evaluation path of a process malfunction. Applying a 60-second-long time window for segmentation, we have derived 10272 event traces with a minimal length of one event and a maximal length of 2454 events. The average number of events in an event trace as the function of the first production unit of the event trace is illustrated in Figure 5.8.

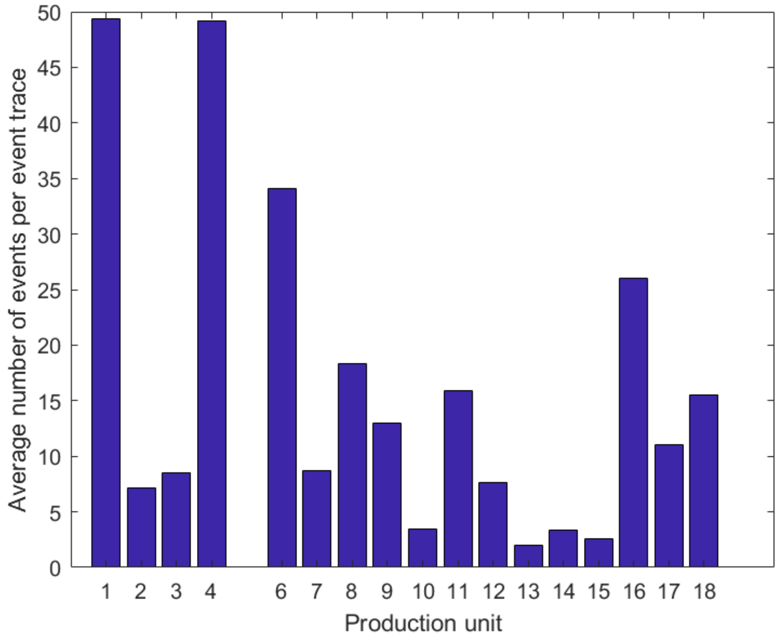


FIGURE 5.8: The average number of events in an event trace starting from the production unit indicated by axis x.

Although changes in the magnitude of production volume are rare in the oil industry, this may occur in flexible production plants. In such a case, the residence time of the equipment may change significantly, which may necessitate the use of different time window values for different modes of operation.

The defined event traces give the opportunity for the analysis of the evaluation path of a process malfunction, which is the primary purpose of our investigations. These results are summarised in Section 5.3.3.

### 5.3.2 Hierarchical alarm sequences of the analysed delayed coker plant

It must be highlighted that the exact values of alarm frequency are cannot be published, thus the presented values are normalized or transformed. However, it is an important question how the support threshold is determined on each level of hierarchy. It is advisable to determine this parameter using a bottom-up approach, since applying expert knowledge the minimal number of occurrences is known, that we want to analyse in case of each alarm. This information can be summed and converted to a higher level of hierarchy. For easier interpretation, I applied 100 as the minimal number of occurrences on every level of hierarchy to calculate the support threshold. Therefore the hierarchical multi-temporal mining algorithm is applied to reveal frequently occurring alarm patterns on each level of hierarchy with the calculated support thresholds, 0.01 confidence threshold and a 60-second-long time window.

The visualization of temporally connected process elements is suitable for the illustrative analysis of the spreading effect of a malfunction. To illustrate the proposed hierarchical approach, we created the drill-down graph presented in Figure 5.9. The heatmaps show the connections in decreasing order of levels of hierarchy from left to right. The values show the sum of support values of the sequences containing two elements ($k$ = 1-length sequences) with the neglection of temporal predicates. The strong self-pointing inner connections are well illustrated in all three heatmaps, as the main diagonals of the figures contain the highest support values. However, there are some nice examples of sequences propagating between process elements, e.g. the alarm sequences pointing to the production unit No. 4 from other production units showing the spillover effect of a malfunction.

The sequence-based representation of discrete events helps the analysis of hidden connections with the relevant probability measures, while the hierarchical information helps the incorporation of process relevant information.

An important element of the proposed methodology is the definition of process periods not containing alarms as events, therefore we completed the event traces with an extra event marking the normal operation, that is the start of the event trace. Analysing these sequences ensures the generation of sequences following

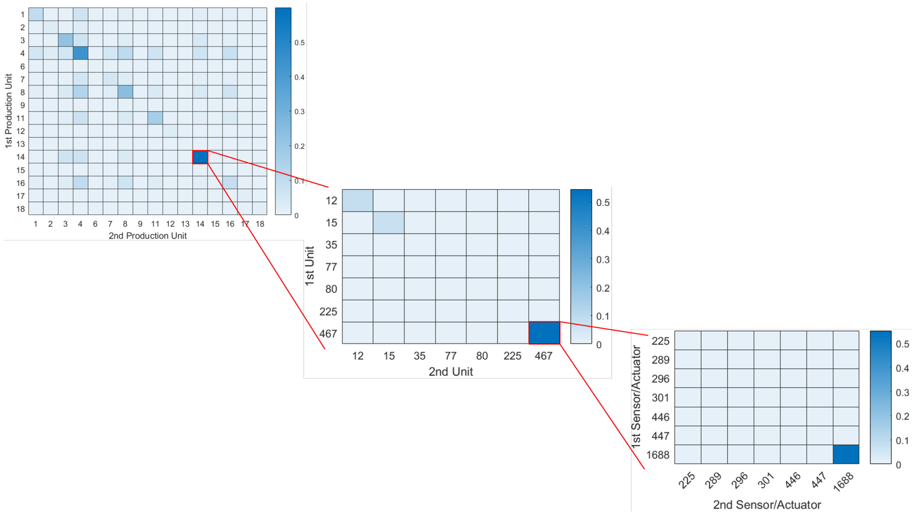


FIGURE 5.9: The heatmaps are showing the connections of process elements on different hierarchical levels. The values show the sum of support values between the specific process elements with the neglection of temporal predicates.

process malfunctions. In the following subsection, the novel methodology is introduced, developed for the analysis of the causal relationship between process elements in case of malfunctions on different levels of hierarchy.

### 5.3.3 Sequence based causality analysis

The domino or spillover effect of malfunctions is informatively represented by the generated sequences, while the quantitative analysis and the risk assessment is well-supported by the support values reflecting the probability of occurrence and the confidence values reflecting the probability of spillover. The support value can be easily interpreted as a normalized form of occurrence counts. However, the interpretation of confidence values can be harder. The confidence gives a measure of the probability that the given sequences will follow the pattern appointed by the sequence elements. Therefore, it describes that after the occurrence of the first element how sure we are in the second element and so on until the last element of the sequence.

However, it is hard to visualize such sequences and get a clear and illustrative overview of the results. To overcome this difficulty, a parallel coordinates-based visualization solution was developed, presented in Figure 5.10. In general parallel coordinates are an effective way of visualizing high-dimensional or multivariate datasets. The horizontal axis represents the sequence length, while the vertical axis shows the production unit in which the given event occurred. Therefore, we can imagine equally spaced copies of axis y in every element of axis x. A $k$-length sequence is represented by a polyline with vertices on the imaginary parallel axes and the position of the vertex on the $i$-th axes corresponds to the production unit in which the $i$-th event occurred. Using the implemented events indicating faultless production at the beginning of every event trace, we can generate sequences starting with these faultless periods. Therefore, we can determine the first production unit (most probably the production unit where the malfunction occurred), this first production unit is indicated by the colour of the polyline. The thickness of the edge between two vertices is determined by the sum of support values between the specific sequence elements with the neglection of temporal predicates, therefore the thicker the edge, the higher the support value is. The polylines in Figure 5.10 highlight that the sequences start from different production units with varying extents. Moreover, there are strong cross effects between production units and back and forth connections can be seen as well, an illustrative example is the high number of links between production units No. 14 and 3.

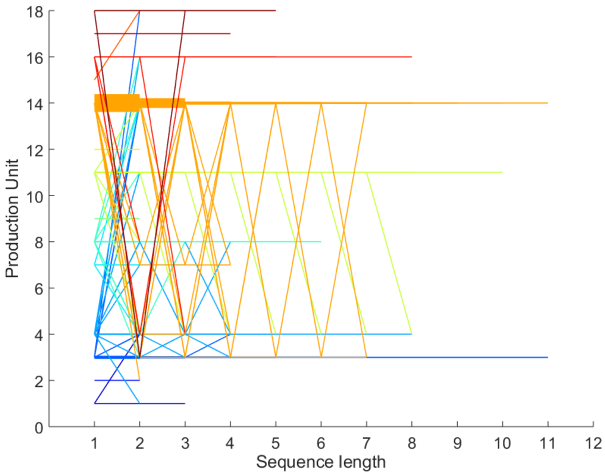


Figure 5.10: The parallel coordinates-based visualization of frequently occurring sequences. A sequence is represented by a polyline with vertices at the elements of the horizontal axis. The y-coordinate of the vertice indicates the production unit in which the given sequence element occurred. The colour of the polyline is determined by the production unit where the first event of the sequence occurred, while the thickness of the edges is proportional to their support value summarised for all the occurring temporal predicates. The spillover effect of malfunctions between production units is well-observable.

An important question is the causal dependency of production units, therefore the orientation of causal relationships. To measure this extent, I have borrowed the dependency measure applied by the heuristic process mining algorithms. The dependency measure is applicable for longer sequences as well, but we applied it for the characterization of sequences containing two events. The left side of Figure 5.11 shows the heatmap of the dependency measures. The causal dependency of Production unit No. 3 from No. 8, No. 2 from No. 14, No. 14 from No. 11 and No. 18 from No. 15 are well visible from the heatmap.

It is interesting to examine the confidence of transitions between states, which gives the opportunity to detect the confidence of a malfunction will spread to another production unit as well. We should note that in the case of sequences containing only two states, the confidence of the sequence is equal to the confidence of transition between states. The right side of Figure 5.11 shows the calculated directionality measures of the state pairs giving information of the confidence of the direction of the causal link.

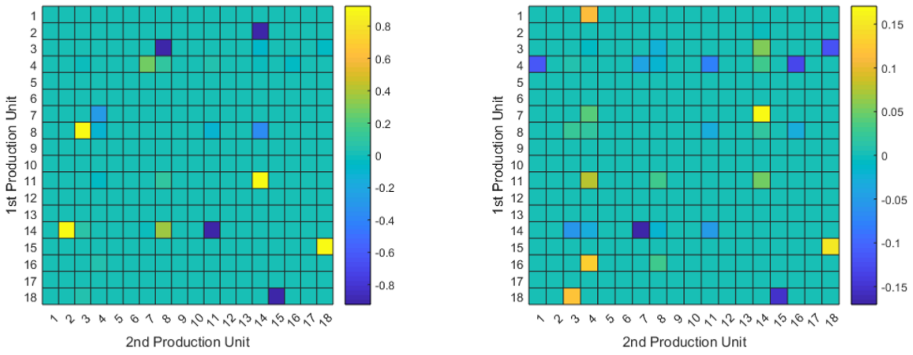


FIGURE 5.11: The left graph of the figure shows the heatmap of the dependency values of transition between states. The right side of the figure illustrates the directionality values of state transitions, showing how confident we are in the direction of causal connections. The causal relationship of production units is well-traceable using the presented measure and its visualization.

Another illustrative visualization of transition between states is the directed network-based visualization presented in Figure 5.12. The nodes show the production units, while the edges and the related weights represent the orientation of the links and the confidence of transition between states, respectively. The confidence values are summarised for all the occurring temporal predicates between the related states. This network-based representation gives the opportunity to apply the centrality measures of network theory. We have found that the ending and central production units of the sequences are characterized by high PageRank and betweenness measures, respectively. The highest PageRank values correspond to the production units No. 2 (0.15), 4 (0.11) and 3 (0.09), while the highest betweenness measures correspond to the production units No. 4 (35.17), 3 (23.00) and 14 (15.17). Comparing chemical technologies to a network of connected units, this result can be explained by the definition of PageRank and betweeness measures: PageRank indicates the "important" nodes of a network, where numerous other nodes are linked, therefore, these are the production units receiving numerous alarm sequences. The betweenness centrality describes the number of shortest paths that pass through the node, therefore, it identifies the central production units that the alarm sequences pass through.

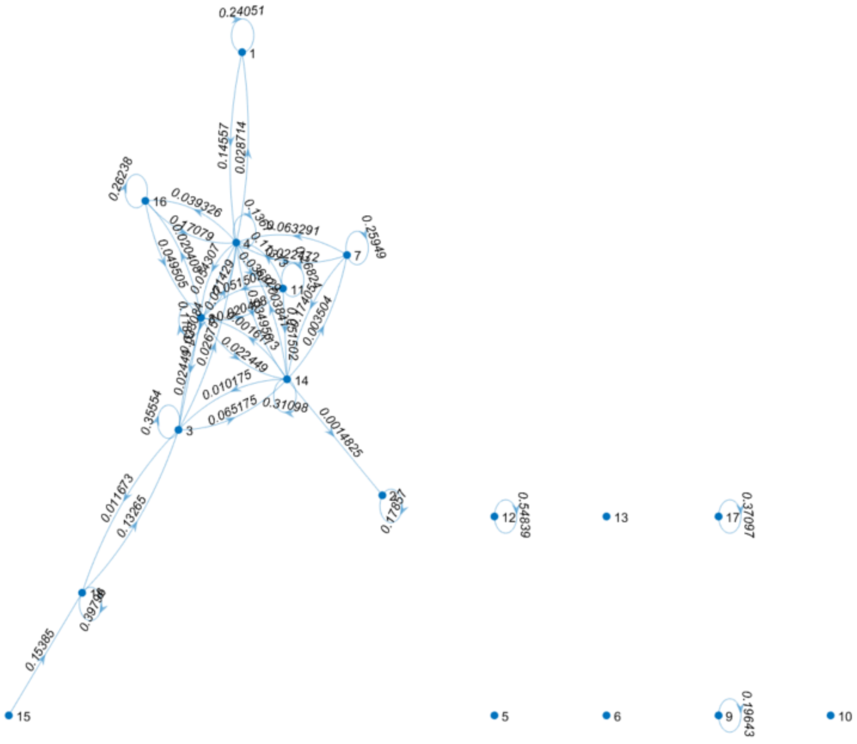


Figure 5.12: Directed-network based representation of spillover effect of malfunction between production units. The nodes, links and weights represent the production units, the orientation of their temporal connection and the confidence of transition between states, respectively.

The most important advantage of hierarchical sequence-based approach for the generation of alarm connections is that it gives the opportunity to analyse lower levels of hierarchy and reveal how the effect of a malfunction spread from one production unit of the process to another and what units or sensors/actuators are involved. To demonstrate this, I have examined the sequences changing production units on the level of units and the sequences changing units on the level of sensors/actuators. The connections and their confidence of transitions are presented in Table 5.2 and Table 5.3 for units involved in the changing of production units and sensors/actuators involved in the changing of units. The low number of process elements suggests that in a well-designed control system the effect of a malfunction cannot spread over the whole plant, but should stay near its origin.

Table 5.2: The units involved in the spreading of the effect of malfunctions between production units and the related confidence values of transition.

| $1^{st}$ Unit No. | $2^{nd}$ Unit No. | Confidence of transition [%] |
|---|---|---|
| 460 | 467 | 40.98 |
| 460 | 467 | 34.43 |
| 464 | 96 | 10.34 |
| 130 | 96 | 23.63 |
| 275 | 456 | 20.31 |

Table 5.3: The sensors/actuators involved in the spreading of the effect of malfunctions between units and the related confidence values of transition.

| $1^{st}$ sensor/actuator No. | $2^{nd}$ sensor/actuator No. | Confidence of transition [%] |
|---|---|---|
| 422 | 420 | 61.54 |
| 486 | 1688 | 11.22 |
| 486 | 1688 | 11.71 |
| 490 | 1688 | 40.98 |
| 490 | 1688 | 34.43 |
| 756 | 137 | 13.64 |
| 368 | 137 | 23.63 |
| 591 | 664 | 84.21 |
| 1720 | 1722 | 67.19 |

## 5.4 Discussion of results

The analysis of a complete production plant is a high complexity task, since the high number of process variables significantly increases the probability of the co-occurrence of unconnected events of the process. The co-occurrence of these events hinders the generation of relevant and informative event sequences. The high number of possibilities also challenges the frequent pattern mining algorithms, raising the need for a new, more fine-tuned solution.

I have recognised that the problem is particularly significant if chattering alarms are present in the process. Taking into consideration that our work aims to automatically reveal the internal causal connections of the process and to provide informative sequences to help the work of the operators, a methodology for removing the irrelevant chattering alarms was developed.

The proposed frequent pattern mining algorithm follows a top-down principle, i.e., it moves down from the higher levels of hierarchy to the bottom ones, analysing the sequences of alarms at the given level of hierarchy. The core concept of the proposed hierarchical sequence mining approach is that we mine the sequences at lower levels of hierarchy based on the sequences revealed on a higher hierarchical level (and possibly validated by expert knowledge), preventing unconnected units at a lower hierarchical level to be part of the same sequence. In the top-down way constrained search space, informative sequences can be revealed more efficiently. The method allows the identification of process elements initiating the alarm sequences and the analysis of domino-like spillover effect of malfunctions based on the evaluation pattern of alarm sequences. We have developed a parallel coordinate-based visualization approach for the investigation of this effect that well-illustrates how alarm signals from a given process element spread over the process.

The probability of the spreading of the effect of malfunction over multiple production units can be nicely characterized by the confidence of the sequences. We have recognised that the connection between two process elements can be a continuous interaction or a one-way dependency. To qualify the orientation of these connections, we have utilized the dependency measure from the field of process mining, and then we have developed a measure based on the difference of confidences, called the directionality measure. Using connections having significant confidences, a network can be defined and analysed to identify critical process elements in the spreading of the effect of errors. We investigated the applicability of the network's centrality measures and found that the betweenness and PageRank measures are utilizable for the identification of the central and ending elements of the potential alarm cascades, respectively. This approach provides an opportunity to coordinate the operator tasks, i.e., to group the units to operators based on their exposure during operation malfunctions.

Analysing a lower level of hierarchy helps the detection of production units, units and control circuits that play a key role in the evaluation of the domino effect-like spreading of the effect of malfunctions. The extracted knowledge can be utilized for the prioritization of alarms and sequences. The revealed information can be validated using the expert knowledge of the technology.

It is an important feature of our work that it relies solely on the log file of the alarm and event log databases and on the hierarchical classification of these events on the level of production units, units and sensors/actuators. The method is based on the qualification of causal relationships in accordance to their order of occurrence (in line with other entropy transfer-based solutions). In the light of this, the time window parameter of the proposed algorithm is of critical importance, since the methodology is based on the assumption that in the knowledge of the technology and the occurring events, a time window can be defined above which no causal relationship is assumed between two events. The proposed approach is very effective giving the opportunity for the definition of event traces and to speed up our algorithm by searching only the sequences starting from the faultless production. In our future work, we would like to examine the effect of this time window parameter and the distribution of time periods between consecutive alarms.

## 5.5 Chapter summary

Alarm systems are crucial parts of industrial processes for managing hazardous situations. However, due to the negligible cost of alarm definition in modern DCS systems, the increasing number of uninformative and redundant alarm signals significantly hinders the work of the operators. Frequent pattern mining-based advanced alarm management is a promising approach for the exploration of redundant and co-occurring signals. To make pattern mining algorithms applicable for large industrial scale datasets, we have introduced a novel process hierarchy-based solution. The methodology is based on the utilization of the principles of decomposition and coordination and of the hierarchical structure of process variables, units and production units, preventing the generation of frequent, but uninformative sequences. The revealed hierarchical sequences give the opportunity to identify alarm sequences crossing more critical units and production units of the process and the proposed quantitative measures help the qualification of the frequency of alarm occurrences, their severity from the view of operability and their possible spread over effect and orientation. The network-based representation of alarm cascade evaluation paths is not just highly informative for process experts but gives the opportunity for the application of network theory-based centrality measures. The applied betweenness and PageRank measures show the central and ending elements of sequences and help the identification of critical process elements from the view of operability. These results can be applied to coordinate operator workloads and group the units to operators based on their exposure during operation malfunctions. Analysing the lower levels of hierarchy gives an insight on the critical units and sensors/actuators in the propagation of malfunctions by showing the elements involved in the spreading of the effect of malfunctions over production units or units.

The application of frequent sequence mining confirms my 1.1 and 1.2 thesis findings, while the incorporation of hierarchical information to the analysis provides the basis of my 3.1 thesis finding. The thesis findings are summerized in Section 8.

# Chapter 6

# Sequence-based fault detection and isolation

The identification of process faults is a complex and challenging task due to the high amount of alarms and warnings of control systems. To extract information about the relationships between these discrete events, multi-temporal sequences of alarm and warning signals are utilized as inputs of a recurrent neural network (RNN) based classifier and the network is visualized by principal component analysis. The similarity of the events and their applicability in fault isolation is evaluated based on the linear embedding layer of the network, which maps the input signals into a continuous-valued vector space. The method is demonstrated on a simulated vinyl acetate production technology. The results illustrate that with the application of RNN based sequence learning not only accurate fault classification solutions can be developed, but the visualization of the model can give useful hints for hazard analysis.

## 6.1 Introduction

Chemometric models are widely applied to fault detection and isolation of chemical processes [135]. Although most of these models utilize continuous multivariate data, plant operators are required to make decisions based on hundreds of discrete data generated by the control systems as warning and alarm signals. In a complex system faults may co-occur in several states, [136], and can generate long sequences of warning and alarm signals, therefore giving effective responses to abnormal situations is a challenging task for even the well-trained operators. Intelligent fault diagnostics, therefore, requires the analysis of temporal relationships of discrete events. To meet this requirement, we propose a sequence-based fault classification algorithm that utilizes the high amount of unexploited event type data of alarm systems.

Discrete event-based fault diagnosis is an important area of research, as this type of information like alarms and warnings frequently occur in the process industry. According to the Engineering Equipment and Materials User's Association (EEMUA), the purpose of an alarm system is to redirect the operator's attention towards plant conditions requiring timely assessment or action [3]. Therefore, a properly designed and operated alarm system helps the operator to keep the processes in the normal operation range by indicating the presence of abnormal situations. Blanke *et al.* give an extensive overview of fault diagnosis [137], while Zaytoon *et al.* focuses on the diagnosis methods of discrete event systems [138]. The central concepts of the diagnosability and fault diagnosis of discrete event systems were defined by Sampath *et al.* [139], [140].

When we build data-driven models for fault detection and isolation purposes, we not only focus on the prediction accuracy, but we also would like to understand the mechanism of the faults by unfolding the relationships between the faults and the process-variables [141]. When events of different states occur at the same time, they can be visualized with the use of a time series cross-sectional data matrix, as it is presented by Chen *et al.* [142]. Correlated events can be visualized with the utilization of a Hinton diagram of joint distributions [143]. The recently developed high-density alarm plot (HDAP) chart highlights top alarms over a given period, and the alarm similarity colour map (ASCM) explores the related and redundant alarms [20]. The methods are used for the detection of correlated alarms in ref. [37], while the application of ASCM and correlation colour maps were also reported in ref. [144]. From the tools of chemometrics, dendrograms were employed in [144] and [142].

The key idea demonstrated in the present chapter is to develop a supervised visualization algorithm to evaluate the similarities of the alarms from the viewpoint of the faults. We assume that, as in the case of natural language processing applications of deep learning, the visualization of the network supports the understanding of the long- and short-term dependencies of the alarm signals and the faults. To test the proposed methodology, a sequential data based classifier was built, applying deep learning solutions.

The complexity of the problems and size of the available datasets tend to be bigger and bigger, resulting in the increased application of deep learning solutions in engineering [145], chemistry [146], computational biology [147], process engineering [148], machine health monitoring [149], anomaly detection [150] and fault detection and isolation [151, 152]. For a detailed description of artificial neural network-based approaches including the distinguish of classical (shallow) neural networks, and deep learning solutions see ref. [153]. From the wide range of models, recurrent neural networks (RNNs) are applied [154] using long short-term memory (LSTM) units [155]. The proposed model uses an embedding layer, a layer with linear transformations, to map the one-hot encoded events into a continuous-valued vector space. Using such embedding is a state-of-the-art approach to sentiment analysis of texts. The main benefit of this linear mapping is that the analysis of the vector space can be used to study the contextual meaning of the words [156], [157]. Based on this analogy, we assume that similar warnings and alarm signals will be close to each other in this embedding space [158]. We apply a linear embedding, assuming that the weights of the similar events will be correlated, so principal component analysis (PCA) can be used to visualize the hidden structure of the events and evaluate the significance of these signals.

To demonstrate the applicability of the proposed approach, the extended simulator of a vinyl acetate production technology [85] was applied similarly to the previous chapters. The applied 11 malfunctions were chosen based on process relevant knowledge, as we wanted to implement malfunctions with a significant effect on the operation in various locations of the process. Using this simulator, we can record the dynamic characteristic following these faults and we can create the log file of the occurring alarms and warnings.

The roadmap of the chapter is as follows. In Section 6.2.1, we define the input of the classifiers as sequences of the temporal relationships of the events. Section 6.2.2 presents the classification task, and Section 6.2.3 describes why we analyse the embedding layer of the model. Although mainly the prediction accuracy is in the focus of the application of deep learning models, we study the applicability of PCA to extract information related to the hidden structure of the problem in Section 6.2.3. We introduce the case study in Section 6.3.1. The results are discussed in Sections 6.3.2-6.3.4.

## 6.2 Fault classification and visualization of process alarms

### 6.2.1 Formulation of the event sequence based fault classifier

Our key idea is that the sequences of process alarms and warnings contain enough information about the technology to serve as an input of a classifier designed to estimate the $y_k = \{c_1, \dots, c_{n_c}\}$ class label of the faults, where $k = 1, \dots, N$ and represents the currently examined sequence of the process.

$$\hat{y}_k = f\left(\boldsymbol{\Phi}_k\right) \tag{6.1}$$

The basic concept of sequence-based representation was already described in Section 4. Therefore, for the description of the structure of the alarm & event-log database, the definition of the states and events of a technology, moreover, their mathematical formulation and the temporally connected events together with the type of temporal predicates, please refer to Section 4.

To utilize the sequences of the symbols as inputs of the neural network, we encode the symbols of the states $S = \{s_1, s_2, ..., s_{n_S}\}$ and the temporal predicates R=$\{E,\ B,\ D,\ O\}$, into vectors of numerical values. From a technical view, first we encode every element in the sequences (event and temporal predicates as well) to a numerical form, and this numerical form is transformed to the sequence of one-hot encoded vectors.

The one-hot encoding is based on binary vectors, $\mathbf{oh}_k^t$ where only one bit related to the encoded signal is fired among the $n_o = n_S + n_R$ bits, where $n_S$ represents the number of states and $n_R$ stands for the number of different temporal predicates.

The embedding layer is a linear transformation, which transforms the one-hot encoded vector of each sequence element into a vector with the specified dimension ($n_e$) with continuous values.

$$\mathbf{x}_k^t = \mathbf{W}\,\mathbf{oh}_k^t \tag{6.2}$$

Figure 6.1 gives a simple, didactic example of the formulation of the input dataset of the embedding layer. Suppose that the $k$-th analysed sequence is the simple overlapping relationship of the events $e_1$ and $e_2$ illustrated in Figure 4.3 in Section 4. The states represented in these events and the overlapping temporal relationship are transformed into a one-hot encoded vector with the dimension of $n_o = 2+1 = 3$, supposing we have no other states or temporal predicates in our examined dataset, therefore $n_S = 2$ and $n_R = 1$. Each of these elements will fire one specific bit of the one-hot encoded vector forming $\mathbf{oh}_k^1$, $\mathbf{oh}_k^2$ and $\mathbf{oh}_k^3$ respectively. We transform these one-hot encoded vectors of each sequence element to vectors of continuous values with a specified dimension ($n_e$, the embedding dimension) in the embedding layer. These vectors form the input variables of the deep learning layer.

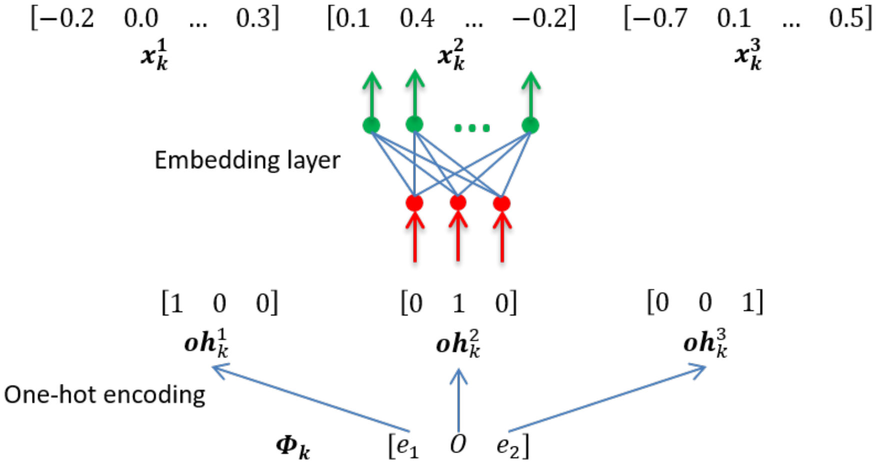


FIGURE 6.1: A simple and didactic example for the formulation of the input variables of the deep learning layer. The elements of the $\mathbf{\Phi}_k$ sequence are transformed to the form of one-hot encoded vectors, with the dimension of the number of the type of sequence elements in the examined database. We transform these one-hot encoded vectors to vectors of continuous values with the help of the embedding layer.

Therefore the deep learning based classifier can be formulated as

$$\hat{y}_k = f\left(\mathbf{x}_k\right) \tag{6.3}$$

where $\mathbf{x}_k$ represents the sequence of $\mathbf{x}_k^i$ $(i = 1...T)$ vectors, the continuous-valued representations of the sequence elements (events and their temporal predicates) calculated by the linear embedding layer and $\hat{y}_k$ stands for the estimated class label.

We train a neural classifier to calculate the $P(y_k = c_j|\mathbf{x}_k)$ fault probabilities and assign the class label that has the highest probability (see Figure 6.2).

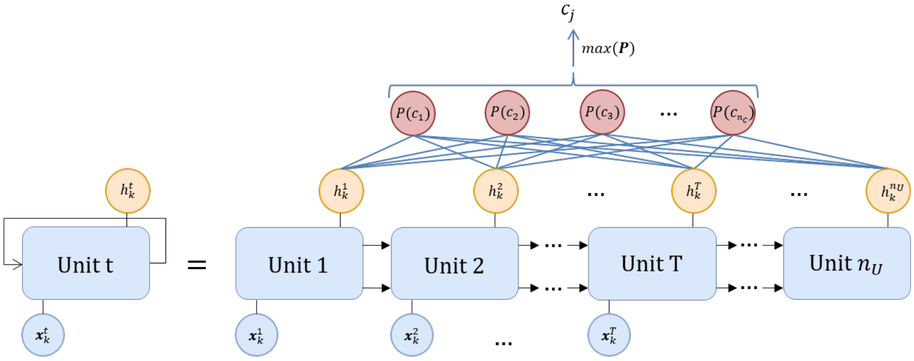


FIGURE 6.2: The structure of the long short-term memory unit-based neural classifier. The input variables of the layer of the long short-term memory units, represented by the $\mathbf{x}_k^i$ $(i = 1...T)$ vectors, are the continuous-valued representations of the sequence elements calculated by the linear embedding layer. We transform the output activity values of this long short-term memory (LSTM) unit-layer $(\mathbf{h}_k)$ to probability values by a linear layer, and we accept the class with the highest probability as the root cause of the sequence.

$$\hat{y}_k = \max_j P(y_k = c_j|\mathbf{x}_k) \tag{6.4}$$

The recurrent neural network maps the $\mathbf{x}_k$ input sequence into a $\mathbf{h}_k$ sequence of real values. This mapped sequence of "hidden variables" is represented as a $\mathbf{h}_k$ vector of the activities of the LSTM units, $\mathbf{h}_k = [h_k^1, \ldots, h_k^{n_U}]$ and used to calculate the fault probabilities by the softmax activation function (where $n_U$ is the number of LSTM units),

$$P(y_k = c_j|\mathbf{x}_k) = P(y_k = c_j|\mathbf{h}_k) = \frac{exp\left((\mathbf{h}_k)^T\mathbf{w}_{s,j} + b_j\right)}{\sum_{j=1}^{N_c} exp\left((\mathbf{h}_k)^T\mathbf{w}_{s,j} + b_j\right)} \tag{6.5}$$

where $\mathbf{w}_{s,j}$ represents the $j$-th column vector of the $\mathbf{W}_s$ weight matrix of the output layer of the network and $b_j$ represents the bias. The denominator of Equation 6.5 is the sum of the elements with different $j$ indices, therefore constant and the resulting class label is proportional to the nominator.

$$\hat{y}_k = \max_j P(y_k = c_j|\mathbf{x}_k) \propto \max_j \left(exp\left((\mathbf{h}_k)^T \mathbf{w}_{s,j} + b_j\right)\right) \tag{6.6}$$

Figure 6.2 highlights that the sequences are processed by a deep recurrent neural network, where the network is unrolled $n_U$ times and each of the first $T$ units process a single sequence element represented as a continuous-valued vector with dimension $n_e$. Although, the units from $T$ to $n_U$ seem to have no significant effect on the classification task, the core principle of the LSTM units is the handling of long-term dependencies and these added units can improve the accuracy of the classification.

In the following subsection, I present how the hidden layer ensures the efficient calculation of the $\mathbf{h}_k$ vector that is informative to build the classifier based on sequential data.

### 6.2.2 Recurrent neural network layer with long short-term memory units

Recurrent neural networks (RNNs) are designed to capture time-dependency in sequential data [154]. These models were popular after the introduction of RNNs with LSTM (long short-term memory) units [155], proposed to overcome the difficulties of handling long-term dependency and vanishing gradient [159].

The LSTM layer maps the $\mathbf{x}_k$ sequence into $\mathbf{h}_k$, the vector of the activities, so it has equal or more units than the sum of the number of events and the corresponding temporal predicates between them in the input sequence. Therefore, sequences containing more events than the specified $T$ value are truncated, and the length of the input sequence is $2 \times T - 1$ (the length of a vector is equal to the number of elements in it, here $T$, and we need to count the corresponding temporal predicates as well). This is due to the fact that a simple sequence $e_1 \overset{R}{\Rightarrow} e_2$ containing only two process states, is consist of $2 \times 2 - 1 = 3$ sequence elements including the temporal relationship. Similarly, the sequence of $e_1 \overset{R}{\Rightarrow} e_2 \overset{R}{\Rightarrow} e_3$ contains only three process states but the number of elements in the sequence is equal to $2 \times 3 - 1 = 5$ pieces. Therefore the "-1" part is implemented to take into consideration the missing temporal predicate after the last process state of the sequence.

Figure 6.3 shows the structure of the LSTM unit. The key feature of the model is that all LSTM units have a cell state ($C_k^t$) [160] that can be used to forward information to the next unit. Therefore, an LSTM unit interacts with its neighbouring cells by gates by either adding or removing information to this memory flow.

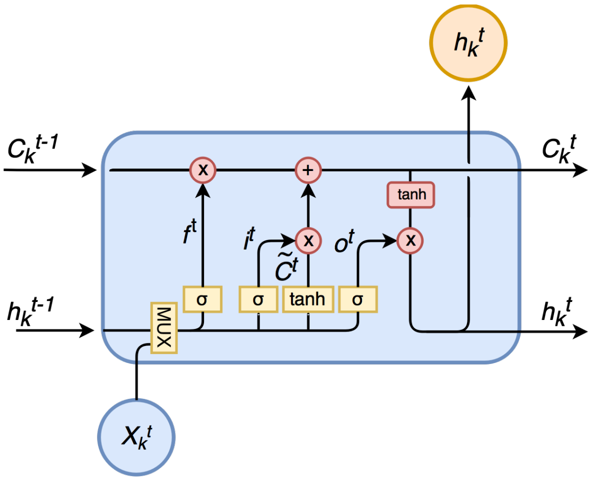


FIGURE 6.3: Structure of a single long short-term memory unit. The input sequence element ($\mathbf{x}_k^t$), and the activity ($h_k^{t-1}$) and cell state ($C_k^{t-1}$) values of the previous long short-term memory unit are modified with use of the sigmoid ($\sigma$) and hyperbolic tangent ($tanh$) functions of the forget ($f_k^t$), input ($i_k^t$) and output ($o_k^t$) gates. The calculated activity ($h_k^t$) and cell state ($C_k^t$) values are transfered to the next long short-term memory unit, while the activity value is directly output as well. The MUX box in the figure indicates the forming of one signal from the $h_k^{t-1}$ and $\mathbf{x}_k^t$ expressions as in $[h_k^{t-1}, \mathbf{x}_k^t]$.

The LSTM unit receives the activation of the previous cell $h_k^{t-1}$, the $\mathbf{x}_k^t$ input that relates to the $t$th element of the sequence, and the $C_k^{t-1}$ cell state of the previous unit. The forget gate $f_k^t$ determines how much information of the earlier units should be kept.

$$f_k^t = \sigma(\mathbf{W}_f[h_k^{t-1}, \mathbf{x}_k^t] + b_f) \tag{6.7}$$

$b$ represents the bias vector of the neurons (and in all of the following equations as well), and $\sigma$ represents the applied sigmoid function.

The state and the activity of the unit will be updated using the $\mathbf{x}_k^t$ and the preceding cell activation $h_k^{t-1}$. This is realised with the sigmoid function of the input gate. The whole process is illustrated in Figure 6.3.

$$\begin{aligned} i_k^t = &\quad \sigma(\mathbf{W}_i[h_k^{t-1}, \mathbf{x}_k^t] + b_i) \qquad (6.8) \\ \tilde{C}_k^t = &\quad \tanh(\mathbf{W}_c[h_k^{t-1}, \mathbf{x}_k^t] + b_c) \qquad (6.9) \end{aligned}$$

The LSTM unit updates its old cell-state $C_k^{t-1}$ using the forget-gate $f_k^t$ and the filtered input gate $i_k^t$:

$$C_k^t = f_k^t C_k^{t-1} + i_k^t \tilde{C}_k^t \qquad (6.10)$$

The activity of the LSTM unit is calculated based on the cell-state and the output gate signals:

$$\begin{aligned} o_k^t = &\quad \sigma(\mathbf{W}_o[h_k^{t-1}, \mathbf{x}_k^t] + b_o) \qquad (6.11) \\ h_k^t = &\quad o_k^t \tanh(C_k^t) \qquad (6.12) \end{aligned}$$

### 6.2.3 Embedding layer based analysis of the alarms and the analysis of node activities

The present work is mainly motivated by the purpose to analyse the similarities of alarm and warning signals. Therefore an embedding layer is introduced to the network, as the concept of the embedding layer is to map the sequence elements into a continuous vector space, which we apply for the visualization of the events. First,every character of the sequence is transformed into one-hot encoded vectors. Then with the linear transformation of the embedding layer, these one-hot encoded vectors of each sequence element are transformed into a vector of continuous values as it is presented in Equation 6.2. According to the presented equation, this linear transformation means a simple matrix multiplication in practice, where the dimension of the resulted vector is $n_e$, the embedding dimension (chosen according to preferences, see Figure 6.8) with continuous values between -1 and 1.

The weights of the embedding layer are trained simultaneously with the LSTM units, therefore the resulted weight matrix, $\mathbf{W}$, stores information related to the contextual connection between the symbols and the classification problem.

$\mathbf{W}$ is multiplied by one-hot vectors; therefore every row of the matrix represents a given symbol, $\mathbf{w}_i$. The similarities $(sim)$ of the alarms $(s_i, s_j)$ can be evaluated based on the Euclidean distances of these vectors $(d\left(\mathbf{w}_i, \mathbf{w}_j\right))$:

$$sim(s_i, s_j) = 1 - \frac{d\left(\mathbf{w}_i, \mathbf{w}_j\right)}{d_{max}} \tag{6.13}$$

where $d_{max}$ represents the maximum Euclidean distance between the rows of the $\mathbf{W}$ matrix and $\mathbf{w}_i$ and $\mathbf{w}_j$ represents the $i$-th and $j$-th rows of the $\mathbf{W}$ matrix respectively.

The resultant pairwise similarities are used to generate a dendrogram to form clusters of the alarm signals.

When the dimensionality of the embedding layer is two, it can be directly used to visualize the relative positions of the alarms. When $n_e > 2$ we can use one of the several data visualization techniques of chemometrics. In the present work, we utilized principal component analysis (PCA) since it can also visualize the correlation of the columns of the $\mathbf{W}$ matrix, which information is useful for the selection of the proper size of the embedding dimension.

The output layer of the network is based on the linear combination of node activities as it was described in Equations 6.5-6.6.

To get an insight into the classification problem we can perform a principal component analysis of the $\mathbf{H} = [h_1, \ldots, h_N]^T$ matrix of the activities, where, $\mathbf{h}_k = [h_k^1, \ldots, h_k^{n_U}]$, $k = 1, \ldots, N$ and map the sequences into a two dimensional space. When these sequences are labeled, the similarity of the faults can be revealed from the resulted plot, similarly to Figure 6.13.

## 6.3 A case study

Previously, a well-documented simulation study was developed by the extension of an existing VAc technology simulator to provide a reproducible benchmark problem of alarm based fault classification. In the following, the generated log files are described. The definition of the fault classification problem will be followed by the details of the RNN and the experiments for the determination of the optimal model structure. Finally, based on the resulting model the alarms and the faults are visualized by the principal component analysis of the embedding layer and the node activities. The discussion of the results will illustrate that the information generated by the proposed methodology can be useful in the risk analysis of complex processes.

### 6.3.1 Fault classification problem of the vinyl acetate process

Modern chemical plants often have historical log files of incoming alarms and warnings, and these log files can be structured to build training data for fault classifiers. Historical process data can be enriched or replaced by events generated by simulators since most of the advanced process technologies are also supported by operator training systems or other model-based solutions. In the present work, the benchmark process simulator of a vinyl acetate process is applied, its description is presented in the Appendix in Chapter 9.

To test our methodology, we generated a database of 200 different event sequences. The inserted faults were related to the malfunction of the controller or manipulator, as the manipulated value of the process variable remained constant for a specified time. Table 9.1 in the Appendix (Chapter 9) shows the values of the manipulators in case of faults.

A 100-minute time window was used to utilize events that we consider as direct consequences of the malfunction. The threshold values of normal operating conditions for each measured process variables were determined based on the analysis of normal operation.

To illustrate the information content of discrete events we generated three different datasets and tested their applicability in the solution of the classification problem.

- *Dataset A*: Signals related to low and high alarms (two states / process variable)
- *Dataset B*: Signals related to low and high alarms and warnings (four states / process variable)

Table 6.1: The digit added to the number of variable indicating the type of event

| Type of event | Added digit |
|---|---|
| Low alarm | 1 |
| Low warning | 2 |
| Target operating range | 3 |
| High warning | 4 |
| High alarm | 5 |

- *Dataset C*: Identical to B, but the normal operation is also defined as an event (five states / process variable)

To examine the information content of temporal relationships between events we generated two cases: (1) in the first we include the temporal relationships and (2) in the second we neglect them. Therefore we generated six datasets labeled as *Dataset A/1*, ..., *Dataset C/2* respectively.

The horizontal axis of Figure 6.4 shows the number of sequences that can follow the given malfunction, while the vertical axis shows the maximal length of these sequences. Figure 6.4 illustrates the faults based on the length of the longest sequence that follows the given fault and as the number of different sequences that can follow the given fault (the size of the sequence is equal to the number of events in it, we do not count the temporal relationships). The core principle of the figure is that faults with fewer types and shorter length of characteristic sequences can usually be treated as trivial by the process experts. Considering this the more complicated sequences are in the upper right part of the graphs. The information content of the databases increases from *Dataset A* to *Dataset C*, as the number and size of the sequences show an increasing trend.

The numbering of the different alarm tags follows an ascending order from 1 to 27 according to Table 10.2 (Acronyms). During the generation of discrete events, we added an extra digit to the end of each variable tag indicating the type of the event happened on that given variable as it can be seen in Table 6.1. Of course from *Dataset A* the events of warnings and the event of the target operating range are missing (therefore the last digit cannot be 2, 3 or 4), while from *Dataset B* the event of the target operating range is missing (consequently the last digit cannot be 3).

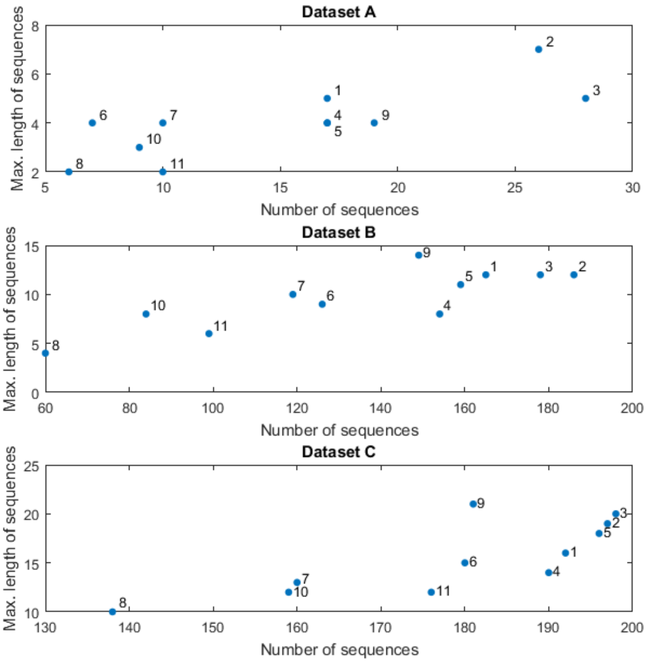


Figure 6.4: The characterization of faults based on the number and maximal length of the following sequences. The identification and classification of malfunctions that generate longer and more type of sequences of events are most likely to be problematic. The length of the sequence is equal to the number of events in it, and we do not count the temporal predicates.

Therefore, Table 9.1 in the Appendix (Chapter 9) shows the causes why the different events on the variables presented in Table 10.2 occur (Acronyms), as events on the controlled variables can be considered as the effects of malfunctions.

### 6.3.2 Application of the proposed recurrent neural network for fault classification

The implementation of the simulator and the data preprocessing was carried out in MATLAB environment. The implementation of the structure of the deep neural network and the training of it was carried out in Python applying Keras and using Tensorflow as backend. We trained the model using a Nvidia GeForce GTX 1060 6GB GPU with the application of CUDA. During the testing of the different model structures 7-fold cross-validation was applied and evaluated, the number of epochs was set to 500, with 512 as batch size.

#### 6.3.2.1 The applicability of different datasets

Figure 6.4 highlights the information content of datasets. The number of sequences that characterise a given fault, their length, and the presence of temporal relationships can all influence the effectiveness of the proposed LSTM network. To determine the most appropriate set of symbols, the efficiency of networks was tested under uniform conditions, with 11 LSTM units and five events in a sequence (longer sequences are truncated) and with four as the dimension of embedding. Firstly, datasets including temporal predicates were used. According to Figure 6.5, *Dataset B/1* is the most applicable for further investigations, with approximately 91.2 % of average accuracy (correct classification rate). We can conclude that the incorporation of warnings can improve the effectiveness of the proposed methodology, but the normal operating range as an event showed a decreased correct classification rate. We proved the effect of different datasets by statistical variance analysis (one-way ANOVA) and found a very low significance value ($p = 2.2E - 05$), therefore we reject the null hypothesis, that the type of the dataset has no significant effect on the correct classification rate.

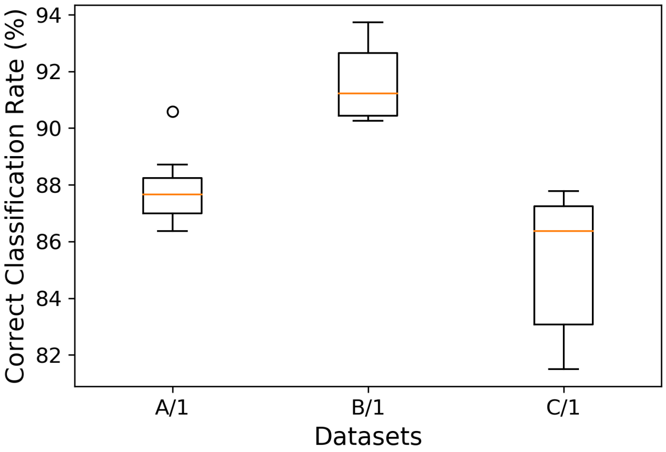


FIGURE 6.5: The effect of the information content of datasets on the effectiveness of the proposed network with 11 long short-term memory units, five events in a sequence and four as embedding dimension, datasets with temporal predicates served as the basis of analysis. According to the results, *Dataset B/1* is the most applicable for further investigations, as it shows the highest correct classification rate. Therefore the characterization of the variables with alarm and warning signals showed improved accuracy comparing to the result of *Dataset A/1*, with only the alarm signals, however the including of target operating ranges as events showed decreased correct classification rate.

We studied the effect of the number of events (i.e. the length of the sequence ($T$), the events after $T$ are truncated) and the incorporation of temporal predicates for *Dataset B*, which showed the highest correct classification rate in the previous analysis. According to Figure 6.6, the incorporation of more than three events indicated no improvement in the performance of the classifier, and the temporal predicates do not significantly influence the results. Two-way ANOVA was applied for the determination of differences in performance, but the analysis indicated that the including of temporal predicates or the application of more events does not result in better performance, as can be seen in Table 6.2. However standard deviations are not within 5%, in the case of the number of events it is very close to it. The good correct classification rate after only a few events shows that a well-trained neural classifier can classify a fault after only a few alarms, suggesting promising industrial application possibilities in the future. To reduce model complexity we applied four events and the dataset without temporal predicates in the following investigations.

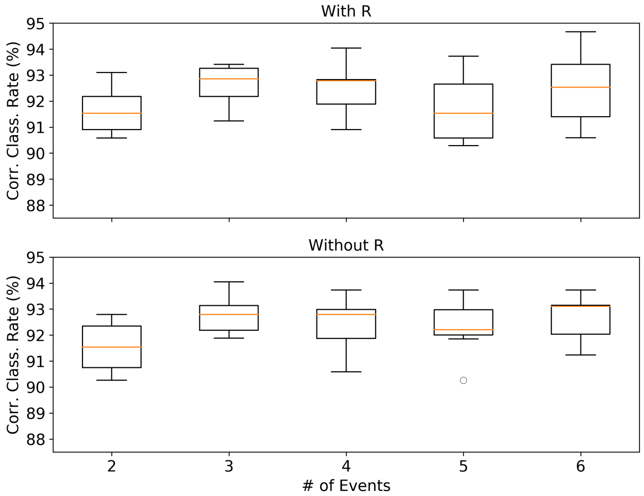


FIGURE 6.6: The effect of the number of events with and without the including of temporal predicates. The sequences longer than the specified number of units are truncated. According to the applied two-way ANOVA the application of more events nor the temporal predicates result in better correct classification rate neither. In the following investigations we applied four events without the including of the temporal relationships.

Figure 6.7 illustrates the effect of LSTM unit number. According to the applied one-way ANOVA, we can neglect the null hypothesis, that the number of LSTM units has no significant effect on the correct classification rate of the model, with a significance value of ($p = 0.001$). According to Figure 6.7, the model with 17

Table 6.2: Two-way ANOVA analysis shows that the temporal predicates do not significantly influence the accuracy of the classifier.

| Factors | Significance Value (p) |
|---|---|
| (1) Temporal predicates | 0.595 |
| (2) Number of events | 0.053 |
| 1 by 2 | 0.924 |

LSTM units slightly outperforms the others, therefore we applied this structure for further analysis.

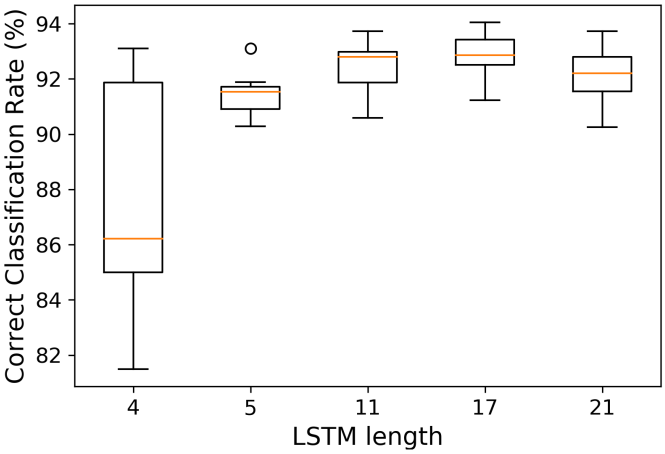


Figure 6.7: The effect of long short-term memory unit number in case of *Dataset B/2* using 7-fold cross-validation. The incorporation of more than 11 units had no significant effect on the correct classification rate.

The size of the embedding layer can also greatly affect the accuracy of the model since this layer maps the one-hot binary vector represented symbols of the states into a continuous vector space. According to Figure 6.8, the highest accuracy is reached by mapping into a four-dimensional embedding space. However, the one-way ANOVA did not verify this increased correct classification rate as the significance value was above 5% ($p = 0.21$), we applied this structure for the testing of the model.

The performance of the described neural classifier is demonstrated with a confusion matrix in Figure 6.9. Only two similar faults are difficult to be distinguished (the $8^{th}$ and the $9^{th}$ faults). This result is in good correspondence with the results presented in Figure 6.6, namely the proposed algorithm can accurately predict the root cause of events after only a few sequence elements, which is highly advantageous from the view of the industrial application. This result can also imply that

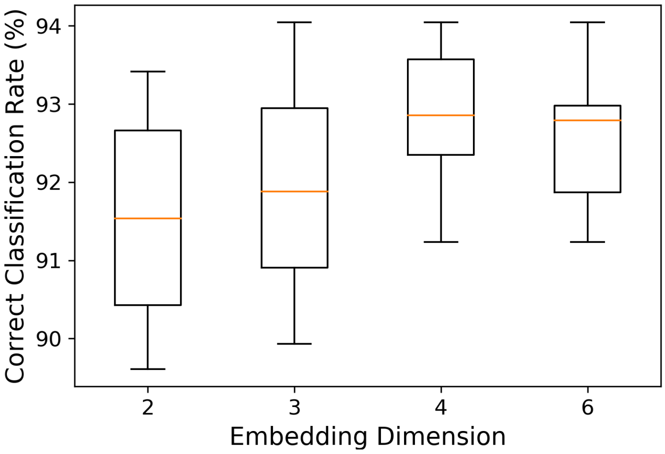


FIGURE 6.8: The effect of the number of units in the embedding dimension in case of *Dataset B/2*

the few characteristic variables, which should be monitored in order to effectively identify the faults can be determined following an analysis of the model.

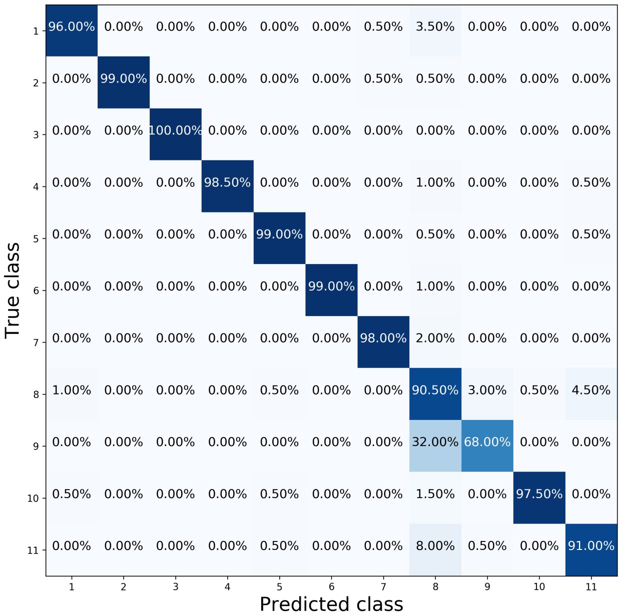


FIGURE 6.9: Confusion matrix showing the accuracy of the classifier. The percentages were calculated after the classification of 200 sequences following each of the faults. The results indicate that faults $8^{th}$ and the $9^{th}$ have similar effects making them difficult to identify.

### 6.3.3 Embedding layer based analysis of process alarms

We visualize the contextual information of sequence elements by the principal component analysis of the **W** weight matrix.

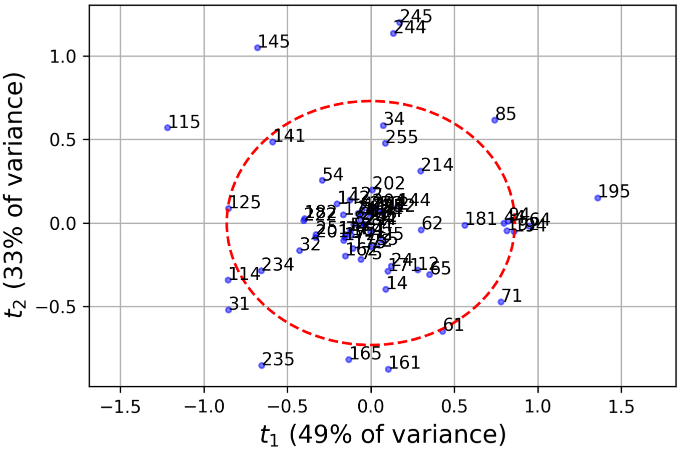


FIGURE 6.10: The result of principal component analysis on the weight matrix of the embedding layer (the last digit of event tags according to Table : 1 - Low Alarm, 2 - Low Warning, 4 - High Warning, 5 - High Alarm).

The results of PCA not only shows the neighbouring relations of the alarms and warnings in this space with reduced dimension, but it also represents its significance in the classification problem. Figure 6.10 illustrates that the most significant events, having large Hotelling's t-squared values are mainly the lower and upper alarms. However, the figure can seem to be overcrowded, the aim of the figure is the visual illustration of the results.

This result offers an outstanding opportunity for the prioritization of incoming warnings and events to facilitate the work of the operators. Table 6.3 shows the fault that these alarms can follow. According to the table, most of these events appear in the case of only one or only a few malfunctions. Therefore the visualization highlights that the principal component analysis of the given alarms can explore the characteristic events of these faults. From an operational point of view, we can see that the high compressor exit temperature (state 115) and the high circulation stream temperature (state 145) event tags are in the same direction at the upper left corner of the diagram, therefore they should indicate similar effects. According to Table 6.3, the high compressor exit temperature (state 115) indicate the presence of the $4^{th}$ fault, which means bad compressor heat flow. The high circulation stream temperature (state 145) also indicates the presence of the $4^{th}$ fault, together with the presence of the $6^{th}$ fault, the bad circulation stream temperature. Therefore as bad compressor exit temperature is experienced the controller of the scrub stream tries to stabilize the temperature of the absorber (the stream coming from the compressor enters the absorber column). However, it is not an outlier of the results of PCA, it is interesting to see that the low circulation stream temperature (state 141) is in the same direction as the high compressor exit temperature (state 115) and the high circulation stream temperature (state 145), and indicates similarly the presence of the $4^{th}$ and $6^{th}$ faults. Similarly, the separator level high alarm (state 85), heater exit temperature high warning (state 64) and the FEHE hot stream exit temperature high warning and high alarm (state 194 and 195) show similarity. This similarity can also be explained by an engineering point of view; these temperatures highly influence the level of the separator. If we take into consideration the results of Table 6.3, it appears all these events are in connection with the $9^{th}$ fault.

A similar analysis can be carried out on the dendrogram in Figure 6.11, that represents the Euclidean distance between the row vectors of the embedding layer. The colours in Figure 6.11 indicate the clusters of process signals, but no former process knowledge-based information was found to be relevant to explain the formed clusters.

For example the pair of the high alarm of gas recycle stream pressure (state 25) and the high alarm of vaporizer pressure (state 55) highlights the problems of the gas cycle, a bad pressure can effect the whole process.The low alarm level for the $H_2O$ content of the column bottom (state 201) and the column bottom level (state 251) are also close to each other since these states are both connected to the bottom of the separation column. A similar connection can be seen between the high warning of reactor exit temperature (state 74) and the scrub stream temperature (state 164), they are connected to each other according to the dendrogram. Similarly, the high warning of heater exit temperature (state 64) and FEHE hot exit temperature (state 194) draw attention on the control problems of the reactor inlet, since a high alarm before the reactor still appears in the outlet temperature of the FEHE unit.

TABLE 6.3: Outlier events of PCA and the faults causing these events. Most of the events occur after only one or very less number of faults, therefore the proposed neural classifier could highlight the characteristic alarms of the process giving a good opportunity for the prioritization of these signals. The tag of the faults can be seen in Table 9.1.

| Event | Name of event | Type of event | Faults |
|---|---|---|---|
| 145 | Circulation Stream Temperature | High alarm | 4, 6 |
| 245 | Decanter Aqueous Level | High alarm | 11 |
| 244 | Decanter Aqueous Level | High warning | 1, 2, 3, 4, 5, 6, 7, 8,9, 10, 11 |
| 85 | Separator Level | High alarm | 3, 9 |
| 195 | FEHE Hot Exit Temperature | High alarm | 9 |
| 64 | Heater Exit Temperature | High warning | 2, 9 |
| 194 | FEHE Hot Exit Temperature | High warning | 4, 6, 7, 9 |
| 71 | Reactor Exit Temperature | Low alarm | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 |
| 61 | Heater Exit Temperature | Low alarm | 2 |
| 161 | Scrub Stream Temperature | Low alarm | 1, 7 |
| 165 | Scrub Stream Temperature | High alarm | 1, 3, 7 |
| 235 | Decanter Organic Level | High alarm | 10 |
| 31 | HAc Tank Level | Low alarm | 1 |
| 114 | Compressor Exit Temperature | High warning | 2, 4, 9 |
| 115 | Compressor Exit Temperature | High alarm | 4 |
| 125 | Absorber Level | High alarm | 1, 2, 3, 4, 5, 6, 7, 9, 11 |

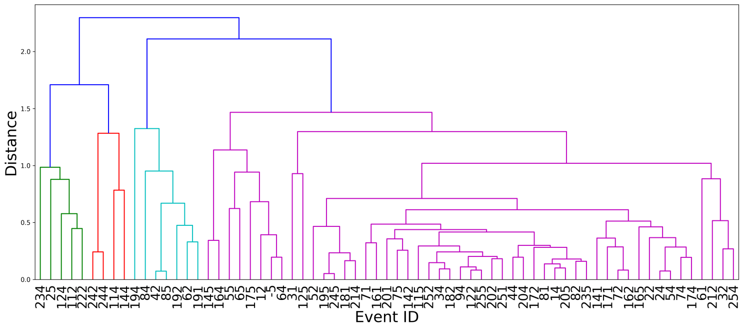


Figure 6.11: The dendrogram of alarms and warnings calculated applying average linkage of Euclidean distances of the row vectors of the weight matrix of the embedding layer (the last digit of event tags according to Table : 1 - Low Alarm, 2 - Low Warning, 4 - High Warning, 5 - High Alarm). The colours of the different states indicate the found clusters, but no process-relevant knowledge has been paired to the clusters.

### 6.3.4 Analysis of the similarity of process faults based on the activities of the long short-term memory units

To better understand the results of the analysis of the embedding layer and to study how the activation values of the hidden layer of LSTM units represent the process faults, we applied principal component analysis to these activation values as well.

The analysis of the eigenvalues of the **H** matrix of the node activities can give information about the proper number of LSTM units (Section 6.2.3). According to Figure 6.12, the eigenvalues confirm the result of the examination of the effect of LSTM unit number (see Figure 6.7): five units are sufficient to represent the classification problem, while eleven units describe almost entirely the cumulative variance of the variables.

The results of the principal component analysis of the activity values of LSTM units (Figure 6.13) are in good agreement with the classification performance illustrated by confusion matrix shown in Figure 6.9. It is interesting to see that the $11^{th}$ fault has two distinct clusters according to the results of PCA. The figure confirms the difficult isolation of the $9^{th}$ fault. The fact that this fault is mostly situated at the centre of PCA projection suggests the lack of an informative state for this fault. The PCA analysis of the alarm and warning signals (Figure 6.10) confirms this result (also see Table 6.3). Such analysis may imply the lack of informative alarms and warnings for the detection of a given fault. Therefore

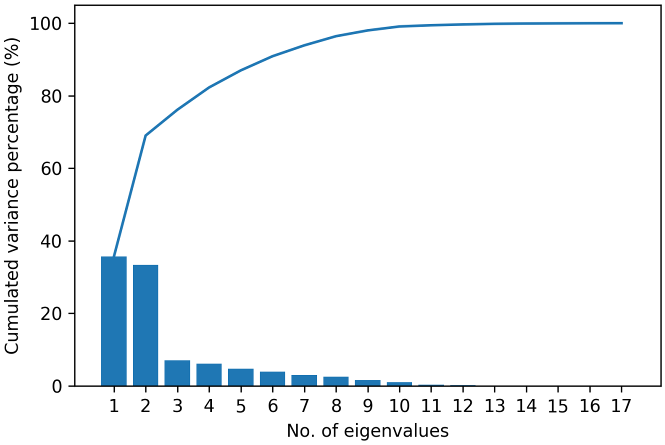


FIGURE 6.12: Eigenvalues of the **H** activity matrix containing the activity values of the layer of the long short-term memory units. According to the figure, the application of four components can sufficiently describe the variance of the activity values.

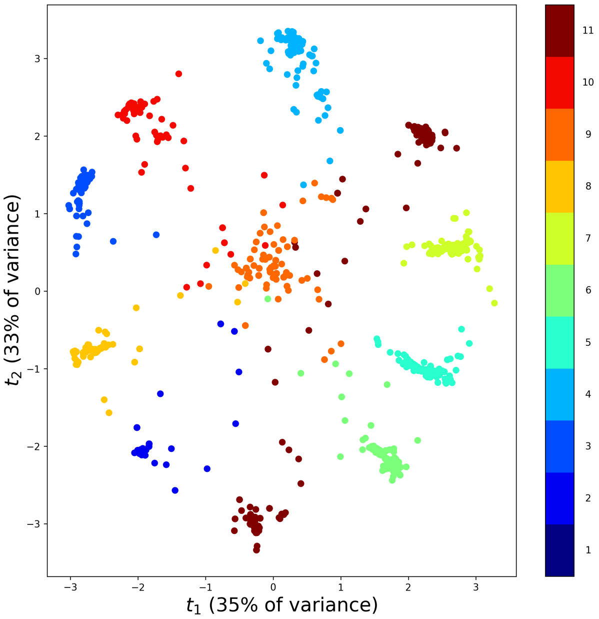


FIGURE 6.13: The principal component analysis of the activity values of the long short-term memory unit layer. The visualization of the activity values highlights the well-separable faults and can give suggestions on poorly monitored malfunctions. In this case, the $9^{th}$ fault reasonably needs more informative alarms and warnings as this fault is in the centre of the graph and its elements show a high distribution in this reduced-dimension space.

the proposed approach can give hints for the development of the control system. Further validation can be provided by improving the definition of alarm messages implemented for the detection of fault 9, as in this case, it should show a better separation from the rest of the faults and better-defined orientation from the center of the figure (however, the recursive definition of new alarm messages based on the LSTM network can be considered a promising future research direction).

## 6.4 Chapter summary

The high amount of alarms and warnings generated by the control systems overload the operators, so the identification and proper handling of malfunctions is a complex and challenging task.

To extract the sometimes hidden information of discrete events in modern chemical plants and the temporal relationships connecting them, we utilized multi-temporal sequences of alarm and warning signals as inputs of long short-term memory (LSTM) unit-based recurrent neural networks trained for classification of process faults and visualized the input and output layers of the model.

The embedding layer is trained simultaneously with the layer of the long short-term memory units and the output layer, resulting in a supervised mapping of the alarm and warning signals into a continuous vector space. This supervised mapping ensures that the principal component analysis of the weight matrix of the embedding layer can highlight the significance of the events from the viewpoint of the process faults.

With the principal component analysis of the activity values, not just the faults become comparable, but it also gives useful information about the proper number of long short-term memory units. The resulted model complexity is in good consistency with the analysis of the effect of the number of units in the embedding layer.

According to our knowledge, the proposed case study is the first that demonstrates that the analysis of the embedding layer and the activity values of LSTM units can give recommendations for the monitoring of malfunctions and offers an outstanding opportunity for the prioritization of incoming warnings and events to facilitate the work of the operators.

The application of the toolbox of deep learning in alarm management and root cause analysis, as well as the extraction of operator information from the trained network form my second thesis finding. The thesis findings are summerized in Section 8.

# Chapter 7

# Conclusions

The presented research explored the application possibilities of advanced machine learning and data analysis within the field of alarm management. The motivation behind the work was to point out the characteristics of these systems in order to improve the safety of processes and facilitate the work of the operators operating them. Assessment of alarm system performance, definition and grouping of alarm messages, root cause analysis and the prediction of future events, as well as the analysis of processes, are just some of the topics incorporated in the thesis, all focusing on the application of data-driven techniques to reduce the human workload of alarm management tasks.

In Chapter 2, I investigated the opportunities for the measurement of the performance of an alarm system and defined seven categories of performance metrics based on the aim of the performance measurement. The chapter provided an overview of the evolution of alarm system performance metrics and highlighted how unbalanced the picture of the applied metrics towards the analysis of the number of different alarm messages. Moreover, I have highlighted how hard it is to measure the quality of a system and defined two possible metrics to incorporate the operator work into this measurement process. It was found that the monitoring of the alarm parameter changes performed by the operators provides a good basis for the recursive improvement of alarm system parameters. However, it is important to consider Goodhart's law during the application of similar metrics: When a measure becomes a target, it ceases to be a good measure. Therefore, however, the work of the operators should be considered in the measurement of the performance of the alarm system, their contribution to the alarm system improvement may not be incorporated in their own performance analysis.

Once we knew what characterizes a good alarm system, Chapter 3 presented how a traditional machine learning technique, the decision trees, can be applied in alarm management for the definition of alarm messages being more informative for fault detection. The findings re-ensured that the proper definition of alarm messages can significantly facilitate the detection of faults in the system by indicating its presence by a relatively small number of alarmed variables. Based on the results obtained by data generated from the VAC simulator, in excess of 90% of the cases the process could be categorized to the proper operating state based on the designed alarm messages.

Moving forward from the definition to the analysis of alarm messages, Chapter 4 presented a novel multi-temporal frequent sequence mining algorithm for the analysis of the alarm & event-log files of industrial plants. The method is capable of the detection of connected alarm sequences that frequently co-occur in the process, moreover, pairing the sequences with probability metrics helps characterization of the alarm events. Finally, by transforming the obtained multi-temporal sequences into Bayes classifiers, the future alarms can be predicted and the uninformative alarms can be suppressed.

Chapter 5 further extends the application of frequent sequence mining in the analysis of alarm & event-log datasets by pairing the analysed events with hierarchical information of the technology. Such hierarchical information puts the information held by the sequences into context and prevents the generation of uninformative sequences. The hierarchical structure is defined in a top-down manner, starting from bigger technological parts towards the individual sensors and actuators monitoring the processes. The hierarchical alarm sequences provide the opportunity to identify the alarm sequences affecting more critical units and the analysis of the spread over effect of malfunctions. Moreover, the network-based representation of hierarchical sequences supports the application of the toolbox of network theory, which is a very promising field of research for process analysis.

Finally, Chapter 6 presents that besides frequent sequence analysis, deep learning is also applicable for the analysis of alarm messages. Assuming a certain number of faults, an LSTM unit-based recurrent neural network model is trained for fault classification utilizing the multi-temporal sequences of alarm and warning signals as inputs. The layers of the network provide supervized mapping of the alarm and warning signals into a continuous vector space, which are analyzed and visualized by traditional data analysis techniques, e.g., principal component analysis and the analysis of the similarity of alarm messages. The technique illustrated how the critical alarm messages can be detected and assigned to malfunctions and how the malfunctions can be analyzed based on the alarm messages following them.

Regarding the future of data-driven alarm management research, three ideas are presented:

- As the alarm system performance metrics have been collected and categorized, an optimal mixture can be defined for the operation for the general monitoring of alarm load and the diagnosis of different problems. Moreover, it would be interesting to make a survey of the applied performance metrics and report their usage in different industrial sectors.
- The sequence-based prediction of future alarms can be further extended by the utilization of sequence similarity metrics: in practice, similar sequences can be related to the same underlying processes and can strengthen (or weaken) each other's prediction.
- The proposed prediction methodologies are applicable for the prediction of multiple future scenarios as well. These scenarios can be represented in an event tree-based structure and the probability of occurrence can be calculated for each scenario.
- The analysis of process related events is the focus of process mining. Although some basic ideas of process mining have been incorporated in the present research as well, the in-depth application of the toolbox of process mining for the analysis of the events of alarm & event-log database to reveal deeper insights into the processes is a promising future research direction.

# Chapter 8

# Thesis findings

Throughout the previous chapters, the theoretic and practical results of my doctoral training were discussed. In the present chapter, I aim to summarize the contributions that I made to the research of data-driven alarm management techniques.

**New scientific results**

1. **I introduced a novel methodology for the prediction of alarm messages in which the state of the technology is represented by the events recorded in the alarm & event-log database of alarm management systems.**

   1.1. I utilized frequent sequence mining algorithms and probability measures to capture the applied operational practices, predict alarm messages, and suppress alarm messages of low-information content. [64, 28, 58, 56, 57]

   1.2. I expanded the frequent sequence-based mining approach with the temporal relationship of events and proposed a multi-temporal sequence mining-based algorithm. [64, 28, 66]

   1.3. I categorized the widely-used alarm system performance metrics into seven application-specific groups, providing the basis for the maintenance of a well-functioning alarm system, which is a prerequisite for the application of advanced data-driven alarm management techniques. [161]

2. **I have demonstrated the capability of deep recurrent neural networks for the extraction of hidden relationships between the variables of large process datasets.**

    2.1. I applied the more and more widespread and trending field of deep learning for the solution of alarm management tasks. [66, 58, 162]

    2.2. I demonstrated how deep recurrent neural network structures facilitate the root cause analysis of alarm sequences. For the extraction of information on the relationships between the discrete events stored in the alarm & event-log databases, multi-temporal sequences of alarm and warning signals are utilized as inputs of a recurrent neural network–based classifier. [66, 58, 162]

    2.3. I presented a novel methodology for the extraction of operational knowledge by the visualization of the weights of deep recurrent neural networks. By visualizing the weights of the linear embedding layer of the proposed deep recurrent neural network, the importance of alarm messages are interpreted, while the weights of the LSTM units provide information on alarm systems capability for the monitor of the effect of faults. [66]

3. **I extended machine learning techniques with process relevant information to provide deeper support for alarm management tasks by facilitating the extraction of relevant engineering knowledge from these models.**

    3.1. I extended the frequent sequence mining-based analysis of the events recorded in the alarm & event-log databases with information on the hierarchical structure of the technology. Based on the approach, the spreading of the effect of malfunctions over the plant is thoroughly traceable on the higher levels of the hierarchy, while the critical elements of the spillover effect are detected on the lower levels. [28]

    3.2. I proposed a novel methodology for the design of alarm messages being informative for fault detection utilizing the distribution of the alarming process variables in the well-known machine learning technique of decision tree classifiers. The chosen decision tree-based technique provides linguistically well-interpretable models with deeper technological insights for the improvement of the operational practices without the modification of the measured process variables. [163]

# Chapter 9

# Appendix

## 9.1 Description of the vinyl acetate process simulator

One of the key drawbacks of alarm management studies is the lack of reproducible results. To provide a benchmark for further research, a well-documented simulation study is developed based on the widely-applied benchmark simulator of a vinyl acetate (VAc) process [85]. In the following, the description of the simulator and the generation of log files are described in detail.

The MATLAB simulator of the benchmark vinyl acetate (VAc) process contains 27 controlled and 26 manipulated variables, therefore it is considered complex enough to test alarm management problems and fault diagnosis algorithms [85]. The process contains 10 basic unit operations with seven chemical components (ethylene ($C_2H_4$), oxygen ($O_2$) and acetic acid (HAc, $CH_3COOH$) are converted into vinyl acetate ($CH_2{=}CHOCOCH_3$) producing the byproducts water ($H_2O$) and carbon dioxide ($CO_2$), moreover, inert ethane ($C_2H_6$) enters with the ethylene feed). A schematic of the production technology can be seen in Figure 9.1.

Regarding the assumptions of the model, please refer to [85], where the simplifications are listed in seven items: (1) there are no light components present in the distillation column, (2) it is assumed that the entire gas loop pressure drop occurs in the reactor, (3) the entire gas loop hold up is assigned to the separator, (4) the pump and compressor dynamics are neglected as they are fast, (5) a constant shell temperature is maintained for the reactor as a manipulated variable,

(6) the cooling jacket temperature of the separator is assumed to be a manipulated variable, (7) a steady-state energy balance is assumed for the heaters and coolers with a two-min time lag to introduce some dynamics. Regarding the thermodynamics and physical property data, the vapor-liquid equilibrium calculations are preformed assuming an ideal vapor phase and standard Wilson liquid activity model. The component vapor pressures are obtained by the Antoine equation.

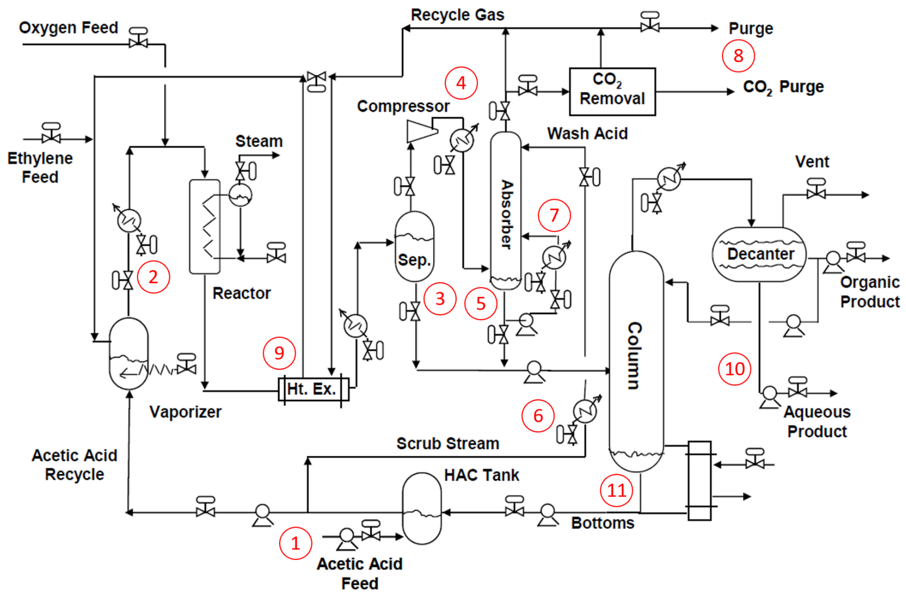


FIGURE 9.1: The flowchart of the VAc production technology (modified version of the figure presented by Chen *et al.*) [85]. The numbers in circle (red) show the place of the implemented fault.

The vaporizer is implemented as a well-mixed unit consisting of seven components, with a gaseous input containing a mixture of the fresh $C_2H_4$ stream and the absorber vapor effluent, as well as a liquid input from the HAc tank. The catalytic plug flow reactor is implemented as a distributed system consisting of ten elements in the axial direction. Inside the reactor, the following exothermic reactions take place:

$$C_2H_4 + CH_3COOH + 1/2\ O_2 \rightarrow CH_2 = CHOCOCH_3 + H_2O \qquad (9.1)$$

$$C_2H_4 + 3\ O_2 \rightarrow 2\ CO_2 + 2\ H_2O \qquad (9.2)$$

The process contains a feed-effluent heat exchanger (FEHE), where a small time constant is added to the exit-temperature sensors to simulate the dynamics of the process. After a pressure letdown valve (which is not shown in Figure 9.1), the effluent is led to a separator. The separator is modeled as a partial condenser, moreover, the flowrates of the leaving liquid and gas stream are calculated with a steady state equilibrium-flash equation. The gas stream enters the bottom part of the absorber unit after having been compressed. The absorber is divided into two parts. The liquid stream of the bottom part, which is the liquid stream leaving the top part, and a circulation stream. The inlet gas of the top part leaves from the bottom part of the absorber, while liquid inlet liquid originates from the HAc tank. A $CO_2$ removal system is implemented after the absorber. A gas removal system is implemented before the azeotropic distillation tower to remove all the light components from the inlet of the tower which originate from the bottom of the separator and absorber units. It is modeled as an ideal separator of components, which completely separates the gaseous components ($O_2$, $CO_2$, $C_2H_4$, $C_2H_6$) and returns them to the inlet of the compressor, while the liquid stream (VAc, $H_2O$, HAc) enters the distillation column. The column is highly nonlinear, with 20 theoretical stages, whose liquid holdup can vary. Downstream the condenser, a decanter is implemented for the separation of the liquid phases. The recirculated liquid stream and the HAc inlet stream are mixed in the HAc tank.

## 9.2 Defining and implementing faults into the vinyl acetate process simulator

The original MATLAB model of the simulator contained five disturbances ( 1.) step change in the composition of $C_2H_6$ in the fresh $C_2H_4$ feed stream from 0.001 to 0.003 mole fraction, 2.) loss of column feed for 5 minutes, 3.) loss of fresh HAc feed stream for 5 minutes, 4.) Loss of fresh $O_2$ feed stream, 5.) an analyzer is off-line (except the $O_2$ analyzer)) [85]. Károly and Abonyi [112] studied the effects of three malfunctions ( 1.) Loss of $HAc$ feed, 2.) Loss of $O_2$ feed and 3.) Loss of column feed), and studied the effect of product changes with the use of the so-called Operating State Matrix (OSM), containing the randomly generated values of the following manipulated variables:

- Operating state start time (min)

- Operating state end time (min)
- Setpoint of the reactor output temperature (150-165 °C)
- $H_2O$ composition in the column's bottom (9 - 18%)
- Vaporizer feed (2.2 - 2.6 $\frac{kmol}{min}$)
- Change of the $C_2H_6$ concentration of the $C_2H_4$ feed from 0.1% to 0.3% (not range based, only two states)

For the efficient and precise work with the VAc simulator, the simulator was revised and additional faults were inserted related to the controllers, as the manipulated value of the process variable remained at a constant value for a specified time. The duration of these faults follows a lognormal distribution. The values of the manipulators in the case of faults can be seen in Table 9.1. The determination of the value of the manipulator variable in case of a fault was determined based on the effect of the chosen manipulator: its effect should be traceable on the simulator, however, it is important to stay in the zone where the simulator produces stable and reproducible results.

## 9.3 Run length distribution based cleaning of alarm data

Good quality alarm data is the prerequisite of advanced alarm management solutions [40]. The aim of our investigations is to explore informative connections between process alarms and plant topology with the determination of frequently occurring alarm patterns. To obtain informative sequences the input database must be cleaned from the less informative alarms, therefore nuisance or constant ones. The most common form of nuisance alarms are the chattering alarms, which do not provide sufficient time for the operators to perform corrective actions and in critical plant conditions can significantly hinder the operators' work. The chattering alarms essentially conflict the philosophy of each alarm should be actionable. Similarly, the constant alarms are also harmful to the quality of the alarm data. In an industrial environment, these long-standing alarms are ignored by the operators either for uninformativeness, or they are hidden from the operators as shelved or

TABLE 9.1: Manipulator values in case of faults (measurement units are neglected as they are not included in the original code and documentation neither)

| Tag of fault | Controlled variable | Manipulated variable | Man. variable value in case of fault |
|---|---|---|---|
| 1 | HAc Tank Level | HAc fresh feed Flow Rate | 0.3 |
| 2 | Heater Exit Temp | Reactor Preheater Heat Flow | 2000 |
| 3 | Separator Level | Separator Liquid Exit Flow Rate | 0 |
| 4 | Compressor Exit Temp. | Compressor Heater Heat Flow | 20000 |
| 5 | Absorber Level | Absorber Liquid Exit Flow Rate | 0 |
| 6 | Circulation Stream Temp. | Absorber Scrub Heat Flow | 5000 |
| 7 | Scrub Stream Temp. | Circulation Cooler Heat Flow | 1000 |
| 8 | $C_2H_6$ in the Gas Recycle | Purge Flow Rate | 0 |
| 9 | FEHE Hot Exit Temp. | Bypass Flow Rate | 0.4 |
| 10 | Decanter Aqueous Level | Aqueous Product Flowrate | 0 |
| 11 | Coloumn Bottom Level | Coloumn Bottom Exit Flowrate | 0 |

forbidden alarms (these alarms are usually still present in the historical datasets). In order to determine chattering alarms, we applied the chattering index introduced by Kondaveeti *et al.* [40] and improved a cleaning methodology based on the counted chattering indexes to avoid uninformatively short or long alarms. The applicability and effectiveness of the proposed methodology are presented through the analysis of the industrial delayed coker plant of the Danube Refinery, MOL Group.

The basis of the calculation of the chatter index is the run-length distribution of the alarms. The *run length* ($r$) of an alarm is the time difference in seconds between two consecutive alarms on the same process variable. The *run-length distribution* (RLD) is the histogram of the different run lengths of the same alarm tag, therefore can be derived by summing up the number of times of different run-lengths appearances. Since process malfunctions prevail for a finite period of time and chattering alarms after a process malfunction do not last more than a specified time period (e.g., $\tau$ seconds), we ignore all the run lengths greater than $\tau$. According to this, the RLD of the alarm tags can be normalized to a Discrete Probability Function (DPF) according to Equation 9.3.

$$P_{r,\tau}(s_i) = \begin{cases} \frac{n_r}{\sum_{r\in\mathbb{N}}^{r=\tau} n_r} & \forall\ r \in \{1,2,3,...,\tau\} \\ 0 & \forall\ r \in \{\tau+1, \tau+2, \tau+3, ..., \infty\} \end{cases} \tag{9.3}$$

where $P_{r,\tau}$ marks the discrete probability of a run length $r$ neglecting run lengths longer than $\tau$ and $n_r$ represents the number of alarms with the defined $r$ run length in the examined time period. $\sum_{r\in\mathbb{N}}^{r=\tau} n_r$ is the normalization factor, which is one less than the total number of alarms on the examined process variable no longer than $\tau$ in the examined time period, since the last alarm does not have a defined run length.

The chatter index is defined by weighting the DPF by the inverse of the run length of the given alarm. The inverse of the run length is the frequency of its occurrence and is used for the emphasis of the alarm counts with short run lengths. According to this, the chatter index ($\psi_\tau$) can be calculated as presented in Equation 9.4.

$$\psi_\tau = \sum_{r\in\mathbb{N}} P_{r,\tau}(s_i)\frac{1}{r} \tag{9.4}$$

According to [13], an alarm tag with a frequency of 3 or more alarms on the same tag in a minute can be considered a chattering alarm tag. Thus we defined a cutoff limit of $\psi$ to identify the problematic tags. This cutoff value is $\psi_{cutoff} = \frac{3}{60} = 0.05$ alarms/s.

We applied this cutoff value as a rule of thumb for the cleaning of the investigated industrial database from uninformatively short alarm run lengths. After the calculation of the DPF, we summed up the probability values in a reverse order starting from the value related to the longest analysed run length, until the sum reached the cutoff value. The first element of these pairs of temporal instances related to the run length values that would increase the summed DPF above this limit was neglected. This way we defined a varying threshold for each alarm tag with an intention to keep only the informative instances of each tag and reduce chattering.

# Chapter 10

# Farewell

Beyond the formerly discussed techniques that I learnt and contributions that I made during the years of my doctoral training, many lifelong lessons shaped my mind and character. With all the love and respect of the research profession, let me present a non-extensive list of these lessons:

- I learnt that "*the spaces of our opportunities and abilities do not always meet*". I learnt that in the hard way. (*A lehetőségeink és képességeink tere nem mindig találkozik.*)
- *On the last 10% of a manuscript, we need to work as much as on the first 90%.* (And as I see, this Pareto-like principle is true to almost everything in life.)
- A crucial message of my PhD is that it is not enough to know that we are capable of doing something. We need to do it, and most importantly, we need to show that we did it.
- I learnt that most things in the world are just "*three rows in Matlab*".
- I did not learn to be patient. However, I started to respect it. Furthermore, covet for it.

As the vast majority of the thesis works, this one is getting longer and more complicated as I am getting more profound in the details. So, Dear Reader, in order to save you from these sufferings, I wrap things up with a frequent sentence (*sequence*) of my time during the doctoral training: "*Let's grab a coffee!*"

# Acronyms

**Chapter 4.** ***Frequent sequence mining for the analysis of industrial processes***

HAZOP - hazard and operability study

EEMUA - Engineering Equipment and Materials Users' Association

$m$ - mean of the lognormal distribution

$v$ - variance of the lognormal distribution

$\mu$ - the mean of the associated normal distribution

$\sigma$ - the standard deviation of the associated normal distribution

$T_{simend}$ - the end time of the simulation

OSM - Operating State Matrix

EEM - Error Event Matrix

$s$ - *state* of the technology

$pv$ - index of the process variable

$a$ - the attribute showing the process variable's value related to the alarm and warning limits, such as $a \epsilon \{Low\ A, Low\ W, High\ W, High\ A\}$

$A$ - alarm

$W$ - warning

$e$ - event

$st$ - starting time of an event

$et$ - ending time of an event

$D_T$ - an example for an event log database

$window$ - *time window* for the determination of event correlation

$E$ - equal temporal predicates between events

$B$ - before temporal predicates between events

$D$ - during temporal predicates between events

$O$ - overlap temporal predicates between events

$R$ - arbitrary temporal predicate between events

$\Phi$ - a temporal pattern of events

$\phi$ - a temporal instance of events

$\Phi_k$ - notation of a $k$-length temporal pattern

$support(\Phi)$ - support value of a sequence

$supp(\Phi)$ - supporting events of a sequence

$|E|$ - the maximal number of events supporting each states in the $D_T$ temporal database

$N$ - the number of states in the $D_T$ temporal database

$minSupp$ - the defined support threshold of the mining algorithm

$minConf$ - the defined confidence threshold of the mining algorithm

$conf(\Phi)$ - the degree of confidence for the given sequence

$P_{F,1,2,3}$ - the probability of the occurrence of the given fault

$PV$ - process value

$Sim_{i,j}$ - Similarity measure of fault type $i$ and $j$

**Chapter 5. *Hierarchical Frequent Sequence Mining for the Analysis of Production Processes***

DCS - distributed control system

PCA - principal component analysis

$s$ - *state* of the technology

$pv$ - index of the process variable

$a$ - the related state signals

$n_h$ - number of levels of hierarchy in a production system

$h$ - level of hierarchy

$\mathcal{S}^i$ - set of states on the hierarchical level indicated by its subscript

$e$ - event

$st$ - starting time of an event

$et$ - ending time of an event

$DB$ - alarm management database

$\mathbf{\Phi}_k$ - $k^{th}$ sequence of states

$S$ - the set of states

$R$ - arbitrary temporal predicate between events

$E$ - equal temporal predicates between events

$B$ - before temporal predicates between events

$D$ - during temporal predicates between events

$O$ - overlap temporal predicates between events

$\phi$ - temporal instance of an event sequence

$\Phi_k$ - a frequent pattern of $k+1$ states, connected with $k$ temporal predicates

$P(s_i)$ - the probability of occurrence of state $s_i$

$supp(\Phi)$ - number of temporal instances of $\Phi$ sequence of states in the examined time temporal database

$|E|$ - the maximal number of events supporting the most frequent state in the examined temporal database

$support(\Phi)$ - support value of a sequence, a measure of relevance

$minSupp$ - the defined support threshold of the Hierarchical Multi-Temporal Mining algorithm

$minConf$ - the defined confidence threshold of the Hierarchical Multi-Temporal Mining algorithm

$P(s_k|\Phi_{k-1})$ - conditional probability of the occurrence of the $s_k$ state after the $\Phi_{k-1}$ sequence

$conf(\Phi)$ - confidence value of a sequence, a measure of reliability

$c_j$ - the $j^{th}$-type root cause of the following events

$dependency(s_0 \overset{R_1}{\Rightarrow} s_1)$ - dependency value of a sequence, describing the causal relationship of sequence elements

$directionality(s_0 \overset{R_1}{\Rightarrow} s_1)$ - directionality value of a sequence, describing the direction of link between sequence elements

$F_{h,k}$ - the stored $k$-length frequent temporal patterns on the $h^{th}$ level of hierarchy

$RLD$ - run-length distribution of the temporal instances of the same type of alarm

$DPF$ - discrete probability function of the run length of temporal instances of the same type of alarm

$r$ - run length of a temporal instance of alarm

$n_r$ - the number of temporal occurrences with the defined run length $r$ from the examined a type of alarm

$\sum_{r\in\mathbb{N}}^{r=\tau} n_r$ - one less than the total number of the examined type of alarms no longer than $\tau$ during the examined time period

$P_{r,\tau}$ - modified discrete probability of the run length $r$ ignoring run lengths greater than $\tau$

$\psi_\tau$ - modified chatter index ignoring run lengths greater than $\tau$

$\psi_{cutoff}$ - the defined chatter index limit above which we consider an alarm problematic from the view of chattering

MDEA - methyl diethanolamine

MEROX - mercaptan oxidation

**Chapter 6. *Sequence-based fault detection and isolation***

RNN - recurrent neural network

LSTM - long short-term memory

PCA - principal component analysis

VAc - vinyl acetate

$y$ - type of fault, $y = \{c_1, \ldots, c_{n_c}\}$, related to the $k$-th sequence of events.

$\hat{y}_k$ - predicted class label of the faults

$c_j$ - $j^{th}$ type fault

$n_c$ - number of fault types

$s$ - *state* of the technology

$pv$ - index of the process variable

$a$ - the related state signals

$e$ - event

$st$ - starting time of an event

$et$ - ending time of an event

$\mathbf{\Phi}_k$ - $k^{th}$ sequence of states

$S$ - the set of states

$R$ - arbitrary temporal predicate between events

$E$ - equal temporal predicates between events

$B$ - before temporal predicates between events

$D$ - during temporal predicates between events

$O$ - overlap temporal predicates between events

$\mathbf{x}_k$ - the numerical representation of $\mathbf{\Phi}_k$ sequence

$P(y_k = c_j|\mathbf{x}_k)$ - the conditional probability of the given fault

$\mathbf{h}_k$ - vector of the activities of the LSTM units

$n_U$ - the number of LSTM units in the RNN

$T$ - length of the input sequence

$\mathbf{W}_s$ - weight matrix of the output layer of the network (referring to the applied softmax function)

$C_k^t$ - cell-state of the $t^{th}$ LSTM unit

$f_k^t$ - forget gate of the $t^{th}$ LSTM unit

$\mathbf{W}_f$ - weight matrix of the forget gate

$b$ - the bias vector of the corresponding neurons

$i_k^t$ - input gate of the $t^{th}$ LSTM unit

$o_k^t$ - output gate of the $t^{th}$ LSTM unit

$\mathbf{W}_o$ - weight matrix of the output gate

$\mathbf{oh}_k^t$ - one-hot binary vector representation of the $t^{th}$ symbol of the $k^{th}$ sequence

$n_s$ - the number of states in the $\mathbb{D}$ temporal database

$n_R$ - number of types of temporal predicates between events

$n_o$ - number of bits in the one-hot binary vector

$n_e$ - dimension of the embedding layer

$\mathbf{W}$ - weight matrix of embedding layer

$sim(s_i, s_j)$ - similarity of the alarms

$d_{max}$ - the maximum Euclidean distance among the rows of the $\mathbf{W}$ matrix

| Name of Variable | Abbreviations | Unit | Tag of Variable | Tag of Low Alarm | Tag of High Alarm |
|---|---|---|---|---|---|
| $\%O_2$ in the Reactor Inlet | $\%O2$ | $\%mol$ | 1 | 11 | 12 |
| Gas Recycle Stream Pressure | Pres | Psia | 2 | 21 | 22 |
| HAc Tank Level | HAc-L | - | 3 | 31 | 32 |
| Vaporizer Level | Vap-L | - | 4 | 41 | 42 |
| Vaporizer Pressure | Vap-P | Psia | 5 | 51 | 52 |
| Heater Exit Temperature | Pre-T | $^\circ C$ | 6 | 61 | 62 |
| Reactor Exit Temperature | RCT-T | $^\circ C$ | 7 | 71 | 72 |
| Separator Level | Sep-L | - | 8 | 81 | 82 |
| Separator Temperature | Sep-T | $^\circ C$ | 9 | 91 | 92 |
| Separator Vapor Flowrate | Sep-V | Kmol/min | 10 | 101 | 102 |
| Compressor Exit Temperature | Com-T | $^\circ C$ | 11 | 111 | 112 |
| Absorber Level | Abs-L | - | 12 | 121 | 122 |
| Absorber Scrub Flowrate | Cir-F | Kmol/min | 13 | 131 | 132 |
| Circulation Stream Temperature | Cir-T | $^\circ C$ | 14 | 141 | 142 |
| Absorber Circulation Flowrate | Scr-F | Kmol/min | 15 | 151 | 152 |
| Scrub Stream Temperature | Scr-T | $^\circ C$ | 16 | 161 | 162 |
| $\%CO_2$ in the Gas Recycle | $\%CO_2$ | $\%mol$ | 17 | 171 | 172 |
| $\%C_2H_6$ in the Gas Recycle | $\%C_2H_6$ | $\%mol$ | 18 | 181 | 182 |
| FEHE Hot Exit Temperature | FEHE-T | $^\circ C$ | 19 | 191 | 192 |
| $\%H_2O$ in the Column Bottom | $\%H_2O$ | $\%mol$ | 20 | 201 | 202 |
| $5^{th}$ tray Temperature | Col-T | $^\circ C$ | 21 | 211 | 212 |
| Decanter Temperature | Dect-T | $^\circ C$ | 22 | 221 | 222 |
| Decanter Organic Level | Org-L | - | 23 | 231 | 232 |
| Decanter Aqueous Level | Aqu-L | - | 24 | 241 | 242 |
| Column Bottom Level | Col-L | - | 25 | 251 | 252 |
| Liquid Recycle Flowrate | Vap-In | Kmol/min | 26 | 261 | 262 |
| %VAc E-3 | $\%VAcE-3$ | $\%mol$ | 27 | 271 | 272 |

TABLE 10.2: The different abbreviations of the variables of the VAC technology. The abbreviations are applied in Chapter 3, the tags of low and high alarms are applied in Chapter 4, while the tags of the variables are applied in Chapter 6.

# Bibliography

[1] Engineering Equipment, Materials Users' Association, Engineering Equipment, and Materials Users Association Staff. *Alarm Systems: A Guide to Design, Management and Procurement.* EEMUA publication. E E M U A (Engineering Equipment & Materials Users Association), 2014.

[2] B.R. Hollifield and E. Habibi. *Alarm Management: A Comprehensive Guide: Practical and Proven Methods to Optimize the Performance of Alarm Management Systems.* International Society of Automation, 2011.

[3] Engineering Equipment, Materials Users' Association, Engineering Equipment, and Materials Users Association Staff. *Alarm Systems: A Guide to Design, Management and Procurement.* EEMUA publication. E E M U A (Engineering Equipment & Materials Users Association), 2015.

[4] Hiroaki Tanaka. Optimize alarm management, 2018.

[5] M.J. Jafari, M. Pouyakian, A. khanteymoori, and S.M. Hanifi. Reliability evaluation of fire alarm systems using dynamic bayesian networks and fuzzy fault tree analysis. 67, 2020.

[6] David A. Lee. Safe operations using advanced operator graphics. *Process Safety Progress*, 39(3):e12119, 2020.

[7] Engineering Equipment, Materials Users' Association, Engineering Equipment, and Materials Users Association Staff. *Alarm Systems: A Guide to Design, Management and Procurement.* EEMUA publication. E E M U A (Engineering Equipment & Materials Users Association), 1999.

[8] TC 65/SC 65A System aspects. *IEC 62682:2014 - Management of alarm systems for the process industries.* International Electrotechnical Commission, 2014.

[9] Kim VanCamp. Alarm management by the numbers. *Chemical Engineering Essentials for the CPI Professional, Automation & Control*, 2016.

[10] P. Goel, A. Datta, and M.S. Mannan. Industrial alarm systems: Challenges and opportunities. 50:23–36, 2017.

[11] Norwegian Petroleum Directorate. *Principles for alarm system design - YA-711.* Norwegian Ministry of Petroleum and Energy, 2001.

[12] NA 102. *Alarm Management.* NAMUR Publication, 2005.

[13] International Society of Automation and American National Standards Institute. *ANSI/ISA-18.2-2009: Management of Alarm Systems for the Process Industries.* AMERICAN NATIONAL STANDARD, 2009.

[14] ANSI/ISA-18.2-2016. *ANSI/ISA-18.2-2016: Management of Alarm Systems for the Process Industries.* AMERICAN NATIONAL STANDARD, 2014.

[15] The International Society of Automation. *ISA TR18.2.1 - Alarm Philosophy.* International Society of Automation (ISA), 2018.

[16] RP 1167. *Pipeline SCADA Alarm Management, 1st Edition.* American Petroleum Institute, 2010.

[17] RP 1167. *Pipeline SCADA Alarm Management, 2nd Edition.* American Petroleum Institute, 2016.

[18] A. Nochur, H. Vedam, and J. Koene. Alarm performance metrics. *IFAC Proceedings Volumes*, 34(27):203 – 208, 2001. 4th IFAC Workshop on On-Line Fault Detection and Supervision in the Chemical Process Industries 2001, Jejudo Island, Korea, 7-8 June 2001.

[19] 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*, 13(2):1045–1061, 2016.

[20] Sandeep R. Kondaveeti, Iman Izadi, Sirish L. Shah, Tim Black, and Tongwen Chen. Graphical tools for routine assessment of industrial alarm systems. *Computers & Chemical Engineering*, 46:39 – 47, 2012.

[21] P. Goel, E.N. Pistikopoulos, M.S. Mannan, and A. Datta. A data-driven alarm and event management framework. 62, 2019.

[22] F. Yang and C. Guo. Survey on advanced alarm strategies based on multivariate analysis. pages 612–617, 2017.

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

[24] Johannes Koene and Hiranmayee Vedam. Alarm management and rationalization. In *Third International conference on loss prevention*, 2000.

[25] Johannes Koene and Hiranmayee Vedam. Alarm management and rationalization. In *Third International Conference on Loss Prevention*, 2000.

[26] Ross Guy. Best practice management of industrial process control alarm floods. 2016.

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

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

[29] B. Scholten. *The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing*. ISA, 2007.

[30] M.A.C. Solutions (UK) Ltd. The sense and nonsense of alarm system performance kpis - what are meaningful values?, 2018.

[31] M. L. Bransby and J. Jenkinson. *The Management of Alarm Systems*. Health and Safety Executive, 2000.

[32] B. A. Walker, K. D. Smith, and M. D. Kekich. Limiting shift-work fatigue in process control. *Chemical engineering progress*, 99:54–57, 2003.

[33] Mauricio Moreno Santos and Bárbara Sá. Alarm management program: Implementation experience in a petrochemical company. In *CCPS Latin American Conference on Process Safety*, 2009.

[34] P. Grosdidier, P. Connor, B. Hollifield, and S. Kulkarni. A path forward for dcs alarm management. *Hydrocarbon Processing*, 82:59–64, 11 2003.

[35] I. Izadi, S. L. Shah, and T. Chen. Effective resource utilization for alarm management. In *49th IEEE Conference on Decision and Control (CDC)*, pages 6803–6808, 2010.

[36] J. Sompura, A. Joshi, B. Srinivasan, and R. Srinivasan. A practical approach to improve alarm system performance: Application to power plant. 27(5):1094–1102, 2019.

[37] Wenkai Hu, Muhammad Shahzad Afzal, Gustavo Brandt, Eric Lau, Tongwen Chen, and Sirish L. Shah. An application of advanced alarm management tools to an oil sand extraction plant, this work was supported by an nserc crd project with suncor energy as an industrial partner. *IFAC-PapersOnLine*, 48(8):641 – 646, 2015. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

[38] P. Goel, A. Datta, and M. Sam Mannan. Data mining and analysis for alarm management. pages 392–401, 2018.

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

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

[41] Ana María Peco Chacón and Fausto Pedro García Márquez. *False Alarms Management by Data Science*, pages 301–316. Springer International Publishing, Cham, 2019.

[42] J. Wang and T. Chen. An online method for detection and reduction of chattering alarms due to oscillation. 54:140–150, 2013.

[43] S. Kondaveeti, P. Grover, M. Khan, and S. Shah. An application for automated reporting of industrial alarm system performance. In *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pages 480–484, 2014.

[44] Yongkui Sun, Wen Tan, and Tongwen Chen. A method to remove chattering alarms using median filters. *ISA Transactions*, 73:201–207, 2018.

[45] J. Sompura, P. Shankar, S. Gamit, B. Srinivasan, and R. Srinivasan. Lessons learnt from alarm management in a combined-cycle gas turbine power plant. 40:2461–2466, 2017.

[46] E. Naghoosi, I. Izadi, and T. Chen. A study on the relation between alarm deadbands and optimal alarm limits. In *Proceedings of the 2011 American Control Conference*, pages 3627–3632, 2011.

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

[48] J. Xu, J. Wang, I. Izadi, and T. Chen. Performance assessment and design for univariate alarm systems based on far, mar, and aad. *IEEE Transactions on Automation Science and Engineering*, 9(2):296–307, 2012.

[49] S. Breznitz. *Cry Wolf: The Psychology of False Alarms*. Lawrence Erlbaum Associates, 1984.

[50] Rachel Adler and Raquel Benbunan-Fich. The effects of task difficulty and multitasking on performance. *Interacting with Computers*, 27, 06 2014.

[51] Stephen Monsell. Task switching. *Trends in cognitive sciences*, 7:134–140, 04 2003.

[52] Rockwell Automation. Performance benchmarking and alarm philosophy development, 2017.

[53] Wenkai Hu, Ahmad W. Al-Dabbagh, Tongwen Chen, and Sirish L. Shah. Process discovery of operator actions in response to univariate alarms**this work was supported by the natural sciences and engineering research council of canada via the crd program. *IFAC-PapersOnLine*, 49(7):1026 – 1031, 2016. 11th IFAC Symposium on Dynamics and Control of Process SystemsIncluding Biosystems DYCOPS-CAB 2016.

[54] W. HU, T. Chen, and G. Meyer. Constructing workflow models of alarm responses via trace labeling and dependency analysis. In *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1615–1620, 2019.

[55] Wenkai Hu, Ahmad W. Al-Dabbagh, David Li, and Tongwen Chen. Extraction and graphical representation of operator responses to multivariate alarms in industrial facilitiesthis work was supported by the natural sciences and engineering research council of canada via the crd program. *IFAC-PapersOnLine*, 51(18):25–30, 2018. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.

[56] Gyula Dorgo, Kristof Varga, Mate Haragovics, Tibor Szabo, and Janos Abonyi. Towards operator 4.0, increasing production efficiency and reducing operator workload by process mining of alarm data. *Chemical Engineering Transactions*, 70:829 – 834, 2018.

[57] J. Abonyi and G. Dorgo. Process mining in production systems. In *2019 IEEE 23rd International Conference on Intelligent Engineering Systems (INES)*, pages 000267–000270, 2019.

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

[59] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Surya N. Kavuri, and Kewen Yin. A review of process fault detection and diagnosis: Part iii: Process history based methods. *Computers & Chemical Engineering*, 27(3):327 – 346, 2003.

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

[61] Hao Zang, Fan Yang, and Dexian Huang. Design and analysis of improved alarm delay-timers. *IFAC-PapersOnLine*, 48(8):669 – 674, 2015.

[62] Fumitaka Higuchi, Ichizo Yamamoto, Tsutomu Takai, Masaru Noda, and Hirokazu Nishitani. Use of event correlation analysis to reduce number of alarms. In Rita Maria de Brito Alves, Caludio Augusto Oller do Nascimento, and Evaristo Chalbaud Biscaia, editors, *10th International Symposium on Process Systems Engineering: Part A*, volume 27 of *Computer Aided Chemical Engineering*, pages 1521 – 1526. Elsevier, 2009.

[63] 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*, 65(9):7290–7300, Sept 2018.

[64] G. Dorgo and J. Abonyi. Sequence mining based alarm suppression. *IEEE Access*, 6:15365–15379, 2018.

[65] J. Shang and T. Chen. Early classification of alarm floods via exponentially attenuated component analysis. *IEEE Transactions on Industrial Electronics*, 67(10):8702–8712, 2020.

[66] Gyula Dorgo, Peter Pigler, and Janos Abonyi. Understanding the importance of process alarms based on the analysis of deep recurrent neural networks trained for fault isolation. *Journal of Chemometrics*, 32(4):e3006, 2018. e3006 cem.3006.

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

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

[69] Matthieu Lucke, Moncef Chioua, Chriss Grimholt, Martin Hollender, and Nina F. Thornhill. Advances in alarm data analysis with a practical application to online alarm flood classification. *Journal of Process Control*, 79:56 – 71, 2019.

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

[71] M. Modarres and T. Cadman. A method of alarm system analysis for process plants. *Computers & Chemical Engineering*, 10(6):557 – 565, 1986.

[72] Luo Yan, Liu Xiwei, Masaru Noda, and Hirokazu Nishitani. Systematic design approach for plant alarm systems. *JOURNAL OF CHEMICAL ENGINEERING OF JAPAN*, 40(9):765–772, 2007.

[73] Kazuhiro Takeda, Takashi Hamaguchi, Naoki Kimura, and Masaru Noda. A design method of a plant alarm system for first alarm alternative signals using a modularized ce model. *Process Safety and Environmental Protection*, 92(5):406 – 411, 2014. Process Systems Engineering.

[74] Naoki Kimura, Takashi Hamaguchi, Kazuhiro Takeda, and Masaru Noda. Determination of alarm setpoint for alarm system rationalization using performance evaluation. In Sakae Yamamoto, editor, *Human Interface and the Management of Information. Information and Interaction for Health, Safety, Mobility and Complex Environments*, pages 507–514, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[75] Yanjun Chang, Faisal Khan, and Salim Ahmed. A risk-based approach to design warning system for processing facilities. *Process Safety and Environmental Protection*, 89(5):310 – 316, 2011.

[76] Chi-Shih Chao and An-Chi Liu. An alarm management framework for automated network fault identification. *Computer Communications*, 27(13):1341 – 1353, 2004.

[77] R. Srinivasan, J. Liu, K. W. Lim, K. C. Tan, and W. K. Ho. Intelligent alarm management in a petroleum refinery. *Hydrocarbon Processing*, 83(11):47–53, 11 2004.

[78] Tao Chen. On reducing false alarms in multivariate statistical process control. *Chemical Engineering Research and Design*, 88(4):430 – 436, 2010.

[79] Iman Izadi, Sirish L. Shah, David S. Shook, and Tongwen Chen. An introduction to alarm analysis and design. *IFAC Proceedings Volumes*, 42(8):645 – 650, 2009. 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.

[80] F. Yang, S. L. Shah, and D. Xiao. Correlation analysis of alarm data and alarm limit design for industrial processes. In *Proceedings of the 2010 American Control Conference*, pages 5850–5855, June 2010.

[81] Wende Tian, Guixin Zhang, and Huiting Liang. Alarm clustering analysis and ACO based multi-variable alarms thresholds optimization in chemical processes. *Process Safety and Environmental Protection*, 113:132 – 140, 2018.

[82] Guixin Zhang, Zhenlei Wang, and Hua Mei. Sensitivity clustering and roc curve based alarm threshold optimization. *Process Safety and Environmental Protection*, 141:83 – 94, 2020.

[83] G. Baumont, F. Ménage, J.R. Schneiter, A. Spurgin, and A. Vogel. Quantifying human and organizational factors in accident management using decision trees: the horaam method. *Reliability Engineering  System Safety*, 70(2):113 – 124, 2000.

[84] Tamas Varga, Ferenc Szeifert, and Janos Abonyi. Decision tree and first-principles model-based approach for reactor runaway analysis and forecasting. *Engineering Applications of Artificial Intelligence*, 22(4):569 – 578, 2009.

[85] Rong Chen, Kedar Dave, Thomas J. McAvoy, and Michael Luyben. A nonlinear dynamic model of a vinyl acetate process. *Industrial & Engineering Chemistry Research*, 42(20):4478–4487, 2003.

[86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[87] Louis Wehenkel. On uncertainty measures used for decision tree induction. In Bernadette Bouchon-Meunier, editor, *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 1996.

[88] Hans Pasman. Chapter 8 - extended process control, operator situation awareness, alarm management. In Hans Pasman, editor, *Risk Analysis and Control for Industrial Processes - Gas, Oil and Chemicals*, pages 355 – 381. Butterworth-Heinemann, Oxford, 2015.

[89] B.R. Mehta and Y.J. Reddy. Chapter 21 - alarm management systems. In B.R. Mehta and Y.J. Reddy, editors, *Industrial Process Automation Systems*, pages 569 – 582. Butterworth-Heinemann, Oxford, 2015.

[90] James Bliss, Richard D. Gilson, and John E. Deaton. Human probability matching behaviour in response to alarms of varying reliability. 38:2300–12, 12 1995.

[91] Robert D. Sorkin and David D. Woods. Systems with human monitors: A signal detection analysis. *Human–Computer Interaction*, 1(1):49–75, 1985.

[92] Joachim Meyer and Yuval Bitan. Why better operators receive worse warnings. *Human Factors*, 44(3):343–353, 2002. PMID: 12502153.

[93] Michael Rayo and Susan Moffatt-Bruce. Alarm system management: Evidence-based guidance encouraging direct measurement of informativeness to improve alarm response. 24, 03 2015.

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

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

[96] Alan J. Hugo. Estimation of alarm deadbands. *IFAC Proceedings Volumes*, 42(8):663 – 667, 2009.

[97] Iman Izadi, Sirish L. Shah, David S. Shook, Sandeep R. Kondaveeti, and Tongwen Chen. A framework for optimal design of alarm systems. *IFAC Proceedings Volumes*, 42(8):651 – 656, 2009.

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

[99] Sandeep R. Kondaveeti, Iman Izadi, Sirish L. Shah, and Tim Black. Graphical representation of industrial alarm data. *IFAC Proceedings Volumes*, 43(13):181 – 186, 2010. 11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems.

[100] Z. Yang, J. Wang, and T. Chen. Detection of correlated alarms based on similarity coefficients of binary data. *IEEE Transactions on Automation Science and Engineering*, 10(4):1014–1025, Oct 2013.

[101] Wenkai Hu, Jiandong Wang, and Tongwen Chen. A new method to detect and quantify correlated alarms with occurrence delays. *Computers & Chemical Engineering*, 80:189 – 198, 2015.

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

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

[104] Vinícius Barroso Soares, José Carlos Pinto, and Maurício Bezerra de Souza. Alarm management practices in natural gas processing plants. *Control Engineering Practice*, 55(Supplement C):185 – 196, 2016.

[105] Savo Kordic, Peng Lam, Jitian Xiao, and Huaizhong Li. Associative data mining for alarm groupings in chemical processes. In Atlantis Press, editor, *Proceedings of the 2007 International Conference on Intelligent Systems and Knowledge Engineering*.

[106] Savo Kordic, Peng Lam, Jitian Xiao, and Huaizhong Li. *Analysis of Alarm Sequences in a Chemical Plant*, pages 135–146. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[107] J. Folmer and B. Vogel-Heuser. Computing dependent industrial alarms for alarm flood reduction. In *International Multi-Conference on Systems, Sygnals Devices*, pages 1–6, March 2012.

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

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

[110] K. Ahmed, I. Izadi, T. Chen, D. Joe, and T. Burton. Similarity analysis of industrial alarm flood data. *IEEE Transactions on Automation Science and Engineering*, 10(2):452–457, April 2013.

[111] Wenkai Hu, Sirish L. Shah, and Tongwen Chen. Framework for a smart data analytics platform towards process monitoring and alarm management. *Computers & Chemical Engineering*, 2017.

[112] R. Karoly and J. Abonyi. Multi-temporal sequential pattern mining based improvement of alarm management systems. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 003870–003875, Oct 2016.

[113] Jonathan Goh, Sridhar Adepu, Khurum Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems, 10 2016.

[114] Sandeep R. Kondaveeti, Sirish L. Shah, and Iman Izadi. Application of multivariate statistics for efficient alarm generation. *IFAC Proceedings Volumes*, 42(8):657 – 662, 2009. 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.

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

[116] Hangzhou Wang, Faisal Khan, Salim Ahmed, and Syed Imtiaz. Risk-based warning system design methodology for multimode processes. *IFAC-PapersOnLine*, 48(8):663 – 668, 2015. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015.

[117] O. Chenaru, D. Popescu, and D. Enache. Practical fault management using real-time decision tree analysis. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 384–389, June 2016.

[118] Leo H. Chiang and Richard D. Braatz. Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics and Intelligent Laboratory Systems*, 65(2):159 – 178, 2003.

[119] S.Y. Yim, H.G. Ananthakumar, L. Benabbas, A. Horch, R. Drath, and N.F. Thornhill. Using the process schematic in plant-wide disturbance analysis. In W. Marquardt and C. Pantelides, editors, *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*, volume 21 of *Computer Aided Chemical Engineering*, pages 1431 – 1436. Elsevier, 2006.

[120] S.Y. Yim, H.G. Ananthakumar, L. Benabbas, A. Horch, R. Drath, and N.F. Thornhill. Using process topology in plant-wide control loop performance assessment. *Computers & Chemical Engineering*, 31(2):86 – 99, 2006.

[121] Jegatheeswaran Thambirajah, Lamia Benabbas, Margret Bauer, and Nina F. Thornhill. Cause-and-effect analysis in chemical processes utilizing xml, plant connectivity and quantitative process history. *Computers & Chemical Engineering*, 33(2):503 – 512, 2009.

[122] G. J. Di Geronimo Gil, D. B. Alabi, O. E. Iyun, and N. F. Thornhill. Merging process models and plant topology. In *2011 International Symposium on Advanced Control of Industrial Processes (ADCONIP)*, pages 15–21, May 2011.

[123] Jiawei Han and Yongjian Fu. Mining multiple-level association rules in large databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(5):798–805, Sep 1999.

[124] Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, and Umeshwar Dayal. Multi-dimensional sequential pattern mining. In *Proceedings of*

*the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 81–88, New York, NY, USA, 2001. ACM.

[125] Todd Eavis and Xi Zheng. Multi-level frequent pattern mining. In Xiaofang Zhou, Haruo Yokota, Ke Deng, and Qing Liu, editors, *Database Systems for Advanced Applications*, pages 369–383, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[126] Fan-Chen Tseng. Mining frequent itemsets in large databases: The hierarchical partitioning approach. *Expert Systems with Applications*, 40(5):1654 – 1661, 2013.

[127] André Melo, , and Johanna Völker and. Probabilistic frequent itemset mining with hierarchical background knowledge. *International Journal of Knowledge Engineering-IACSIT*, 1(2):92–99, 2015.

[128] Marijana Zekić-Sušac and Adela Has. Discovering market basket patterns using hierarchical association rules. *Croatian Operational Research Review*, 6(2):475–487, 2015.

[129] Ann Devitt, Joseph Duffin, and Robert Moloney. Topographical proximity for mining network alarm data. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data*, MineNet '05, pages 179–184, New York, NY, USA, 2005. ACM.

[130] W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business process mining: An industrial application. *Information Systems*, 32(5):713 – 732, 2007.

[131] A.L. Barabási and M. Pósfai. *Network Science*. Cambridge University Press, 2016.

[132] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report

1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[133] Loet Leydesdorff. Betweenness centrality as an indicator of the interdisciplinarity of scientific journals. *Journal of the American Society for Information Science and Technology*, 58(9):1303–1319, 2007.

[134] Dileep Buddaraju. Performance of control room operators in alarm management. Master's thesis, Louisiana State University and Agricultural and Mechanical College, 2011.

[135] Barry M. Wise, Neal B. Gallagher, Stephanie Watts Butler, Daniel D. White, and Gabriel G. Barna. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics*, 13(3-4):379–396, 1999.

[136] Feng Lin. Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems*, 4(2):197–212, May 1994.

[137] Mogens Blanke, Michel Kinnaert, Jan Lunze, and Marcel Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag Berlin Heidelberg, 3 edition, 2016.

[138] J. Zaytoon and S. Lafortune. Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control*, 37(2):308 – 320, 2013.

[139] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, Sep 1995.

[140] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Teneketzis. Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, 4(2):105–124, Mar 1996.

[141] Wei Guo and Ashis G. Banerjee. Identification of key features using topological data analysis for accurate prediction of manufacturing system outputs.

*Journal of Manufacturing Systems*, 43(Part 2):225 – 234, 2017. High Performance Computing and Data Analytics for Cyber Manufacturing.

[142] Yan Chen and Jay Lee. Autonomous mining for alarm correlation patterns based on time-shift similarity clustering in manufacturing system. In *2011 IEEE Conference on Prognostics and Health Management*, pages 1–8, June 2011.

[143] Jianfeng Zhu, Chunli Wang, Chuankun Li, Xinjiang Gao, and Jinsong Zhao. Dynamic alarm prediction for critical alarms using a probabilistic model. *Chinese Journal of Chemical Engineering*, 24(7):881 – 885, 2016.

[144] F. Yang, S.L. Shah, D. Xiao, and T. Chen. Improved correlation analysis and visualization of industrial alarm data. *{ISA} Transactions*, 51(4):499 – 506, 2012.

[145] Luyang Jing, Ming Zhao, Pin Li, and Xiaoqiang Xu. A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox. *Measurement*, 2017.

[146] Alessandro Lusci, Gianluca Pollastri, and Pierre Baldi. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *Journal of chemical information and modeling*, 53(7):1563–1575, 2013.

[147] Søren Kaae Sønderby, Casper Kaae Sønderby, Henrik Nielsen, and Ole Winther. Convolutional lstm networks for subcellular localization of proteins. In *International Conference on Algorithms for Computational Biology*, pages 68–80. Springer, 2015.

[148] John BO Mitchell. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4(5):468–481, 2014.

[149] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, and J. Wang. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, PP(99):1–1, 2017.

[150] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *CoRR*, abs/1607.00148, 2016.

[151] Shyamapada Mandal, B Santhi, Sridhar Sethumadhavan, K Vinolia, and P Swaminathan. Nuclear power plant thermocouple sensor-fault detection and classification using deep learning and generalized likelihood ratio test. PP:1–1, 04 2017.

[152] F. Lv, C. Wen, Z. Bao, and M. Liu. Fault diagnosis based on deep learning. In *2016 American Control Conference (ACC)*, pages 6851–6856, July 2016.

[153] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61(Supplement C):85 – 117, 2015.

[154] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[155] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[156] Oswaldo Ludwig, Xiao Liu, Parisa Kordjamshidi, and Marie-Francine Moens. Deep embedding for spatial role labeling. *arXiv preprint arXiv:1603.08474*, 2016.

[157] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[158] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[159] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):107–116, 1998.

[160] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.

[161] Gyula Dorgo, Ferenc Tandari, Tibor Szabo, Ahmet Palazoglu, and Janos Abonyi. Quality vs. quantity of alarm messages - how to measure the performance of an alarm system? *Chemical Engineering Research and Design*, ACCEPTED, 2021.

[162] Gyula Dorgo, Peter Pigler, Mate Haragovics, and Janos Abonyi. Learning operation strategies from alarm management systems by temporal pattern mining and deep learning. In Anton Friedl, Jiří J. Klemeš, Stefan Radl, Petar S. Varbanov, and Thomas Wallek, editors, *28th European Symposium on Computer Aided Process Engineering*, volume 43 of *Computer Aided Chemical Engineering*, pages 1003 – 1008. Elsevier, 2018.

[163] Gyula Dorgo, Ahmet Palazoglu, and Janos Abonyi. Decision trees for informative process alarm definition and alarm-based fault classification. *Process Safety and Environmental Protection*, 149:312–324, 2021.