Advanced Machine Learning-based Alarm Flood Monitoring Using Alarm Data

by

Haniyeh Seyed Alinezhad

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

Doctor of Philosophy

in

Control Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Monitoring industrial processes is essential to maintain efficient and safe operation. An alarm system is an imperative component of industrial process monitoring, as it alerts operators to abnormal conditions. Alarms indicate when the process is disrupted, which allows operators to react accordingly. A modern industrial plant consisting of many interconnections is susceptible to fault propagation via information and material flow pathways. During fault propagation, a large number of alarms are triggered in control rooms, which results in a phenomenon known as alarm flood. As a result, operators are overwhelmed with a high volume of alarms and may miss out on safety and efficiency measures. It is therefore crucial to deal with the problem of alarm floods, particularly for online applications. The development of efficient online alarm flood analysis can provide operator decision support for root cause analysis of abnormal events and prevent the occurrence of destructive effects. The large amount of data generated by modern computerized processes has recently led to considerable interest in data-based methods. Developing data-driven methods in the field of machine learning can reduce reliance on expert knowledge and human effort in online alarm monitoring. Therefore, this thesis focuses on the development of machine learning-based methods for alarm management and alarm flood monitoring using alarm data.

Our research primarily focuses on the investigation of methods for transforming alarm floods from time stamped alarm sequences into inputs suitable

for machine learning algorithms. As a result of making alarm floods compatible with machine learning, effective online operator assistance mechanisms can be implemented. We begin by developing a modified version of an alarm flood vector representation based on exponentially attenuated component analysis, which utilizes the time information of unlabeled historical alarm floods. To ensure safe and efficient operation, it is beneficial to classify ongoing alarm floods as early as possible. It can provide online decision support for plant operators to take timely action, without waiting for the end of an alarm flood. We propose an approach employing the Gaussian mixture model to address the early classification problem with unlabeled historical data. It includes two phases: offline clustering and online classification, where the clustering step is automated in terms of choosing the optimal number of clusters by applying an efficient cluster validity index.

In a plant operation, there can be alarm flood scenarios that correspond to previously unseen abnormal situations. Therefore, early online assistance for plant operators in both previously known and new situations is of great importance. To address this issue, we propose an operator assistance system that relies on similarity analysis of alarm floods and alarm scoring. First, inspired by natural language processing, a vector representation called the Modified Bag-of-Words is devised to turn alarm floods into feature vectors. Modified Bag-of-Words vectors are then used in an offline clustering algorithm for grouping similar alarm floods using efficient similarity measurement. Subsequently, we extend the study to the case of online alarm flood analysis, where an open set early classification method based on systematic similarity threshold estimation is proposed to handle the new alarm flood scenarios. These studies are based on an alarm weighting strategy reflecting the key features of alarm floods. It provides alarm ranking to assist operators in identify-

ing alarms relevant to specific abnormal situations in both offline and online applications.

Order ambiguity of alarms in the alarm sequences associated with alarm floods is an important problem that has not been extensively investigated. Finally, we developed a probabilistic framework capable of incorporating the triggered alarm tags and their timestamps, as well as tolerating the effect of irrelevant alarms and alarm order ambiguities. In this study, an ML-based alarm flood analysis is established, where a convolutional neural network is trained to predict upcoming fault scenarios by observing online alarm floods.

The proposed methods are evaluated through case studies using alarm datasets from the well-established Tennessee Eastman benchmark, a Vinyl Acetate Monomer process and an industrial facility. Comparative studies with state-of-the-art alarm flood analysis methods are also provided to show the effectiveness of the proposed approaches. In light of these findings, online operator assistance mechanisms can be implemented to provide early decision support and ensure safe and efficient plant operation.

Preface

The ideas in Chapters 2, 3, and 4 were from my discussions with Dr. Jun Shang and Prof. Tongwen Chen. The idea in Chapter 5 was evolved from my discussions with Prof. Sirish L Shah, Jun Shang and Tongwen Chen. The review of alarm root cause analysis methods presented in Chapter 1 was from a collaboration with Dr. Mohammad Hossein Roohi. The algorithms, the theoretical analyses, and the case studies in this thesis are part of my original work. My publications and their connections to this thesis are summarized below.

- An overview of alarm root cause analysis methods presented in Chapter 1 has been published as: Haniyeh Seyed Alinezhad, Mohammad Hossein Roohi, and Tongwen Chen, "A review of alarm root cause analysis in process industries: Common methods, recent research status and challenges," *Chemical Engineering Research and Design*, vol. 188, pp. 846–860, 2022.

- Chapter 2 has been published as: Haniyeh Seyed Alinezhad, Jun Shang, and Tongwen Chen, "Early classification of industrial alarm floods based on semisupervised learning," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1845–1853, 2022.

- Chapter 3 has been published as: Haniyeh Seyed Alinezhad, Jun Shang, and Tongwen Chen, "A modified Bag-of-Words representation for industrial alarm floods," *in 9th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, Vancouver, BC, August 2022.

- Chapter 4 has been published as: Haniyeh Seyed Alinezhad, Jun Shang, and Tongwen Chen, "Open set online classification of industrial alarm floods with alarm ranking," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. article 3500811, 2023.

- Chapter 5 has been submitted for publication as: Haniyeh Seyed Alinezhad, Jun Shang, Tongwen Chen, and Sirish L. Shah, "A probabilistic framework for online analysis of alarm floods using convolutional neural networks," *IEEE Transactions on Instrumentation and Measurement*.

To *My Family*

Acknowledgements

I would like to take this opportunity to extend my sincere appreciation to many individuals in my research and personal life for their help and support. First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Tongwen Chen, for his guidance, patience, and support in achieving my research goals. The completion of this thesis would not have been possible without his encouragement, suggestions and rigorous research attitude. Along the same note, I would also like to thank Prof. Sirish L. Shah for offering his precious time and valuable comments on my work.

Also, I would like to extend special thanks to my supervisory committee members Prof. Qing Zhao and Prof. Mahdi Tavakoli for their valuable time and feedback.

I would like to express my thanks to my colleagues and collaborators Dr. Jun Shang and Dr. Mohammad Hossein Roohi for the numerous discussions we had and for their suggestions and comments on my work. I would also like to thank all current and previous members of our research group, especially Boyuan, Mani, Jing, Li, Hari, Junyi, Iman, Donny, Rezwan, Habib, Ziyi, and Jinyuan for their kindness, conversations, encouragement, and help.

I also wish to acknowledge the financial support from Natural Sciences and Engineering Research Council of Canada (NSERC) and our industrial partners.

I would also like to express my warmest appreciation for everything that my family has done for me, especially my beloved Mom and Dad; this thesis is dedicated to them. Finally, and on a personal note, I want to express my heartfelt gratitude to my husband, Amirhossein, for his unwavering support

throughout my academic journey. Your encouragement, love, and patience have been a constant source of strength, and I could not have completed this thesis without you by my side.

Contents

List of Tables

List of Figures

List of Acronyms

A&E	Alarm and Event
AHC	Agglomerative Hierarchical Cluster
ALSTM	Attention-based Long Short-Term Memory
ANN	Artificial Neural Network
AR	Auto-Regressive
BN	Bayesian Network
CBN	Causal Bayesian Networks
CCF	Cross-Correlation Function
CNN	Convolutional Neural Network
CPDF	Conditional Probability Distribution Function
DBN	Dynamic Bayesian Network
DTE	Direct Transfer Entropy
EAC	Exponentially Attenuated Component
EM	Expectation–Maximization
GC	Granger Causality
GMM	Gaussian Mixture Model
GPR	Gaussian Process Regression
HMM	Hidden Markov Model
IDF	Inverse Document Frequency

LR	Logistic Regression
LSTM	Long Short-Term Memory
MBoW	Modified Bag-of-Words
MBTE	Multiblock Transfer Entropy
MFM	Multilevel Flow Models
MGC	Multivariate Granger Causality
ML	Machine Learning
MSW	Modified Smith–Waterman
NLP	Natural Language Processing
NMI	Normalized Mutual Information
OOBN	Object Oriented Bayesian Networks
P&ID	Piping and Instrumentation Diagrams
PDF	Probability Density Function
PGM	Probabilistic Graphical Models
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SDG	Signed Directed Graph
SDNTE	Symbolic Dynamic-based Normalized Transfer Entropy
T0E	Transfer 0-Entropy
TE	Transfer Entropy
TEP	Tennessee Eastman Process
TF	Term Frequency
TTE	Trend Transfer Entropy
VAM	Vinyl Acetate Monomer

Chapter 1

Introduction [*]

This chapter introduces the research background on alarm flood monitoring and root cause analysis, and provides a mathematical overview of three data-driven strategies that have been commonly used to analyze the root causes of abnormal conditions. Subsequently, a literature survey is presented to summarize recent developments in root cause analysis and alarm flood management in abnormal situations. Thereafter, the contributions of the thesis are highlighted, and a thesis outline is provided.

1.1 Research Background

Safety and reliability have become increasingly important in modern industrial processes. There have been numerous methods proposed in recent years that cover process monitoring [70, 115], fault diagnosis [49, 58], and remaining useful life prediction [114]. These methods were designed to improve the health management of industrial equipment and the smooth operation of the system. In all aspects of system operation, indicating whether the system is functioning properly or not is critical. These functions are often integrated into an alarm system, which is a crucial asset in industrial process monitoring.

[*]An overview of alarm root cause analysis methods presented in this chapter has been published as: Haniyeh Seyed Alinezhad, Mohammad Hossein Roohi, and Tongwen Chen, "A review of alarm root cause analysis in process industries: Common methods, recent research status and challenges," *Chemical Engineering Research and Design*, vol. 188, pp. 846–860, 2022.

Alarm systems serve as warning mechanisms to notify plant operators about possible failure or performance degradation so that they can take the necessary safety measures. Alarms are configured on process variables to indicate abnormal conditions, when measurements deviate from predetermined normal operating ranges. In modern process industries, process monitoring systems such as distributed control systems (DCSs) and supervisory control and data acquisition (SCADA) systems are used to control processes. Integrating these computerized monitoring systems into industrial facilities has led to highly automated operations. Process automation has facilitated alarm configuration in alarm systems and has been beneficial in improving plant efficiency and safety compared with high-cost hardware-based monitoring systems [46]. However, owing to the enormous amount of process data in large-scale plants, easier alarm configuration in computerized alarm systems increases the number of alarm variables considerably. In addition, complex physical connections in large-scale industrial plants may cause the propagation of faults and abnormalities leading to sequences of alarm annunciations. Consequently, plant operators are overloaded with an excessive number of triggered alarms in a short period of time preventing them from taking timely safety precautions [102]. Therefore, investigating methods for supporting on-site operators in case of alarm overloading has recently become an important topic in the area of alarm management.

1.2 Alarm Floods

Plants are usually composed of numerous interconnected devices and multiple control loops. In case of failure, complex connections in modern industrial processes lead to the propagation of faults along the material and information flow pathways. This is the main cause of a situation called alarm flood, which is a special case of the alarm overloading phenomenon [102]. It refers to a situation with a large number of consecutive alarm annunciations in a short period of time, where operator action is needed for the sake of efficiency and

safety. Alarm floods could distract plant operators from the root cause of an abnormal situation and may lead to serious safety issues. Hence, developing methods to assist operators in case of alarm floods has been of significant importance for safe plant operation.

1.2.1 Root Cause Analysis

The development of methods for root cause analysis of abnormal conditions can provide plant operators with valuable support when they are overloaded with alarms. The existing methodologies in the literature can generally be classified into two types, namely, knowledge-based methods and data-based methods. In knowledge-based methods, a root alarm is identified mainly based on qualitative process information and expert knowledge. These methods try to find the fault propagation paths among different units of the process, which can be used for the localization of the root cause in certain abnormal situations and, thus, identifying the most plausible root cause. Efficient qualitative models such as multilevel flow models (MFMs) [56], signed directed graphs (SDGs) [98] and adjacency matrices [51] were adopted to provide a representation for process connectivity. An MFM provides a straightforward graphical model to describe the plant objectives, functions, and causal relations among them. Process knowledge including piping and instrumentation diagrams (P&IDs) and differential equations are utilized to model SDGs. An SDG is used to represent causal relations in processes as a graphical model, which shows the information flow as well as the cause-effect direction. Another technique to model plant connectivity and causality is by using the concept of an adjacency matrix, which is a common representation for directed graphs. Kirchhübel *et al.* introduced a method to represent the knowledge of an industrial process based on modeling the mass and energy flows to be used for causal discovery in case of failures [56]. In their method they identified propagation rules of abnormalities and built a failure tree, which could show the root cause and possible direction for fault propagation. The concept of

adjacency matrices was used in [51] for root cause diagnosis by converting the process schematic to a directed graph. Wan *et al.* proposed a method that first identified a list of potential root nodes using qualitative reasoning on SDG; Next, quantitative statistical tests were applied to those nodes and finally a cause-effect graph was obtained [98]. Knowledge-based methods exploit the process model to capture the cause-and-effect relationships in the process, which are used for root cause analysis. Although process knowledge is a reliable source for alarm root cause identification, obtaining accurate process information depends on expert knowledge that is not always easily available. Obtaining process knowledge for the large-scale complex processes is a time-consuming and difficult task. Thus, this class of methods is more appropriate for analyzing alarm root causes in processes that have relatively low complexity.

Data-based root cause analysis methods have recently received considerable attention due to large amount of data in modern computerized processes. This class of methods investigates measured process variables or alarm data to uncover the root cause of abnormal conditions and reduce dependency on expert knowledge and human effort for process monitoring. Data-based root cause analysis methods can be viewed as different categories, namely, times series causality analysis, probabilistic graphical models (PGMs), machine learning and other data-driven strategies. In the area of times series causality analysis, Granger causality (GC) and transfer entropy (TE) are the two widely used methods for alarm root cause analysis. GC is a concept originated from econometrics, and investigates the causal associations between time series [37]. TE is an information-theoretic data-driven approach for testing the causality between two variables, which is defined based on the information entropy proposed by Shannon [87]. The most popular types of PGMs utilized in root cause analysis area are Bayesian networks (BNs) that are directed acyclic PGMs [74]. There also exist approaches exploiting different machine learning (ML) methods and other data-driven strategies such as correlation analysis,

nonlinearity indices, and nearest neighbors. There exist methods that utilize process knowledge in combination with process data as a complimentary source for root cause analysis. Knowledge-based causal maps combined with multivariate statistics were used by Chiang *et al.* for analyzing the causal dependency between measured process variables [23]. In [94], a cause-and-effect matrix derived from process measurements was combined with qualitative information of the process. Schleburg *et al.* integrated the process knowledge and connectivity information with alarm data [82]. Although process knowledge is a reliable source for alarm root cause identification, obtaining accurate process information for large-scale complex processes is a time consuming and difficult task and depends on expert knowledge that is not always easily available. To overcome this limitation and thereby reduce the reliance on human effort, data-based root cause analysis methods have been established for process monitoring. The following provides a mathematical background to three data-driven strategies that have been commonly used for root cause analysis, namely GC, TE, and BN.

Granger Causality

The underlying principle of Granger causality (GC) is based on predictability improvement. Consider the time series X_1 and X_2. X_1 is a Granger cause of X_2 if future values of X_2 can be predicted better when its past values are used combined with the past values of X_1 (in comparison with the case that the prediction is solely based on X_2). The Granger's definition of causality is based on temporal precedence, which assumes that the cause occurs before its effect. This could be similar to a situation in process industries, where a fault appears in a process variable and then propagates to other variables with time lags. Process variables can reflect the characteristics of the industrial process. Thus, causality analysis of process variables, which are in the form of time series data, could provide useful information for diagnosing the root cause variable corresponding to the original fault and its propagation path. Several data-driven methods using GC have been proposed for root cause analysis

based on the causality inference between process variables.

Conventional GC Test The most common mathematical interpretation of the GC is based on bivariate and univariate auto-regressive (AR) models for time series. Consider two stationary time series $X_1 = \{x_1(1), x_1(2), \cdots, x_1(n)\}$ and $X_2 = \{x_2(1), x_2(2), \cdots, x_2(n)\}$. They can be expressed by the following bivariate AR models (also known as full models):

$$x_1(t) = \sum_{p=1}^{L} \alpha_{11,p} x_1(t-p) + \sum_{p=1}^{L} \alpha_{12,p} x_2(t-p) + \varepsilon_{12}(t) \tag{1.1}$$

$$x_2(t) = \sum_{p=1}^{L} \alpha_{21,p} x_1(t-p) + \sum_{p=1}^{L} \alpha_{22,p} x_2(t-p) + \varepsilon_{21}(t) \tag{1.2}$$

The corresponding univariate AR models (or reduced models) are defined as follows:

$$x_1(t) = \sum_{p=1}^{L} \beta_{1,p} x_1(t-p) + \varepsilon_1(t) \tag{1.3}$$

$$x_2(t) = \sum_{p=1}^{L} \beta_{2,p} x_2(t-p) + \varepsilon_2(t) \tag{1.4}$$

where, $\alpha_{ij,p}$ and $\beta_{i,p}$ are the coefficients of the AR models, ε_{ij} shows the prediction errors or residuals of the full AR model, and ε_i represents the residuals of the univariate AR model, which is used to predict the current value of signal X_i considering only past values of itself. The order of AR model, denoted as L, is the time lag length, which defines the number of historical values from time series used for prediction.

The variability of model error reflects prediction accuracy, which can be utilized to quantify the evaluation of Granger causality relationship between time series as follows:

$$F_{X_j \to X_i} = \ln \frac{var(\varepsilon_i)}{var(\varepsilon_{ij})} \tag{1.5}$$

Comparing the variance of residuals, if $var(\varepsilon_{ij}) \leq var(\varepsilon_i)$, there is improvement in predicting X_i including the past information of X_j, where $i = 1, 2$, $j = 1, 2$ and $i \neq j$. Accordingly, when $F_{X_j \to X_i} \geq 0$, X_j is the Granger cause

of X_i. Otherwise, there is no Granger causality between time series. Note that, $F_{X_j \to X_i}$ can never be negative. To test whether X_j Granger-causes X_i, the null hypothesis, which aims at discarding the possibility of adding predictive power by X_j, must be rejected. The statistical significance of this causal influence can be tested using the following F-statistic:

$$F_{statistic} = \frac{(RSS_0 - RSS_1)/L}{RSS_1/(N - 2L - 1)} \sim F(L, MN - 2L - 1) \tag{1.6}$$

where RSS_0 and RSS_1 are the residual sum of squares in the reduced model and full model respectively, and N denotes the total number of observations used to build the model. If the null hypothesis is rejected with a significance level α for the distribution F, then X_j is said to be the Granger cause of X_i.

Multivariate GC In process industries, the two time series used in GC-based formulation are two candidate process variables. Process variables are mostly correlated due to multisource correlations and interconnections in large-scale processes. This could make the conventional pairwise GC inefficient for an accurate root cause analysis. The conditional GC or multivariate Granger causality (MGC), which is a generalization of the bivariate GC is used to address this problem [16]. For root cause analysis using multivariate GC, multiple candidate process variables are simultaneously integrated into the AR model such that the full AR model is defined as follows:

$$x_1(t) = \sum_{p=1}^{L} \alpha_{11,p} x_1(t-p) + \sum_{p=1}^{L} \alpha_{12,p} x_2(t-p) + \sum_{j=3}^{J} \sum_{p=1}^{L} \alpha_{1j,p} x_j(t-p) + \varepsilon_{12}(t) \tag{1.7}$$

$$x_2(t) = \sum_{p=1}^{L} \alpha_{21,p} x_1(t-p) + \sum_{p=1}^{L} \alpha_{22,p} x_2(t-p) + \sum_{j=3}^{J} \sum_{p=1}^{L} \alpha_{2j,p} x_j(t-p) + \varepsilon_{21}(t) \tag{1.8}$$

Also, the corresponding reduced AR models are defined as follows:

$$x_1(t) = \sum_{p=1}^{L} \beta_{1,p} x_1(t-p) \; + \sum_{j=3}^{J} \sum_{p=1}^{L} \beta_{1j,p} x_j(t-p) + \varepsilon_1(t) \tag{1.9}$$

$$x_2(t) = \sum_{p=1}^{L} \beta_{2,p} x_2(t-p) \; + \sum_{j=3}^{J} \sum_{p=1}^{L} \beta_{2j,p} x_j(t-p) + \varepsilon_2(t) \tag{1.10}$$

Here, J represents the total number of variables under consideration. For root cause analysis using the multivariate GC, direct GC relationships for all variable pairs are determined by building the above AR models and repeating the GC test proposed in (1.6).

Transfer Entropy

By quantifying variable uncertainty using the Shannon entropy, the concept of transfer entropy (TE) was proposed by Schreiber to measure the information exchange between two variables [83]. This work provides the information flow by measuring the uncertainty reduction of one variable under the influence of another variable. Compared with GC, the major advantage of TE is that it is suitable for causality inference for both linear and nonlinear relationships. Thus, TE can be considered as a useful method for causality analysis in industrial processes using both process variables and alarm variables.

Definition of TE Measuring information flow in a process can show how the variation in process variables transfers from one variable to another. Therefore, application of TE using process variables has attracted the attention of researchers and was successfully used by Bauer *et al.* to address the problem of causality analysis in chemical processes [12].

Let X_1 and X_2 denote two stationary time series obtained by sampling two continuous process variables $x_1(t)$ and $x_2(t)$ at time instances t with $t = 1, 2, 3, n$. The transfer entropy from time series X_1 to X_2 is defined as

$$\begin{aligned} T(X_1|\, X_2) \; &= \sum p\left(x_1(t+h),\, \chi_1^k(t),\, \chi_2^l(t)\right) \\ &\qquad .\log_2 \frac{p\left(x_1(t+h)|\, \chi_1^k(t),\, \chi_2^l(t)\right)}{p\left(x_1(t+h)|\, \chi_1^k(t)\right)} \\ &= \; H(x_1(t+h)|\, \chi_1^k(t)) \; - H\left(x_1(t+h)|\, \chi_1^k(t),\, \chi_2^l(t)\right) \end{aligned} \tag{1.11}$$

where $p(\cdot)$ and $p(\cdot\,|\,\cdot)$ denote the joint and conditional probability density functions (PDFs), respectively; h is called the prediction horizon, and time instant $t+h$ means h steps in the future from t; $\chi_1^k(t) = \{x_1(t), x_1(t-\tau), \cdots, x_1(t-(k-1)\tau)\}$ and $\chi_2^l(t) = \{x_2(t), x_2(t-\tau), \cdots, x_2(t-(l-1)\tau)\}$ are referred to as embedding vectors including the past values of X_1 and X_2, respectively; the integer τ is the sampling period; the integers k and l are the dimensions of the historical vectors $\chi_1^k(t)$ and $\chi_2^l(t)$, respectively; the sum symbol represents $k+l+1$ sums over all amplitude bins of the probability distribution functions; $H(\cdot)$ denotes the Shannon entropy.

The mathematical definition proposed in (1.11) is based on conditional probabilities that contain causal information between two variables. According to this definition, if there exists causality from X_1 to X_2, then in the argument of the logarithm, the conditional probability in the numerator is greater than that of the denominator and TE measure is greater than zero; otherwise, the value of $T(X_1|X_2)$ is zero. From Shannon entropy's point of view, TE represents the transferred information from X_1 to X_2 by measuring the reduction of the uncertainty when predicting a future observation of variable X_1 with the help of the historical values of both variables X_1 and X_2, instead of using only the past values of X_1 (Figure 1.1). TE is an asymmetric method for distinguishing the causality between variables. Therefore, in [12] a causality measure for determining the direction and quantity of information transfer was defined as $t_{X_1\to X_2} = T(X_1|\,X_2) - T(X_2|\,X_1)$. According to this definition, positive values of $t_{X_1\to X_2}$ represent that X_1 is the cause of X_2, negative values represent the reverse case and zero means no causality. It can be verified that the basic concept of TE is similar to that of GC. It was shown by Barnett *et al.* that GC and TE are equivalent for Gaussian distributed variables with linear relationships [10].

Estimation of TE For calculating the TE using (1.11), conditional PDFs are replaced by joint PDFs based on the Bayesian principle. Then, the value of TE is calculated by estimating the PDFs from time series X_1 and X_2 via the

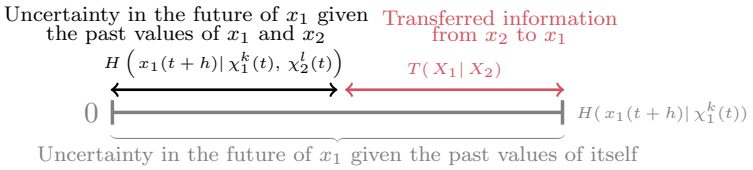

Figure 1.1: Interpretation of TE using the concept of Shannon entropy.

histogram or kernel method [90], which are nonparametric approaches utilized for fitting distributions. The kernel method has been widely used for PDF estimation in TE estimation due to its more robust and precise estimation compared with the histogram method. Moreover, because of the high order of PDFs, an extremely large number of samples is required in the histogram method that makes the kernel method [12, 27] advantageous.

In the kernel method, a univariate PDF is estimated by the following kernel estimator:

$$\hat{p}(x) = \frac{1}{n}\sum_{t=1}^{n} K\left(x - x(t)\right) \tag{1.12}$$

The kernel function K is centered at every sample point and summed to estimate the PDF. The Gaussian kernel is mostly used in TE methods, which is defined as:

$$K\left(x - x(t)\right) = \frac{1}{\sqrt{2\pi\theta}} \exp\left(-\frac{\left(x - x(t)\right)^2}{2\theta^2}\right) \tag{1.13}$$

Here θ is the estimator width defined as $\theta = 1.06\sigma N^{-1/5}$, where σ is the standard deviation of data samples [12]. θ is chosen such that the mean integrated squared error of the PDF estimation is minimized. A multivariate J-dimensional joint PDF can be estimated in a similar manner by using the following kernel estimator:

$$\hat{p}(x_1, x_2, \cdots, x_J) = \frac{1}{n}\sum_{t=1}^{n} K\left(x_1 - x_1(t)\right) K\left(x_2 - x_2(t)\right) \cdots K\left(x_J - x_J(t)\right) \tag{1.14}$$

In this case, the estimator width parameter for each univariate kernel function is defined as $\theta_i = 1.06\sigma_i N^{-1/(4+J)}$, where $i = 1, 2, \cdots, J$ and σ_i is the standard deviation of data samples of the ith variable [12, 90].

There are four adjustable parameters, namely k, l, h and τ, that could greatly affect the estimation results of the TE and need to be determined before calculating TE. Parameters k and l should be carefully chosen as the computational complexity and the number of samples required for PDF estimation are highly dependent on the values of the embedding dimensions. Therefore, embedding dimensions should be selected as small as possible. On the other hand, Schreiber suggested selecting a greater value of l compared to k to focus more on the effect of X_2 on X_1 rather than the effect of X_1 on X_1 [83]. Following this suggestion, efficient values were determined for these parameters in several studies like [12]. A more systematic method for determining embedding dimensions was also proposed in [27, 112], which was based on the change rate of the conditional Shannon entropy. As stated in [12], the optimal values of h and τ can be determined based on the prior knowledge of the process dynamics. As the TE measure requires robustness for parameter changes [12], small values for prediction horizon h and sampling period τ could lead to reliable results if the process dynamics are unknown. Duan *et al.* recommended to choose h and τ as a rule of thumb [27] and a modified mutual information (MI) was proposed in [92] to determine the values for the prediction horizon and sampling period.

Significance Test The TE measure shown in equation (1.11) is zero if there is no causal relationship between the corresponding process variables. However, in real industrial processes, the TE measure for variables without causality may not be exactly zero owing to noises or disturbances. Thus, a threshold should be obtained for causality significance level to distinguish the true causal relationships from the false results. The Monte Carlo method was suggested in a majority of studies to identify such a threshold. This method provides a significance test by constructing surrogate data with some desired property.

Surrogate data can be constructed by randomly disorganizing original time series or by iterative amplitude adjusted Fourier transform (iAAFT) [27]. The significance test is defined based on rejecting the null hypothesis, which means there is no significant causality between time series. To this end, surrogate time series are generated from N_s simulations such that the causality in new data is destroyed, and the causality measure for each pair of new time series is calculated as

$$\lambda_i = t^i_{X_1^{New} \to X_2^{New}} \tag{1.15}$$

Then the threshold, above which $t_{X_1 \to X_2}$ is identified as a valid causality relationship, is defined based on the mean value and standard deviation of λ_i's with $i = 1, 2, \cdots, N_s$. Denoting the mean and the standard deviation as μ_λ and σ_λ respectively, the significance level is defined as $s_{X_1 \to X_2} = \mu_\lambda + \gamma\, \sigma_\lambda$, where $\gamma = 3$ and $\gamma = 6$ are two values that have been mostly used in the literature. When the estimated causality measure for X_1 and X_2 exceeds this threshold, the null hypothesis is rejected, which means there is significant causality between time series.

Bayesian Networks

Bayesian networks (BNs) are directed acyclic probabilistic graphical models consisting of nodes and arcs. In root alarm identification using BNs, nodes are corresponding to the alarm tags $\mathcal{A} = \{A_1, A_2, \cdots, A_N\}$ and their causal relationships are shown by arcs. Considering a Bayesian network $\mathcal{G}$, the joint probability of random variables can be simplified to

$$p(\mathcal{A}|\mathcal{G}) = \prod_{A_v \in \mathcal{A}} p(A_v | A_{\mathrm{pa}_v}), \tag{1.16}$$

where A_{pa_v} denote the *parent* set of A_v, which are the nodes with an edge toward A_v, and $p(A_v|A_{\mathrm{pa}_v})$ represent the conditional probability distributions. By learning an underlying BN from observational data (as opposed to interventional data) one can narrow down the possible causal relationships among alarm tags to some Markov equivalence class. Before formally defining the

Markov equivalence class, let us illustrate it with an elementary example. Assuming that analysis of data regarding two alarm tags A_1 and A_2 indicates that they are correlated. This investigation encodes both $A_1 \rightarrow A_2$ and $A_2 \rightarrow A_1$. This implies that these two structures are not distinguishable using only the observational data. More generally, two graphs are Markov equivalence if they possess the same set of links (arcs without direction) and v-structures [97]. For three nodes A_1, A_2, A_3 belonging to graph G, a v-structure is defined as $A_1 \rightarrow A_2 \leftarrow A_3$. As a trivial example, $A_1 \rightarrow A_2$ and $A_1 \leftarrow A_2$ are Markov equivalent as they are free of v-structures and in both graphs, A_1 and A_2 are directly connected.

Methods introduced in the literature for learning BNs from data can be categorized as constraint-based, optimization-based (also known as score-based) and hybrid approaches. Constraint-based methods determine dependencies among nodes by indicating some arc directions regarding v-structures. More details on this approach can be found in [84].

Constraint-Based These methods generally consist of two main steps:

- Identification of links between each pair of variables in the network by performing conditional independence tests.
- Obtaining directions of some of the arcs based on a set of rules [24].

It is important to note that the second step does not necessarily indicate the direction of all arcs, which results in a set of graphs that are a subset of a Markov equivalence class. Further details on constraint-based approaches can be found in [91].

Optimization-Based In these methods, a score is defined, which represents goodness of fit for each BN structure, and the one with the highest score is selected. The score is defined as $p(\mathcal{G}|\mathcal{A})$. Using the Bayes' rule, multiplying this score to $\frac{p(\mathcal{A})}{p(\mathcal{G})}$ yields $p(\mathcal{A}|\mathcal{G})$. We assume no expert knowledge is available

on structure $\mathcal{G}$. So to maximize the score, it is equivalent to maximize $p(\mathcal{A}|\mathcal{G})$, which is calculated as

$$p(\mathcal{A}|\mathcal{G}) = \int p(\mathcal{A}|\mathcal{G}, \Theta)p(\Theta|\mathcal{G})\mathrm{d}\Theta \tag{1.17}$$

where Θ indicate parameters of the Bayesian network.

Similar to the constraint-based methods, this method can discover up to some Markov equivalence class of structures that have the same score. Details of this method can be found in [25].

Hybrid Methods Hybrid algorithms inherit from both of the above approaches. They use the conditional independence test of constraint-based algorithms to narrow down candidate DAGs. Then, an optimization-based algorithm is applied to those DAGs to identify the optimal one. These methods are commonly assumed to be faster and/or more accurate than the optimization-based or constraint-based algorithm. However, a comparison performed by [85] reveals that the swiftness and accuracy of structure learning are related more to the selected statistical criteria than the algorithms themselves.

1.2.2 Data-Based Alarm Flood Analysis

The majority of existing data-driven approaches that address the problem of root cause analysis and operator assistance are based on process data. The use of alarm data is an alternative to using process data in alarm flood monitoring and root cause analysis. An alarm is raised only in the event of a fault occurrence and contains useful information regarding abnormal conditions. However, process data is generated by regularly monitoring and measuring process variables during process operation. Therefore, the volume of alarm data is generally lower than that of process data, which leads to lower computational complexity in data-based methods which is of great importance for online applications. The data collected about historical alarms can provide valuable insight into abnormal situations. In general, alarm floods resulting from the same abnormality consist of common alarms that are raised

chronologically in a similar order. Comparing similarities among alarm flood data has been shown to be a useful means of handling alarm flood situations [2,33]. It has been shown that pattern mining methods can be effectively used to analyze the similarity among alarm sequences and alarm flood management [22,38,45,47,61,75,118]. Most research in this area has focused on offline alarm flood analysis, which is aimed at investigating data from alarm and event (A&E) logs. By utilizing these methods, plant operators can be provided with valuable information regarding alarm patterns that can assist them in root cause analysis and making decisions. However, developing online decision support mechanisms to reduce the reliance on operator workload and involvement is of great importance for managing ongoing alarm floods safely [62,77]. ML has become increasingly appealing in many areas of research due to its ability to reduce the human effort involved in performing complex tasks accurately. In online applications, ML-based data classification techniques can be employed to design effective alarm flood management solutions. This concept has been used in the literature to come up with fault classification techniques that rely on process data [8,15,67,104]. Making use of alarm data to develop ML-based classification methods for alarm flood management can lead to effective operator assistance solutions. Categorizing online alarm floods into groups of historically similar alarm floods can help operators in identifying the underlying cause of the current alarm floods [32,68]. Moreover, to prevent ongoing alarm floods from turning into major incidents and process failures, it is imperative to provide early assistance to plant operators [86].

Motivated by the aforementioned considerations, this thesis proposes several advanced machine learning-based methods using alarm data to develop operator assistance mechanisms for early online alarm flood management and root cause analysis.

1.3 Literature Review

This thesis focuses on the development of online operator assistance mechanisms for early online alarm flood management and root cause analysis. This section provides a comprehensive overview of the research status on alarm flood monitoring and root cause analysis focusing on state-of-the-art data-based studies.

1.3.1 Causal Inference Incorporating GC

The phenomenon of plant-wide oscillations, resulting from plant and control interactions, is a common performance degradation in processes with closed-loop control systems. Root cause diagnosis of plant-wide oscillations based on MGC was studied in [113], where principal component analysis (PCA) was employed as a pre-processing step to exclude the process variables irrelevant to the oscillations from root cause analysis. Chen *et al.* proposed a grouping MGC (GMGC) method to address the root cause analysis of multiple plant-wide oscillations in process control systems with time-varying oscillations [18]. The authors combined MGC with a grouping strategy based on multivariate nonlinear chirp mode decomposition (MNCMD), which was utilized to detect plant-wide oscillations and cluster process variables with similar oscillations into the same group. Then, Using MGC, the causal relationships between the variables with similar oscillation frequencies were evaluated to locate the root causes. They aimed at capturing a clearer causal network and calculation efficiency by avoiding causality analysis for all oscillating variables in plants with multiple oscillation frequency components.

In [44], GC was used to address the problem of alarm root cause diagnosis for petrochemical plants, where the process variables that could be the possible cause of a triggered alarm were selected based on prior knowledge of interactions and relationships among process parameters. The fault propagation path corresponding to the occurred alarms was determined by causality inference between process variable time series using the GC test. The process

variable located at the end of the path was then identified as the root cause of the current fault. To improve the accuracy of GC-based causality analysis in batch processes, a root cause diagnosis approach named comparative Granger causality (CGC) was proposed in [30], where process variables were divided into time series slices to perform causality analysis. By performing the GC test on time slices, a sequence of causality values for any pair of variables was obtained, which was used to identify the abnormal causalities based on comparative results. The variable with the greatest number of abnormal causalities was then identified as the root cause variable.

The MGC method was used in [78] for root cause diagnosis via constructing the causality matrix for the magnitude variables affected by the fault. Hierarchical magnitude sensors were identified for the process faults by analyzing the fault contributions based on singular value decomposition (SVD). In an MGC-based root cause analysis of the abnormalities in multivariate industrial processes, repeated causality analysis between all pairs of process variables is needed. The correlation among process variables can result in a complex causal map, which makes it difficult to analyze the cause-and-effect relationships accurately. To facilitate root cause analysis, a simplified causal map was developed in [17], where the maximum spanning tree was found for a causal map derived using a conditional GC test. The causal map was constructed by utilizing graph theory, where the causality strength between process variables in the causal graph was represented by employing the F-statistic used in the conditional GC test as the weight of each edge. The maximum spanning tree was performed as a simplification step to determine the most significant causal sub-graph by eliminating unnecessary links from the original causal graph.

It should be noted that the GC-based methods discussed above are based on the linear regression of stationary time series data. To apply GC method on non-stationary data either a transformation (removing the trend) needs to be performed to make the data stationary or the first (or a higher) difference of the data should be used. However, there could be situations in which

linearity and stationarity assumptions are not applicable to complex industrial processes. Thus, it is necessary to develop methods to overcome linearity and stationarity restrictions. In [88], an extension of the conventional GC, called Copula-based GC, was proposed to obtain the causality analysis for stationary and nonlinearly related process signals. In this method, the GC was combined with the Copula function from the field of statistics so that by converting the GC into a log likelihood ratio, an estimation was derived via the conditional copula. For root cause diagnosis of triggered alarms in chemical processes, a causality analysis framework based on a multivariate GC test and Gaussian process regression (GPR) was proposed in [16]. GPR was used to address the problem of GC-based causality inference for nonlinearly related and non-stationary time series. Alarm root cause analysis was conducted by constructing a causal map via determining the pairwise causal relationships between faulty process variables identified in a fault isolation pre-step. An important advantage of GPR-based GC to deal with non-stationary signals is that it avoids missing any trends from time series, which is a possible issue in the mostly utilized first order difference method [44].

1.3.2 Causality Analysis Using the Concept of TE

The concept of direct transfer entropy (DTE) was proposed in [27], to detect spurious causalities and differentiate between direct causal pathways and indirect causal pathways with some intermediate process variables. This study was based on the differential TE and aimed to improve the fault root cause analysis through reducing the number of connections in the causal map. In [28], a transfer 0-entropy (T0E) concept was proposed, which does not assume the existence of a well-defined probability distribution for process data. In addition to the T0E for capturing total causal relationships, a direct T0E (DT0E) was also developed to distinguish between direct and indirect causalities in multivariate cases. Although the stationarity of the data is not a necessary requirement in this method and there is no need for very large data

lengths, the captured causal relationships might be conservative. In [89], a modification was made to the TE to achieve an improved estimation of time delays in causal relationships. In this approach, for each pair of process variables, the maximum value of the measured TE over different values of the prediction horizon was defined as the causality strength and the corresponding prediction horizon was defined as the time delay.

By considering the trends of process variables in causality analysis, a trend transfer entropy (TTE) method was developed in [39] to represent trend causality instead of value causality for variables. The TTE is based on calculating the TE for a symbolic series generated via piece-wise linearization of original variables, which leads to the reduction of computational burden and robustness towards noise and data drifting. Aiming at providing a fast and efficient real-time root cause diagnosis, a symbolic dynamic-based normalized transfer entropy (SDNTE) method was proposed in [79], which was defined based on the concepts of Shannon entropy, time series symbolization, and xD-Markov machines. In comparison with conventional kernel-based methods, SDNTE requires a smaller amount of historical data and has a significantly lower computational burden when calculating transfer entropies. However, it cannot handle non-stationary process variables for root cause analysis. By incorporating the concept of information granulation as a data compression technique, a novel TE-based causality inference was proposed in [116] to address the computational complexity of TE in high-dimensional embedded spaces.

Compared with causality analysis methods that determine system dynamic relationships by using process data, alarm data-based approaches are more computationally efficient. Reference [112] was the first attempt to use binary alarm series directly for causality analysis of process variables based on TE, which could facilitate the management of alarms by inferring causal relationships in abnormal situations. The concepts of normalized TE (NTE) and normalized DTE (NDTE) with a modified statistical test for calculating the significance threshold were proposed in [48] to infer causal relationships based

on binary alarm data by taking into account the random occurrence delays and the mutual independence occurrences of alarms. By using alarm data, a modified conditional mutual information (CMI) was proposed in [92] to identify direct causal relationships from the causal map derived by TE, where a multi-valued alarm signal definition was proposed to provide more information for causality analysis. In [29], the alarm log file was divided into groups of timely close alarms, named timed-clusters, and similar clusters were further used for root cause analysis using TE.

1.3.3 Probabilistic Graphical Model

Probabilistic graphical models (PGMs) are probabilistic models, in which the conditional dependencies of random variables are expressed by directed or undirected graphs. Reference [107] presented a comparison among various probabilistic graphical models for root causes of alarms in case of alarm floods in industrial plants. The studied models are Markov chains, hidden Markov models, Bayesian networks, timed automata and restricted Boltzmann machines. The comparison reveals that Bayesian networks outperform others based on criteria including (but not limited to) capability of presenting causal relations, being trained from data, and scalability. However, Bayesian networks have more complex calculations in comparison to Markov chains and timed automata.

Bayesian Networks

By using alarm floods data, reference [108] evaluated major classes of Bayesian network learning algorithms, namely, score-based, constraint-based, and hybrid: the best result was achieved using a hybrid algorithm, which in turn came at the cost of longer computation time. As a continuum to [108], [109] assumed that a causal model for a plant was available, and investigated different inference methods to determine the respective root causes, including variable elimination, logic sampling and likelihood weighting. While the latter two approaches were approximate inference, they provided similar detection

accuracy as variable elimination. The paper also raised an important question about reliability of using plant's causal model in case that root alarm node was not the root of this model. In [35], the authors integrated kernel principal component analysis as a fault detection tool with a Bayesian network that represented the process knowledge. They also proposed a method to convert cyclic causal networks to acyclic ones so that they could be modeled as BNs. Reference [1] proposed an architecture to combine expert knowledge and alarm data for constructing a causal map, which aimed at finding the root cause in case of an alarm flood. In this approach, the Bayesian network structure was identified using a constraint-based method. Paper [55] used plant information to construct an MFM, which was a modeling approach for complex industrial processes. To build the model, the authors used piping, instrumentation and process flow information together with operation manuals. Based on this MFM they built a BN, which was finally used for causal inference. MFM was also used in [54], which was first transformed to a fault tree model and subsequently to a BN that facilitated root alarm analysis. Industrial plants contain control loops, which inevitably result in the existence of loops in the constructed causal models. Bayesian networks are acyclic models, which generally do not account for loops. As a solution, reference [119] devised a BN with cyclic structures such that each node was duplicated into two state nodes allowing the BN to implicitly represent cycles. Reference [20] followed a similar approach by adding a dummy node in case that the process knowledge revealed the existence of loops. Another shortcoming in some BN-based root alarm analysis research is the inability to handle the existence of multiple root alarms. In [103] an approach was proposed to overcome this issue by limiting space of possible structures to those with one child and multiple parent nodes. Reference [81] introduced the application of causal Bayesian networks (CBNs) for identification of causal relationships among alarm tags. The core concept that differentiates BNs and CBNs is intervention, which is the manipulation of nodes by some external agents. The authors studied the analogy between fault

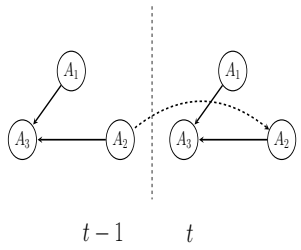

Figure 1.2: Illustration of dynamic Bayesian networks. The dashed arc shows the conditional dependency of A_2 at t given A_2 at $t-1$.

and intervention and formulated the root cause identification as discovering intervention of some unknown alarm tags. In [42], root alarms corresponding to the abnormal situation were identified by using the BN approach based on process variables, where hazard and operability analysis (HAZOP) were applied to determine the nodes of the model.

Dynamic Bayesian Networks

A dynamic Bayesian network (DBN) is a BN that is extended to model temporal dependencies of the corresponding stochastic variables at different instants of time. A DBN can be viewed as a sequence of BNs, in which each time slice represents the underlying system. There can be arcs from a time slice to its consecutive slices (see Figure 1.2) to describe temporal probabilistic dependencies of the variables.

DBNs are adopted by many researchers to address the causal relationship identification problem in case of abnormalities and hazards [4, 43, 117]. A comparison of DBN-based and BN-based root alarm analysis methods was provided in [5], which suggested the superiority of the DBN-based approach. Reference [64] used DBNs to deal with the problem of "phantom alarms", which are raised when a plant is in transition from abnormal to normal modes. In [106], the exploited DBNs were BNs that for each slice of time are object-oriented Bayesian networks (OOBNs). An OOBN adopts the ideal object-oriented programming for BNs, in which the OOBN contains instance nodes (in addition to ordinary nodes of BNs), which themselves are BNs. Reference

[111] combined process data with knowledge-based monitoring methods to construct a DBN to identify root cause as well as propagation paths of faults.

Combination of TE with BN

There are methodologies that combine the concepts of TE and BN for achieving a more efficient root cause analysis in industrial processes. To address the difficulties in BN learning, reference [73] integrated TE and BN to define the concept of a family transfer entropy (FTE) for alarm variables, which was used to develop a scoring function for BN learning. The multiblock transfer entropy (MBTE) method was proposed in [121] to define a scoring function for BN learning based on dividing complex processes into some submodules using process knowledge and searching the structure of each submodule according to the corresponding alarm data. Aiming at achieving a more accurate BN structure scoring, an improved TE was proposed to develop a MBTE-based Bayesian network. In [69], a novel multiblock BN model based on the concept of active dynamic transfer entropy (ADTE) was proposed to simplify the construction of BN structure and improve its accuracy, where the alarm propagation time could also be extracted. In [26], a modified K2 algorithm was proposed for BN structure learning, where TE was utilized to generate a causal graph for pre-ordering nodes within the network structure. To address the problem of cyclic causal relationships in process faults, a modified Bayesian network was developed in [60], where a TE-based score was used to quantify the causality strength and identify the weakest causal relationships in cyclic loops. The identified causal relations were converted into temporal relations, which were then used to decompose networks with cyclic loops into acyclic causal networks over time, leading to a more accurate root cause analysis.

1.3.4 ML-Based Approaches

ML algorithms uncover useful relationships among historical data to construct models for analyzing new data. The application of ML methods to

process measurements and alarm data can be beneficial to alarm root cause analysis. There are studies in the field of alarm management that take advantage of ML algorithms for alarm root cause analysis.

Machine learning can be used for causality analysis by exploring the correlations between sequentially dependent data from the process. It is capable of providing a means of identifying the cause of alarms through the investigation of the alarm propagation pathways. Recurrent neural networks (RNNs) are artificial neural networks (ANNs) that can capture sequential characteristics and patterns of data by investigating the temporal correlations between past experiences and current observations. An RNN is a predictive model that uses a self-learning mechanism to predict the correlations between its inputs and outputs. A Long short-term memory (LSTM) is an ANN architecture that follows a more complex and improved predictive framework when compared to an RNN. There are special blocks in LSTM known as memory blocks that give it the ability to detect the long-range dependencies of time series making it an improved alternative to RNN. An ML-based GC technique incorporating an attention-based LSTM (ALSTM) and MGC was developed in [40] to infer causal relationships among process variables and investigate the alarm propagation pathways. The proposed ALSTM model was created by adding an attention mechanism between the input and the first hidden layer of an LSTM network. It was used to convert the MGC AR model into a set of nonlinear ANN-based regression estimators using the time series data of the process variables as inputs. A soft attention for nonlinear causality modeling was developed by using the Softmax activation function, and a robust strategy was devised for training the model. This method can deal with long-term and varying transmission delays and make causality analysis more computationally efficient by avoiding nested loops. In addition, ALSTM addresses the problem of spurious causalities caused by unobservable variables via a sensitivity-based method and provides a simplified structure for alarm traceability. By utilizing model reduction and recursive variable selection, an algorithm for causality

analysis of process data based on the support vector machine (SVM) method was developed in [59]. Aiming at providing more evidence for the obtained causal relationships, GC, TE and cross-correlation function (CCF) methods were also applied and their corresponding results for causality strength between variables were combined into a causal matrix that was further used to generate a root cause priority list. Reference [9] devised a method based on decision trees. This method resulted in models with a high degree of intractability. As using a single tree may lead to an over-fitted model with high variance, the author also expanded the method to random forests. In [76] a map was created by extracting patterns, which related alarms to known faults using an alarm and event log. Patterns were then exploited for an online implementation step. If a match was found, the root cause could be determined based on the generated map.

Managing abnormal situations in process industries can be viewed as a ML-based classification problem. There have been several data-based approaches proposed in the literature, which utilize process measurements to devise fault classification mechanisms. A random forest technique was used in [15, 67] to develop classification methods based on process variables. The former was based on decision tree selection and a weighted voting rule, and the latter proposed an enhanced random forest algorithm incorporating static and dynamic information. ANNs are well-established models in the ML area that can be used for root cause analysis. This can be accomplished by building an ANN using historical data that characterize fault scenarios in an industrial process. This concept was employed in [8] to train a neural network based on process data for fault classification and root cause identification. In this approach, a specific number of samples from process variables were fed as inputs to an ANN with one hidden layer including ReLU activation functions. A fault scenario was identified as an output for the proposed ANN-based fault classification model, where a permutation algorithm was utilized to determine the root cause of the detected faulty condition by investigating the most

contributed process variable for the classified fault. In [104], a deep neural network framework was devised to learn fault-relevant features from raw input data by using a supervised stacked auto-encoder, where the output was the fault type label. The authors in [66] proposed two methods for root cause analysis of anomalies called sequential state switching and artificial anomaly association. The former was based on restricted Boltzmann machines (RBMs) and the latter could be viewed as a classification approach based on deep neural networks (DNNs), which refer to ANNs with multiple hidden layers. This approach involved a supervised learning strategy to identify the relationships between anomalous patterns in process data and their associated root causes as labels. By using a spatiotemporal pattern network (STPN), multivariate process time-series were defined as input vectors to the classification model.

Alarm data can be utilized as an alternative to process data to establish classification schemes for handling abnormal conditions. During the monitoring procedure, alarms are triggered whenever an abnormal situation or fault is identified in the process. On the other hand, process data are recorded by continuously measuring process variables during process operations. As a result, alarm data are normally of lower volume compared to process data and carry information concerning abnormal conditions. This makes alarm data an advantageous candidate for developing ML-based classification strategies aiming to address the problem of online alarm floods.

By using alarm data, a classification-based fault diagnosis method using a discrete hidden Markov model (HMM) was proposed in [7], which aimed to provide an online root cause analysis mechanism. An HMM is a probabilistic model that assumes the system being modeled is in one of a variety of states at any given moment. In this model, the states are subject to a Markov property and evolve by relying on the transition probabilities of each state. Given a set of observations, the estimation of the HMM parameters can be obtained by using the Baum-Welch algorithm that is based on the expectation maximization (EM) algorithm. With the aid of the Viterbi algorithm,

the trained HMM model can be utilized to uncover the hidden state sequence that explains a given sequence of observations. The authors in [7], assumed the possible process faults from historical data as the states of an HMM and alarm sequences associated with fault scenarios as observation sequences. By using historical fault scenarios and their associated alarm sequences, an HMM was trained for an online fault diagnosis and root cause analysis. By using the trained HMM, the fault scenario (most probable hidden state sequence) for a particular alarm sequence was identified. To find the most probable root causes, the first and second most probable hidden faults were determined by identifying the first and second most likely states in the most probable hidden state sequence. In [68], a classification-based approach using binary alarm signals was proposed for online analysis of alarm floods, where coactivations of alarm signals were used to measure similarity. A method for classifying online alarm floods was developed in [32], where alarm floods were modeled as numerical vectors without taking the time factor into account. Online root cause analysis can be facilitated by categorizing ongoing alarm floods into groups that have historically been encountered. As an important challenge in online alarm flood management, providing early online assistance for plant operators can significantly reduce the risk of process failures and hazardous incidents. As a result of incorporating the triggering time of alarms, a method for early classification of alarm floods was developed in [86], where a vectorization mechanism was proposed to represent alarm floods as feature vectors.

1.3.5 Other Data-Based Approaches

There has also been research on other concepts that can be used for alarm flood management and root cause analysis. Reference [95] devised a method to locate the root cause of oscillations in dynamic systems with control loops. The key concept of this approach was a nonlinearity score, which was illustrated to be strongest at the source of abnormalities. In [13] the interdependence of process variables was analyzed based on the nearest neighbors calculation

method, where the direction of the fault propagation path was investigated to address the ambiguous nonlinearity problem obtained in [95]. Correlation analysis is another method that has been employed to analyze the cause-and-effect relationships. The CCF was utilized in [14] to create a causal map by detecting the time delay between two consecutive process measurements during the disturbance propagation. It is worth noting that, models that rely on correlation analysis do not necessarily reveal the causality between variables and can lead to unreliable results.

Some research exploited contribution plots for root alarm discovery tasks. The contribution values of process variables (various scores can be defined for this purpose) were evaluated, and the one with the most significant contribution was obtained as the root alarm. Paper [120] formulated the mutual contributions of variables and defined a new score based on these contributions, which could be calculated in an iterative manner. Then, the process variable that showed the first abnormal fluctuation was identified as the root cause. Also, in [41], the authors used contribution plots to discover the root cause of alarms. They improved these plots by applying an iterative approach based on a filtered version of contribution values.

The authors in [99] introduced an approach to determine a set of root cause process variables according to their qualitative trends. These qualitative trends were captured from historical data based on amplitude changes, time duration and correlation coefficients of process variables. They defined primary process variables as those whose abnormalities had the highest concerns for safety and/or proficiency of the process. Root cause variables were determined to be those whose qualitative trends had the most significant effect on trends of primary process variables. A graph-based causal discovery model was proposed in [19] to avoid some of the limitations encountered in BN modeling. Although the proposed method provided a more compact system structure and could be more easily implemented compared with BN models, it was computationally expensive in solving large-scale problems with a high

number of process variables. Convergent cross mapping (CCM) is another method that was used for causality analysis in the literature, see e.g., [96]. The theory behind this technique states that causality can be detected if time series of the affected variable can be used to recover the state of the causal variable.

A&E logs have been increasingly employed by researchers in the area of alarm management to conduct data-driven alarm flood monitoring and root cause analysis. By defining an alarm time series using the information of historical alarm logs, [31] used statistical approaches to investigate the temporal dependencies between alarm events by considering the time interval between successive triggered alarms as a random variable. For root cause analysis in non-stationary faulty processes, paper [65] proposed a causality index based on dynamic time warping, which was a method commonly used for similarity analysis of temporal sequences. In [101], a fuzzy association rule mining approach was developed to find potential linguistic alarm association rules by analyzing consequential alarms. The authors in [63] established a framework for alarm analysis, where they first created batches of alarm sequences corresponding to system stoppages and the first alarms that appear simultaneously in the batches were identified as the root alarms. An alarm root cause analysis method based on evaluating the correlations and time delays between alarm data was developed in [21].

Alarm floods originating from the same abnormal situation are likely to contain common alarms triggered in a particular chronological order. Similarity analysis of historical alarm floods has been found to be an effective strategy for managing alarm flooding. Based on representing alarm floods as binary vectors and using consecutive alarm frequencies, an alarm flood similarity analysis using the Jaccard distance and dynamic time warping was devised in [2]. In recent years, sequence alignment methods have been effectively used for similarity analysis of alarm floods. Inspired by pattern matching methods, a modified Smith–Waterman (MSW) algorithm was proposed in [22] to exam-

ine the similarity of alarm sequences associated with two alarm floods. In [61], the MSW method was extended to consider pattern mining in multiple alarm floods. A method for local sequence alignment using the basic local alignment search tool (BLAST) was presented in [47], in which set-based pre-matching was incorporated to reduce computational complexity. In [38], a match-based accelerated alignment (MAA) strategy was developed aiming at improving robustness against nuisance alarms and computational efficiency. A modified PrefixSpan algorithm to detect similar sequential alarm patterns across different alarm floods was proposed in [75]. Hu *et al.* devised a frequent pattern mining approach to detect groups of alarms requiring suppression [45]. An alarm sequence mining mechanism based on a modified CloFAST algorithm was developed in [118], which aimed at extracting compact sequential alarm patterns and addressing the problem of alarm order switching.

According to the reviewed literature, existing research mostly deals with the problem of alarm floods and root cause analysis for offline applications. Aside from the attempts at developing online approaches discussed in the literature review, some pattern mining approaches have also been proposed to address online similarity analysis of alarm flood sequences (see [62, 77]). However, high computational complexity is a key feature of these approaches, which makes them inappropriate for solving online problems, such as early classification. Developing methods to provide useful online operator assistance for taking timely corrective actions is still of significant importance for process safety and efficiency. It is possible to develop machine learning-based approaches to fill the gaps existing in the field of online alarm flood monitoring and root cause analysis, which was the motivation for our research.

1.4 Thesis Contributions

Aiming at providing decision support mechanisms for on-site plant operators, this thesis proposes advanced machine learning-based methods for online alarm flood management and root cause analysis using alarm data. Following

is a summary of the major contributions that differentiate this thesis from related literature.

- First, Alarm floods are modeled as feature vectors based on exponentially attenuated component (EAC) analysis, which uses the temporal information of unlabeled historical alarm floods. A uniform strategy based on a Gaussian mixture model (GMM) that incorporates offline labeling and online classification is developed for early classification of ongoing alarm floods. During the offline phase, training data is automatically prepared based on partial labeling of historical alarm floods.

- Second, inspired by natural language processing (NLP), a novel vector representation called Modified Bag-of-Words (MBoW) is developed, which could capture the key features of alarm flood sequences including the chronological order of triggered alarms. Similarity analysis for alarm floods is addressed by grouping similar alarm flood vectors via an ML-based clustering method while using efficient similarity measurement. A weighting strategy that ranks alarms based on their relevance to specific abnormal situations is proposed to assist operators in alarm flood management. The proposed method can provide insight from historical data, and facilitate the handling of large datasets.

- Third, an online extension of the MBoW vectorization model and alarm ranking is employed to develop an online operator assistance mechanism. An ML-based open set classification strategy based on a systematic similarity threshold estimation technique is proposed to deal with previously unseen situations. The open set classifier can avoid incorrect classifications by excluding samples with low classification confidence via incorporating a reject option. The proposed classification method along with alarm ranking can help operators make timely decisions to handle both previously seen and new alarm flood scenarios.

- Finally, A novel alarm flood representation based on a probabilistic

framework is proposed, which is capable of tolerating alarm order ambiguities and the effect of irrelevant alarms. The proposed representation transforms the time stamped alarm sequences associated with alarm flood data into suitable inputs for a convolutional neural network (CNN). An online alarm flood analysis is then carried out through training a CNN, which predicts upcoming fault scenarios by observing online alarm floods. Due to a modified training process, the classification accuracy of online alarm floods with alarm order ambiguity is improved even at early stages of alarm flood occurrence.

1.5 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2 the modified EAC analysis for modeling alarm floods as feature vectors is presented and a Semi-Supervised Learning strategy based on the GMM is developed for early classification of industrial alarm floods. In Chapter 3 an MBoW alarm flood vector model is introduced along with an alarm ranking technique, and similarity analysis of alarm floods is conducted. Chapter 4 extends the proposed MBoW representation for online alarm floods and develops an online alarm flood classification mechanism to deal with previously unseen abnormal situations. In Chapter 5 a novel alarm flood representation based on a probabilistic framework is proposed aiming at tolerating alarm order ambiguities and the effect of irrelevant alarms. An analysis of online alarm floods is then conducted using a CNN, where upcoming abnormalities are predicted based on observations of online alarm floods. Finally, Chapter 6 concludes the thesis and suggests some potential directions for future research.

Chapter 2

Early Classification of Industrial Alarm Floods Based on Semisupervised Learning*

In this chapter, we propose a semi-supervised data-driven method for the problem of early classification of ongoing alarm floods with unlabeled historical data. EAC analysis is used to represent alarm floods as feature vectors. A method based on the unlabeled historical alarm floods is formulated to determine the attenuation coefficient for EAC representation. The proposed semi-supervised approach is formulated based on the GMM and includes two phases, namely, 1) offline clustering and labeling, and 2) online early classification. By applying an efficient cluster validity index, the proposed clustering method is automated in terms of choosing the optimal number of clusters. In the labeling step, it is not necessary to label all the individual historical alarm floods, making the proposed strategy much more practical. Moreover, a uniform strategy based on GMM is developed for an accurate early alarm flood classification. The effectiveness of the proposed approach is then shown by using the Tennessee Eastman process (TEP) benchmark and an industrial alarm flood dataset.

This chapter is organized as follows. In Section 2.1, a brief review of

*The material in this chapter has been published as: Haniyeh Seyed Alinezhad, Jun Shang, and Tongwen Chen, "Early classification of industrial alarm floods based on semisupervised learning," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1845–1853, 2022.

the concepts used in flood analysis is provided. Section 2.2 shows how the EAC representation is used to define alarm flood feature vectors and proposes a method for determining the attenuation coefficient. The developed semi-supervised alarm flood classification strategy is elaborated in Section 2.3 with discussion on both offline labeling and online early classification steps. In Section 2.4, two case studies are presented to verify the efficiency of the proposed algorithm. Section 2.5 concludes this chapter.

2.1 Historical Alarm Flood Data

During an industrial plant operation, the information of triggered alarms is recorded into a database called historical A&E log. To have an efficient alarm flood detection, it is necessary to preprocess the stored alarm data by removing chattering alarms. Preprocessing A&E log and historical alarm flood detection are performed offline, and the outcome is a list of alarm floods stored as time-stamped alarm sequences.

2.1.1 Removing Chattering Alarms

It is necessary to eliminate the alarm chatters from historical A&E log before detecting alarm floods. A chattering alarm is an alarm tag with repeated transitions between normal and abnormal states in a short period of time. Whereas chattering alarms include the useful information regarding the corresponding alarm tags, repeatedly triggered alarm tags increase the number of alarms within a short time interval, which could lead to the misidentification of alarm floods. Therefore, removing the chattering alarms in addition to preserving the information of their alarm tags is important for an effective alarm flood detection and analysis. Chattering alarms can be detected by using a time window to count the number of a certain alarm tag activations within a short time period without considering other associated alarm tags. For chattering alarm removal, different alarm suppression techniques, such as delay timers, deadbands, and filters can be applied [93, 100, 105]. For instance,

a method for suppressing the alarm chatters is to consider only one alarm instead of chattering alarms by merging the repeating alarm tags within the corresponding time interval [2].

2.1.2 Detecting Alarm Floods

After suppressing the chattering alarms, historical alarm floods can be extracted from A&E log for further analysis. General conditions can be defined for identifying the start and the end of an alarm flood sequence. To this end, arising at least σ alarm messages within time interval $\mathcal{T}$ is considered as the triggering condition of an alarm flood. For example, in ANSI/ISA-18.2, alarm rate in an alarm flood is assumed to be at least 10 alarms per 10 minutes: $\sigma = 10$ and $\mathcal{T} = 10$ [50]. Similarly, triggering less than $\kappa\sigma$ alarms, where $\kappa \in (0, 1]$, shows the end of the alarm flood. Based on these conditions, alarm floods can be detected by using a $\mathcal{T}$-width sliding time window, which counts the number of historical alarm messages within time $\mathcal{T}$. Then an alarm flood indicator η_t can be defined as

$$\eta_t = \begin{cases} 1, & \text{if } \eta_{t-\delta} = 0, \pounds(t - \mathcal{T}, t) > \sigma \\ 1, & \text{if } \eta_{t-\delta} = 1, \pounds(t - \mathcal{T}, t) > \kappa\sigma \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

where δ is the step size of the sliding window, and $\pounds(t_1, t_2)$ denotes the number of activated alarms within the time interval $[t_1, t_2]$ [86]. While the alarm flood indicator is equal to one, detected alarm sequence from A&E log is identified as an alarm flood. Finally, detected alarm floods are recorded as sequences of time-stamped alarm tags in a historical alarm flood database.

2.2 Alarm Flood Vector Representation

Similar alarm floods can be assumed to be caused by the same abnormal situation. Thus, classifying an online alarm flood to the group of similar historical alarm floods is an idea for providing online assistance for plant operators. Inspired from pattern mining methods, several efficient approaches,

such as MSW, have been proposed to address the similarity analysis and classification of flood sequences. However, high computational complexity is an important feature of pattern mining approaches, which makes them inappropriate for solving online problems like early classification. To fill this gap, the classification problem of alarm floods can be studied based on machine learning, which includes powerful tools for clustering and classification.

Representing alarm floods in the form of feature vectors makes alarm sequences fit for machine learning applications. In general, sequences of alarm messages in alarm floods can be converted into binary vectors such that each entry indicates the state of an alarm with values "1" and "0" serving as the abnormal and normal states, respectively. Each binary alarm flood vector is denoted as $\boldsymbol{F} \in \mathbb{R}^n$, where n is the total number of unique alarm tags in the monitoring system. Nonzero entries of $\boldsymbol{F}$ correspond to the annunciated alarms in the flood sequence and the total number of annunciations in $\boldsymbol{F}$ is equal to the 0-norm of $\boldsymbol{F}$, $\|\boldsymbol{F}\|_0$. This way, an alarm flood database, including m flood sequences, can be defined as a set of binary vectors, namely, $\mathfrak{F} = \{\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_m\}$. As the original alarm floods are in the form of time-stamped alarm sequences, it is not suitable to use $\boldsymbol{F}$ directly as feature vector because it ignores the time information of flood sequences. Moreover, owing to the importance of earlier triggered alarms in providing the early classification of alarm floods, it is not efficient to use equal weights for alarms in the flood vector. Therefore, establishing a vector representation, which dedicates different weights to alarms while preserving the chronological order of alarms, is necessary.

2.2.1 EAC Feature Vector

In addition to representing alarm states of the jth flood sequence, for $j \in \{1, 2, \ldots, m\}$, in the form of the binary vector $\boldsymbol{F}_j$, the triggering time of each alarm can also be recorded as a relative time vector denoted as $\tau_j \in \mathbb{R}^n$. For defining this vector, the time instant of the first annunciated alarm in

the jth flood is considered as a baseline, and the time durations between the later annunciated alarms and this baseline are calculated. The resulting values serve as the nonzero entries of τ_j, which are recorded in the locations same as those of their corresponding alarms in $\digamma_j$. The entries of τ_j corresponding to the alarms that are not annunciated during the alarm flood are then set to zero. Finally, by applying these steps to all of the historical alarm floods, a historical time dataset including all relative time vectors can be constructed, denoted as $\mathfrak{T} = \{\tau_1, \tau_2, \ldots, \tau_m\}$.

In early classification, the information of alarms activated in the early stages of a flood is available, and efficiently utilizing this limited number of alarms for achieving accurate classification is important. Following the effect of fault propagation in many industrial processes, the triggered alarms during an alarm flood are likely to be causally interrelated rather than following a coincidental relationship. This usually makes the alarms activated at the earlier steps more relevant to the root causes of abnormalities that lead to alarm floods. An EAC vector representation was proposed to take the important roles of the earlier annunciated alarms into account for enhancing an accurate early classification [86]. Relative triggering time information in $\mathfrak{T}$ is embedded into this representation as exponentially attenuated weights, which are dedicated to alarm tags with respect to elapsed times. Accordingly, the EAC feature vector is defined as follows:

$$\varkappa = \digamma \circ \exp(-\lambda\tau) \tag{2.2}$$

where λ is the attenuation coefficient, $\circ$ denotes the Hadamard product, which is an elementwise product between two matrices, and $\exp(\cdot)$ is an elementwise exponential function. In contrast to the binary flood vector $\digamma$, vector representation $\varkappa$ not only includes the information about the chronological order of the alarms in the flood sequence but also takes advantage of the earlier activated alarms to achieve an accurate early classification. Note that the attenuation coefficient determines the attenuation degree of the weights dedicated to alarms. Thus, choosing an appropriate λ is crucial to fully reflect the

time information in $\mathfrak{T}$.

2.2.2 Determining Attenuation Coefficient

By assuming the attenuation coefficient λ as a description for the characteristics of the system, it is considered to be constant for a specific industrial process, and a reasonable value for λ can be determined by using the historical alarm flood data. However, under this assumption, unbalanced weights may be dedicated to the alarm floods with different durations. Therefore, providing an efficient way to determine λ is necessary for the purpose of accurate early classification. Shang and Chen [86] proposed a supervised method to learn λ from labeled historical flood data. In practical cases, it is difficult and time-consuming to obtain enough labeled data for an effective offline training. To address this problem, an efficient method without needing *a priori* knowledge about the labels of alarm floods is proposed in this section.

According to (2.2), the weights for the jth flood are defined in the form of an exponentially decaying quantity as

$$q(t(i_j)) = \exp(-\lambda t(i_j)) \tag{2.3}$$

where $t(i_j)$ denotes the relative triggering time corresponding to the alarm tag $i_j \in \mathfrak{A}_j$ such that $\mathfrak{A}_j = \{a_j^{\text{start}}, \ldots, a_j^{\text{end}}\}$ denotes the set of the unique alarm tags triggered in the jth alarm flood sequence. The nonzero elements of flood vector $\digamma_j$ correspond to the alarm tags in the set $\mathfrak{A}_j$ and the vector τ_j includes the relative triggering time instants $t(i_j)$.

For $j = \{1, 2, \ldots, m\}$, $q(t(i_j))$ creates a set of exponentially attenuated elements depending on the information of τ_j and the value of λ. The attenuation coefficient λ is a positive constant and determines how rapid the attenuated elements vanish, such that a larger λ leads to a more rapid vanishing of the decaying quantity (2.3). The behavior of the exponentially attenuated quantity in (2.3) with different values of λ is shown in Figure 2.1. According to this property, a too large λ may ignore many of later triggered alarms in long alarm floods and a too small λ could not properly reflect the importance of

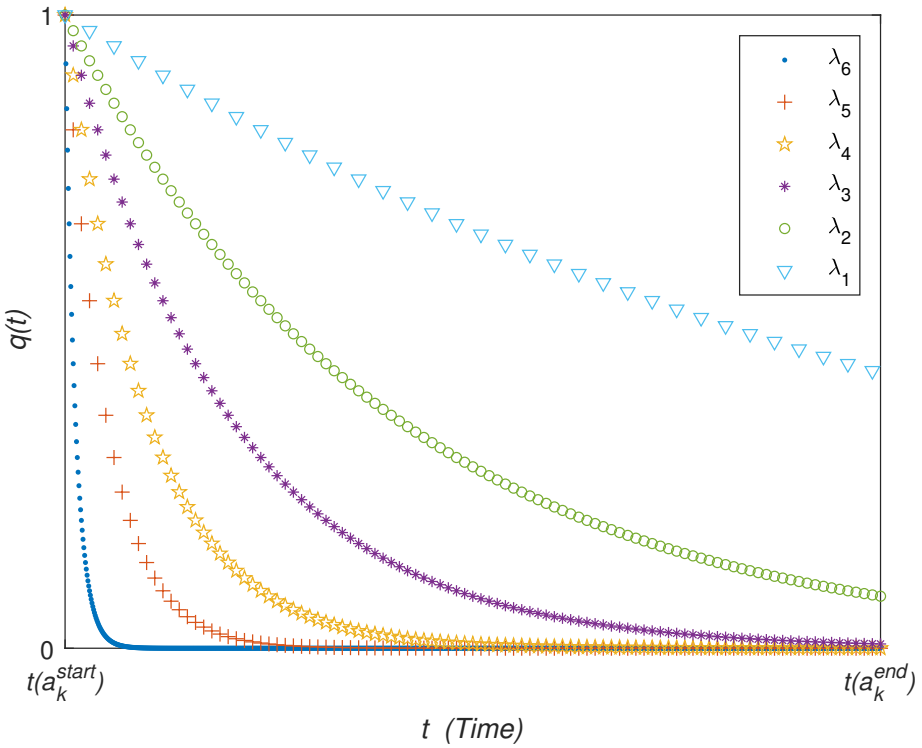

Figure 2.1: Exponentially attenuated weights of the jth alarm flood with respect to elapsed time for different values of λ, where $\lambda_6 > \lambda_5 > \cdots > \lambda_1$.

earlier triggered alarms in short flood sequences. Note that in Figure 2.1, the function in (2.3) is depicted in the time interval $[0, t(a_j^{\text{end}})]$ including the time instants of τ_j.

Since the lengths of historical alarm flood sequences can vary remarkably, a careful selection of λ is necessary to preserve the important information of all floods in the EAC flood representation. Following the stated properties, an efficient method can be provided to determine a proper value for λ based on the information of the maximum triggering time of historical alarm floods in $\mathfrak{T}$. An important prerequisite for an efficient selection of λ is to keep the units of all historical time vectors in $\mathfrak{T}$ consistent. The dataset for the purpose of offline learning of λ is defined by extracting the maximum triggering time instants as $\mathfrak{T}_e = \{t(a_1^{\text{end}}), \ldots, t(a_m^{\text{end}})\}$. Then a reasonable way to determine λ is to consider the median value of the set $\mathfrak{T}_e$, as the mean lifetime of decaying

quantity (2.3), which is denoted as $\mathfrak{M}$. Mean lifetime can be defined as $\mathfrak{M} = 1/\lambda$, and it is the time at which the initial value of (2.3) is reduced to $1/e$, where e is Euler's number [57]. This way, the alarms with the triggering times belonging to the interval $[0, \mathfrak{M}]$ are allocated weights with values from the interval $[0, 1/e]$. Then, alarms triggered at time instants greater than $\mathfrak{M}$ obtain attenuated weights less than $1/e$. Choosing a uniform attenuation coefficient for a set of historical alarm floods with different durations contains a trade-off between dedicating balanced weights to all floods and preserving the information of all triggered alarms. By choosing the value of attenuation coefficient in terms of the concept of mean lifetime, i.e., $\lambda = 1/\mathfrak{M}$, reasonable weights are dedicated to all triggered alarms in historical alarm flood data. Therefore, important features of floods with different durations are reflected in EAC vectors while the importance of earlier triggered alarms is preserved. In addition, the proposed method does not need labeled alarm flood data and the training dataset just contains the relative triggering time information of the historical alarm flood sequences.

2.3 Semi-Supervised Early Classification

For classifying an online alarm flood in the machine learning framework, a dataset including historical alarm floods with their class labels corresponding to abnormal situations is needed to train a classifier. Providing such a dataset is a complex and time-consuming task, which may impose unnecessary costs. In this section, a semi-supervised approach based on GMM is formulated to address the problem of early classification using unlabeled historical data. The proposed method consists of two phases: A) An offline phase for data preprocessing and providing the labeled training dataset; B) An online phase for early classification of ongoing alarm floods.

2.3.1 GMM-Based Offline Alarm Flood Labeling

Data clustering is an unsupervised learning strategy used for grouping similar data in an unlabeled historical dataset. This machine learning method can be used for data labeling by considering the data of each cluster from the same category. Exploring the group-structures in the context of GMM is a probabilistic cluster analysis, which is renowned for its efficiency and flexibility. In this framework, data observations are considered to be realizations of a random variable with a PDF in the form of a finite GMM.

GMM-Based Clustering

Consider the set of historical alarm flood data with m observations. $\chi = \{\varkappa_1, \ldots, \varkappa_m\} \in \mathbb{R}^n$ are realizations of a random vector $\mathbf{X}$. In GMM, it is assumed that the observed flood data are made up of a mixture of c Gaussian distributions in proportions $\rho_1, \ldots, \rho_c$ as follows:

$$g(\varkappa; \Phi) = \sum_{i=1}^{c} \rho_i g_i(\varkappa; \varphi_i) \tag{2.4}$$

where $g(\varkappa; \Phi)$ is the PDF of the mixture model for random variable $\mathbf{X}$. Mixing weights ρ_i are nonnegative such that $\sum_{i=1}^{c} \rho_i = 1$, and $g_i(\varkappa; \varphi_i)$ is the PDF of the ith Gaussian component. $\varphi_i = (\mu_i, \Sigma_i)$ denotes the vector of unknown parameters including covariance matrix and mean of the ith Gaussian distribution, denoted as Σ_i and μ_i, respectively. Also, $\Phi = [\theta^T, \rho_1, \ldots, \rho_c]^T$ denotes the vector of mixture model parameters such that θ^T includes the elements of all φ_i's.

Given data observations in χ, mixture model can be fitted via the maximum likelihood approach, which aims at estimating mixture model parameter vector Φ, by maximizing the log-likelihood of (2.4) [71]. However, in the context of data clustering, there is some missing information including data labels, denoted as $z_1, z_2, \ldots, z_m$, and the number of clusters, which makes χ incomplete for the maximum likelihood estimator.

In GMM clustering, each Gaussian component is considered as a cluster

and the number of clusters in mixture model is denoted as c. Assuming that the number of components is known, clusters can be assigned labels defined as $\mathfrak{C} = \{1, 2, ..., c\}$. Then, the data labels $\mathcal{Z} = \{z_1, \ldots, z_m\}$ can be considered as observations of a categorical random variable $\mathfrak{Z}$ taking its values from the set $\mathfrak{C}$ with probabilities $\rho = \{\rho_1, \ldots, \rho_c\}$. The random categorical variable $\mathfrak{Z}$ can be defined in the form of a c-dimensional random vector Z such that its observation corresponding to the label of data point $\varkappa_j$ is defined as $z_j = [z_j^i]_{c\times 1}$, where

$$z_j^i = \begin{cases} 1, \text{ if } x_j \text{ has arisen from the } i\text{th component} \\ 0, \text{ otherwise} \end{cases} \tag{2.5}$$

Then, the possible results of the random variable Z are defined through a multinomial distribution $z_j = \text{Mult}_c(1, \rho)$ including one trial over c categories belonging to the set $\mathfrak{C}$ with probabilities $\rho_1, \ldots, \rho_c$. Based on this definition, the complete-data vector for maximum likelihood estimation in the context of GMM clustering is defined as $\varkappa_j^{\text{complete}} = (\varkappa_j^T, z_j^T)^T$.

For the purpose of clustering using GMM, the posterior probability that $\varkappa_j$ arises from the ith mixture component indicates whether the jth observation belongs to the ith group or not. From Bayes' theorem it can be verified that the posterior probability is given as follows [71]:

$$\mathcal{P}_i(\varkappa_j; \hat{\Phi}) = \Pr\{Z^i = 1 | X = \varkappa_j\} = \frac{\hat{\rho}_i g_i(\varkappa_j; \hat{\varphi}_i)}{g(\varkappa_j; \hat{\Phi})} \tag{2.6}$$

where Z^i is the random variable corresponding to z_i^j. The estimates of parameter vectors $\hat{\Phi}$ and $\hat{\varphi}$ are obtained via the expectation–maximization (EM) algorithm. This way, a probabilistic clustering is provided to group data observations χ into c groups. This clustering algorithm is based on fitted membership posterior probabilities such that data points are assigned to components having highest $\mathcal{P}_i$.

The EM algorithm is an iterative scheme including two steps, used for estimating mixture parameters [71]. This algorithm uses the following complete

log-likelihood function for the purpose of inferring the mixture model:

$$\mathfrak{L}_c(\Phi) = \sum_{i=1}^{c} \sum_{j=1}^{m} z_j^i \{\log \rho_i + \log g_i(\varkappa_j;\, \varphi_i)\} \tag{2.7}$$

For obtaining $\hat{\Phi}$ in the EM algorithm, the conditional expectation $\Psi(\Phi; \Phi^*) = E\{\mathfrak{L}_c(\Phi)|\chi, \Phi^*\}$ given the observed data and current fit of Φ, Φ^*, is iteratively maximized. To this end, two steps, namely the E-step and M-step, are alternated iteratively until a specific condition is satisfied.The criterion for stopping the EM iterations is defined as $|\mathfrak{L}_c(\Phi^{(k+1)}) - \mathfrak{L}_c(\Phi^{(k)})| < \varepsilon$, where ε is a small positive value.

Silhouette Validity Index

Note that the number of Gaussian components needs to be specified before running the clustering algorithm. In real applications, there is mostly no prior information about the exact number of alarm flood categories. Therefore, providing a measurement to assess the quality of clustering results for different numbers of clusters is needed for determining the optimal cluster number. The clustering quality can be measured via an internal cluster validity index, called Silhouette index, in terms of the concepts of cohesion and separation [6].

In a clustering partition, for a predefined number of clusters $\mathcal{K}$, consider that the flood observation $\varkappa_j$ is assigned to the cluster C such that $C \in \{1, \ldots, \mathcal{K}\}$. Then, the Silhouette coefficient for the jth historical flood data is defined as

$$\text{Silhouette}_{\varkappa_j} = \frac{m_j - a_j^C}{\max\{m_j, a_j^C\}} \tag{2.8}$$

where a_j^C denotes the average distance of $\varkappa_j$ to all other flood data in the cluster C. The indicator m_j is defined as $m_j = \min_{\bar{C} \neq C}\{d_j^{\bar{C}}\}$ wherein $d_j^{\bar{C}}$ represents the average distance of the $\varkappa_j$ to all the data points belonging to all other clusters denoted as $\bar{C}$. The proposed index ranges from 0 to 1; a higher value shows a better assignment of data point $\varkappa_j$ to the cluster C. Similarly, the average Silhouette index of all historical flood data can be used as a validity measurement for the quality of the overall clustering result. But this needs

the computation of all distances among all m data points, which imposes the computational cost as $\mathcal{O}(nm^2)$. To reduce the computational burden, a simplified Silhouette index can be provided by redefining the index (2.8) based on the distance of the data points to cluster centers, denoted as $s_{\varkappa_j}$. Then, the following average index is used to assess the quality of the clustering result:

$$\mathcal{S} = \frac{1}{m} \sum_{j=1}^{m} s_{\varkappa_j} \tag{2.9}$$

which reduces the computational cost to $\mathcal{O}(cmn)$.

In the offline phase, each component of the resulting GMM can be labeled by utilizing the information of the abnormal situation corresponding to the alarm flood vector with the highest posterior probability in the component. Plant experts can use the original sequence representations of alarm floods to determine their class labels by using their knowledge about the plant. To this end, some annotations are provided for each alarm flood to be used for labeling its corresponding mixture component. It should be noted that data labeling based on expert knowledge could be considered as a limitation of the proposed method; but this partial labeling would be more applicable and efficient than labeling all alarm floods in the training dataset. Moreover, by using this training dataset, the online phase of the proposed method may not directly give root causes of ongoing alarm floods. However, by early classification, some previously diagnosed and annotated information is provided for on-site operators, which can help them know the root causes and then take a timely action. The overall procedures for providing the labeled GMM is summarized in Algorithm 1.

2.3.2 Online Alarm Flood Classification

In the online phase, an ongoing alarm flood is assumed to come from one of the c components in the training dataset, and it is assumed that each of the components relates to one unique root cause. Then, the label of the corresponding mixture component is considered as the class label of the observed

Algorithm 1: Offline alarm flood labeling

Input: $\{\mathfrak{F}, \mathfrak{T}\}$

Output: $g(\varkappa; \hat{\Phi})$ with labeled components

begin

Create $\mathfrak{T}_e$;

Compute median value of $\mathfrak{T}_e$ and save as $\mathfrak{M}$;

Set $\lambda = 1/\mathfrak{M}$;

Create EAC flood dataset χ by (2.2);

for $C \in \{1, \ldots, \mathcal{K}\}$ **do**

Initialization: Select $\Phi^{(0)}$;

Set k = 0;

while $|\mathfrak{L}_c(\Phi^{(k+1)}) - \mathfrak{L}_c(\Phi^{(k)})| > \varepsilon$ **do**

E step:

Compute $\Psi(\Phi; \Phi^{(k)})$;

M step:

Find $\Phi^{(k+1)}$ by solving $\Phi^{(k+1)} = \arg\max_{\Phi} \Psi(\Phi; \Phi^{(k)})$;

$k \longleftarrow k + 1$;

$\hat{\Phi}_C \longleftarrow \Phi^{(k+1)}$;

Compute $\mathcal{S}_C$ for GMM $g_C(\varkappa; \hat{\Phi}_C)$ by (2.9);

$c = \arg\max_{C} \mathcal{S}_C$;

$g(\varkappa; \hat{\Phi}) \longleftarrow g_c(\varkappa; \hat{\Phi}_c)$;

for $i = 1$ **to** c **do**

Solve $\varkappa_{label_i} = \max_{\varkappa \atop S.t., \varkappa \in \chi} \mathcal{P}_i(\varkappa; \hat{\Phi})$;

Convert $\varkappa_{label_i}$ to its original sequence form;

Determine the category of the resulting sequence;

Label the ith component by determined category;

alarm flood. This is achieved via classification based on the estimated posterior probabilities defined in (2.6). For classifying an ongoing alarm flood in early stages, an immediate classification needs to be generated when an alarm flood is detected. To this end, once an online flood sequence is detected based on (2.1), it needs to be converted to the form of EAC vectors as

$$\hat{\varkappa}_{\text{online}} = \hat{F}^{\frown} \circ \exp(-\lambda \hat{\tau}) \tag{2.10}$$

Owing to smaller numbers of activated alarms, $\hat{F}^{\frown}$ and $\hat{\tau}$ include less information than those of full flood sequences, namely $F^{\frown}$ and τ; consequently, $\|\hat{\varkappa}_{\text{online}}\|_0 \leq \|\varkappa\|_0$. Then, $\hat{\varkappa}_{\text{online}}$ is assigned to the mixture component with the highest estimated posterior probability such that

$$\hat{z}^i_{\text{online}} = \begin{cases} 1, & \text{if } i = \arg\max\limits_{\alpha \quad S.t., \alpha \in \mathfrak{C}} \mathcal{P}_\alpha(\hat{\varkappa}_{\text{online}}; \hat{\Phi}) \\ 0, & \text{otherwise} \end{cases} \tag{2.11}$$

This way, the estimated class label is defined as $\hat{z}_{\text{online}} = i$. In the cases that the current alarm flood is caused by a new fault, there is no matching alarm flood in the historical data. These types of alarm floods are referred to as new classes and can be used later for updating the trained GMM. A threshold, denoted as γ, is introduced for the highest estimated posterior probability in (2.11) to prevent the hazardous consequence of misclassifying a new unseen alarm flood. This threshold is defined as a cutoff parameter to determine if $\hat{\varkappa}_{online}$ belongs to a new class or to a class previously seen in the training dataset. To this end, an ongoing alarm flood is assigned to the mixture component with the highest estimated posterior probability greater than or equal to γ; otherwise it is identified as an alarm flood caused by a new fault. By triggering new alarms in the ongoing flood sequence, flood vector (2.10) is updated and subsequently corresponding generated classification is updated until the end of the alarm flood. These updates lead to higher classification accuracies due to using more alarm information. However, using the EAC representation for alarm floods could lead to an acceptable classification accuracy even in the early stage of an alarm flood [86]. The summary of the proposed online classification is presented in Algorithm 2.

Algorithm 2: Online alarm flood classification

Input: Online alarm sequence, training data $g(\varkappa; \hat{\Phi})$;
Output: Class label of the ongoing alarm flood
begin
  while $\eta_t = 1$ **do**
    Convert the detected flood sequence to EAC vector representation $\hat{\varkappa}_{\text{online}}$ by (2.10);
    Solve $i = \arg\max_{\substack{\alpha \\ S.t., \alpha \in \mathfrak{C}}} \mathcal{P}_\alpha(\hat{\varkappa}_{\text{online}}; \hat{\Phi})$
    if $\mathcal{P}_i(\hat{\varkappa}_{\text{online}}; \hat{\Phi}) \leq \gamma$ **then**
      $\hat{\varkappa}_{\text{online}}$ is caused by a new fault;
    else
      Show the label of the ith component as class label of the $\hat{\varkappa}_{\text{online}}$;
    Update $\hat{\varkappa}_{\text{online}}$;

As mentioned, an important feature of the alarm sequence alignment approaches, among which the MSW method [22] is chosen as a benchmark for alarm flood analysis, is the high computational complexity. The proposed approach addresses this problem aiming at achieving an acceptable computational complexity in alarm flood classification. Given one flood pair the computational complexity for calculation of similarity score in MSW algorithm is $\mathcal{O}(MNn)$, where M and N are the numbers of alarms that are included in two floods. For classifying a new alarm flood by using the historical data including m floods, the total computational burden of the similarity measurements is $\mathcal{O}(MNn(m+1)^2)$. As the MSW algorithm needs to utilize full sequences of alarm floods, comparison is carried out on complete alarm flood sequences. Therefore, the proposed classification method is applied to the full episode of the alarm flood represented as $\varkappa \in \mathbb{R}^n$. In the proposed algorithm, the computational complexity for classifying an ongoing alarm flood in any of its updating steps is the same. This is due to the vector representation of alarm floods, which leads to the same dimension of the flood vector at any step. The inverse and determinant of the estimated covariance matrices can be calculated once before the classification step. Then, the total complexity

Table 2.1: Silhouette scores of GMMs with different numbers of components

Number of clusters	$C = 2$	$C = 3$	$C = 4$	$C = 5$	$C = 6$	$\mathbf{C = 7}$	$C = 8$	$C = 9$	$C = 10$
$\mathcal{S}_C$	0.4613	0.4728	0.5226	0.6052	0.6417	**0.6632**	0.6524	0.6505	0.6504

of computing the posterior probabilities for classifying the alarm flood $\varkappa$ is $\mathcal{O}(cn)$, which is much lower than that of the MSW algorithm.

2.4 Case Studies

2.4.1 TEP Dataset

The TEP benchmark with closed-loop plant simulator, developed in [11], is utilized to evaluate the performance of the developed approach. For simulating the alarm system, 39 process variables are considered for alarm configuration with four different states, namely "PVH," "PVL," "PVHH," and "PVLL". The simulation time is set as 10h and process measurements are uniformly sampled with a 0.6 min sampling interval. Seven faults, leading to different abnormal conditions, are considered as alarm flood categories [86]. For generating the alarm floods, the standard definition in (2.1) is applied by setting the parameters as $\mathcal{T} = 10$ min, $\sigma = 8$, and $\kappa = 0.75$. The flood dataset includes 280 alarm flood sequences with 40 floods in each category. The data are further split into training and test datasets, which contain 80% and 20% of total data samples.

According to the value $\mathfrak{M} = 25.5$ obtained from the extracted set $\mathfrak{T}_e$ for training data, the attenuation coefficient is determined as $\lambda = 0.039$. Algorithm 1 is implemented by setting $\mathcal{K} = 10$, and the Silhouette scores for resulting GMMs with different numbers of components are recorded in Table 2.1. It can be seen that the maximum value of the Silhouette validity index corresponds to the GMM with 7 components, which is equal to the number of alarm flood categories in the historical dataset. This shows the efficiency of the proposed algorithm in automatically determining the optimal number of clusters.

Table 2.2: External validity indices for different methods

Method	Proposed method	$\lambda = 0$	Method in [32]
Purity	**99.52%**	71.42%	75.71%
NMI	**0.9892**	0.8285	0.8557

Apart from the developed clustering method, we also implement the method in [32] and Algorithm 1 without attenuation for comparison. To show the efficiency of the alarm flood clustering and proposed EAC flood representation, true class labels of data are utilized to provide a clearer cluster evaluation in terms of two external cluster validity measures, namely, Purity and normalized mutual information (NMI) [3]. These two external cluster validity indices can be calculated by

$$\text{Purity}(\mathfrak{C}, L) = \frac{1}{m} \sum_{i \in \mathfrak{C}} \max_{l \in L} |i \cap l| \tag{2.12}$$

$$\text{NMI}(\mathfrak{C}, L) = \frac{2 \sum_{i \in \mathfrak{C}} \sum_{l \in L} \Pr(i \cap l) \log \frac{\Pr(i \cap l)}{\Pr(i) \Pr(l)}}{H(\mathfrak{C}) + H(L)} \tag{2.13}$$

where $\mathfrak{C}$ and L are set of evaluated clusters and set of true classes, respectively. $\Pr(i)$, $\Pr(l)$, and $\Pr(i \cap l)$ are the probabilities of an alarm flood being in cluster i, class l, and intersection of i and l, respectively. $H(\cdot)$ denotes the entropy. The measured validity indices are recorded in Table 5.2, which confirm the effectiveness of the proposed method in achieving an accurate data grouping. It can be seen that the clustering method developed in this study outperforms that of [32].

The efficiency of the proposed method for determining the attenuation coefficient is also investigated by fixing the value of C in Algorithm 1 and using different values for λ. The clustering results with different values of λ are shown in Figure 2.2, evaluated by Purity and NMI.

Algorithm 2 is then implemented to provide early classification of the test flood dataset. In Figure 2.3, the average classification accuracies in different time intervals are demonstrated with respect to the elapsed time. Online

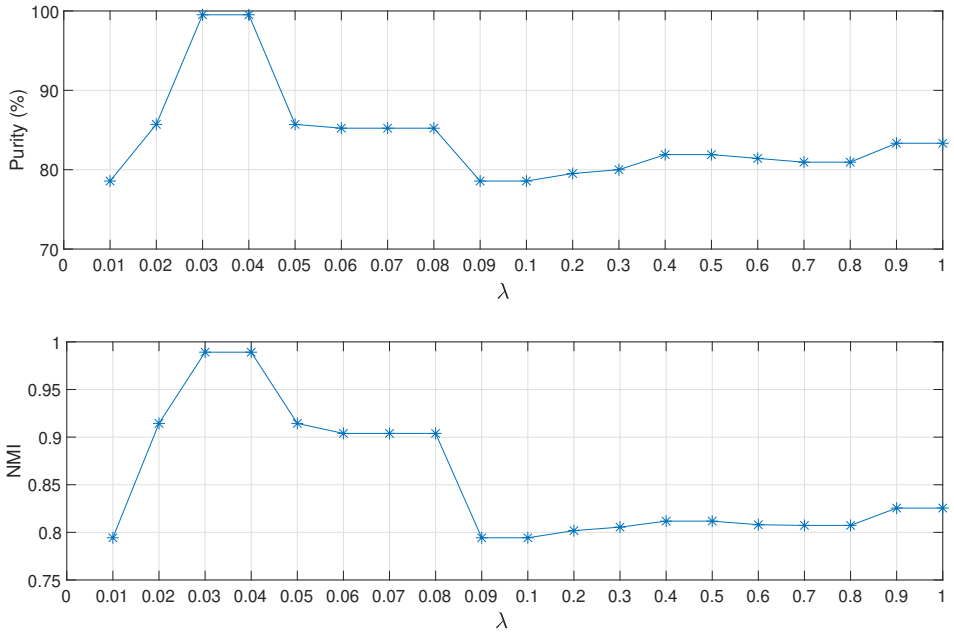

Figure 2.2: Clustering performance for different values of λ.

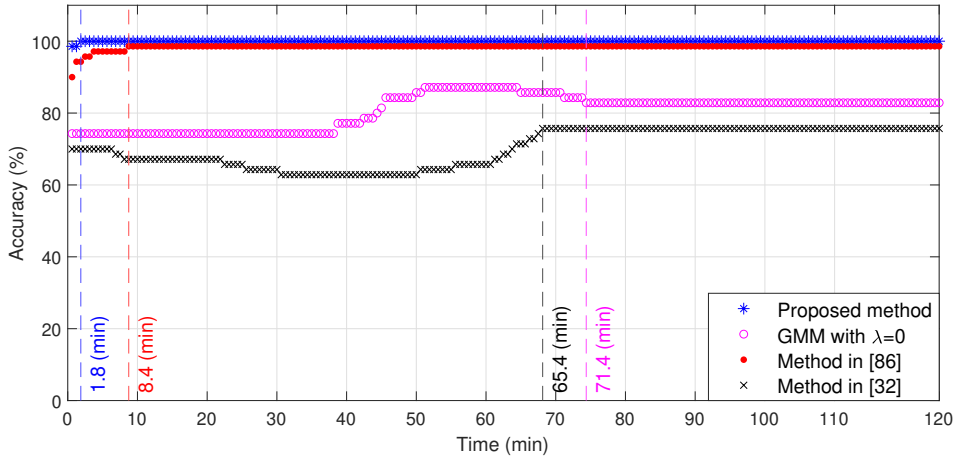

Figure 2.3: Average classification accuracy for TEP data.

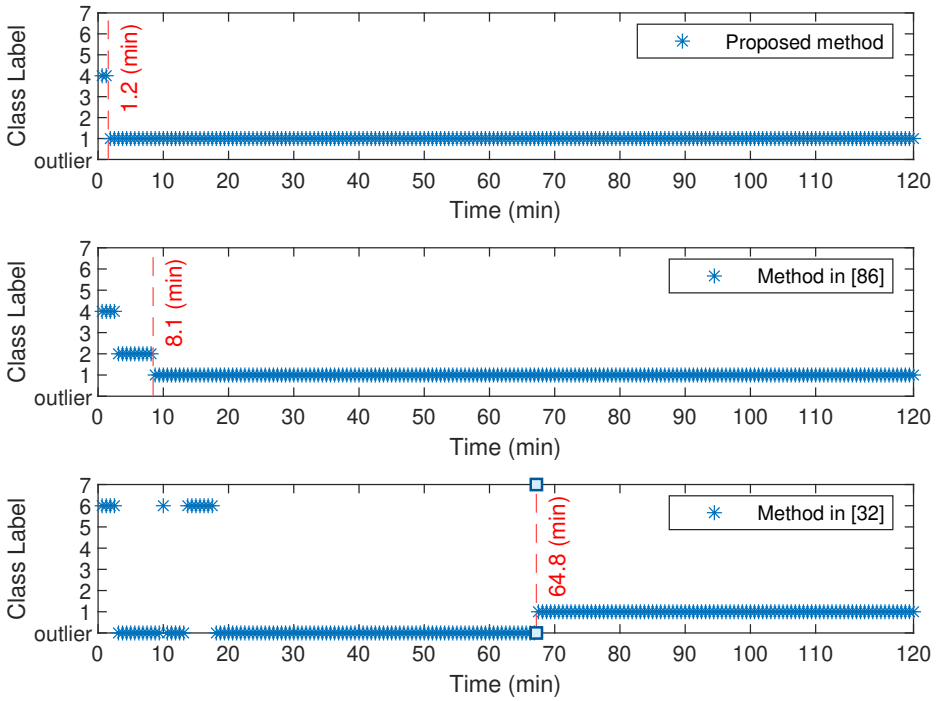

Figure 2.4: Classifications for an alarm flood from the first category in TEP data.

classification results for a single alarm flood in different time intervals are shown in Figure 2.4. Vertical dashed lines in Figs. 2.3 and 2.4 show the amount of time that each method takes to reach a classification accuracy, which will not change in the next updates of the online classification. It can be seen that the proposed method outperforms the others on different aspects such as early classification, maximum classification accuracy, and the number of labeled data required for classification.

The average running times of different methods for classifying an alarm flood in test dataset are recorded in Table 5.3. The methods are implemented in MATLAB 2020 on a 64-bit Windows PC with Intel Core i7-8700 CPU @ 3.20GHz and 8.0 GB RAM. These results show that the proposed algorithm is also more efficient in terms of running time. This is because the proposed classification method uses only the means of the c Gaussian components for

Table 2.3: Running times of different methods

Method	Proposed method	$\lambda = 0$	Method in [86]	Method in [32]
Running time (ms)	**0.0129**	0.0155	0.118	0.141

similarity analysis, but the other two approaches utilize the 1-NN algorithm that uses all m historical data points for classification.

2.4.2 Industrial Dataset

The alarm data corresponding to one-year operation of an industrial process is utilized to demonstrate the effectiveness of the proposed algorithm in real alarm flood data analysis. The recorded A&E log consists of the information of triggered alarms in the monitoring system from March 3, 2019 to May 2, 2020. An off-delay timer is applied to reduce chattering alarms in the A&E log. The alarm flood dataset is generated based on definition (2.1) by using industrial standard ANSI/ISA 18.2 [50]. The resulting alarm flood dataset includes 6 groups of similar alarm floods with maximum and average durations as 36.4 min and 12.19 min, respectively. Like the TEP dataset, alarm flood data are split into training dataset and test dataset to be used in offline training and online classification phases.

By implementing Algorithm 1 for training dataset, the value of attenuation coefficient is determined as $\lambda = 0.0017$, and the optimal number of components is achieved as $c = 6$. By using the true labels of the alarm floods, the clustering result is shown in Table 2.4 in terms of the confusion matrix and two validity indices. These results confirm the efficiency of the proposed algorithm in partitioning similar alarm floods. Finally, Algorithm 2 is implemented for test data, and the early classification performance is validated by depicting the evolution of the online classification accuracy with respect to elapsed time in Figure 2.5.

Table 2.4: Clustering results of the industrial dataset

		Predicted category					
		1	2	3	4	5	6
Actual category	1	9	-	-	-	-	-
	2	-	12	-	-	-	-
	3	-	-	5	-	-	-
	4	-	-	-	4	-	1
	5	-	-	-	-	5	-
	6	-	-	-	-	-	9
Purity=97.77%				NMI=0.9628			

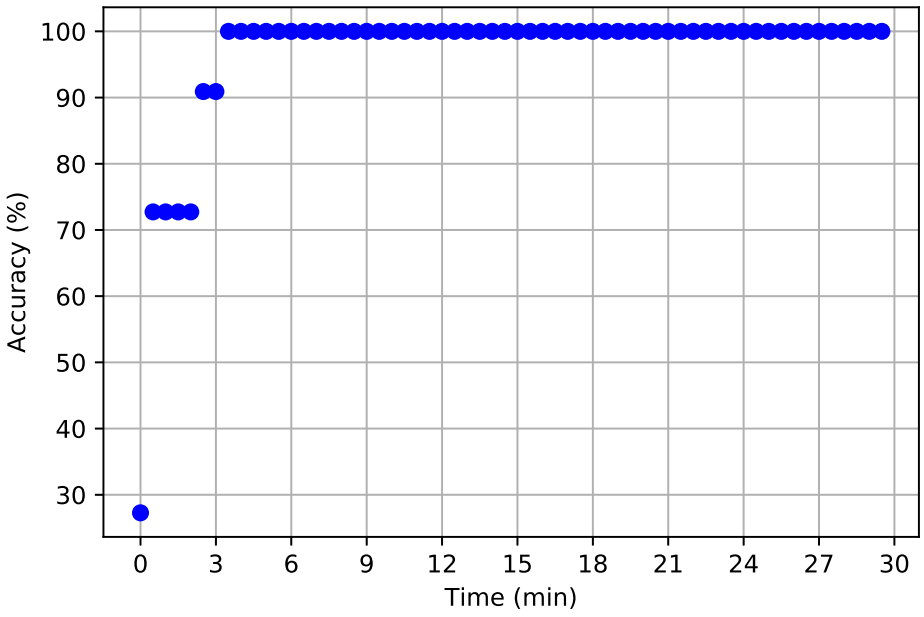

Figure 2.5: Average classification accuracy for industrial data.

2.5 Conclusion

In this chapter, a semi-supervised data-driven approach based on GMM was proposed to address the problem of early classification of ongoing alarm floods with unlabeled historical data. A vector representation called EAC was utilized to convert alarm flood sequences to feature vectors, which can reduce the computational complexity encountered in online pattern mining approaches. EAC feature vectors could also lead to an accurate early classification of alarm floods by considering the important role of the earlier triggered alarms. The attenuation coefficient was determined via an efficient approach based on the time information of historical alarm flood data. The performance of the developed approach is validated by the simulations on the TEP benchmark and a real industrial dataset.

Chapter 3

A Modified Bag-of-Words Representation for Industrial Alarm Floods [*]

In this chapter, a vector representation inspired by NLP is developed, which could capture the important features of alarm flood sequences including the chronological order of triggered alarms. For defining feature vectors, a weighting strategy is proposed such that the ranking of alarms is provided to help operators distinguish key alarms relevant to specific abnormal situations. Offline similarity analysis of alarm floods is addressed by grouping similar alarm flood vectors via an ML-based clustering method while using an efficient similarity measurement. An evaluation of the proposed approach is made using the TEP benchmark.

This chapter is organized as follows. Section 3.1 gives a brief review of the alarm flood detection. Section 3.2 presents an MBoW vector representation for alarm floods. Section 3.3 describes the utilized similarity measurement and clustering method. Section 3.4 provides a case study to validate the proposed approach. Finally, Section 3.5 concludes this chapter.

[*]The material in this chapter has been published as: Haniyeh Seyed Alinezhad, Jun Shang, and Tongwen Chen, "A modified Bag-of-Words representation for industrial alarm floods," *in 9th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, Vancouver, BC, August 2022.

3.1 Historical Alarm Flood Detection

The history of alarms activated during industrial plant operation is stored in a database known as the A&E log, which can be used to detect historical alarm floods for further analysis. There is a class of nuisance alarms known as chattering alarms, which are typically encountered in stored alarm data. As the name implies, a chattering alarm refers to an alarm tag that frequently switches between normal and abnormal states within a short amount of time. For instance, a noisy process variable fluctuating around an alarm limit could cause alarm chatters. Chattering alarms mistakenly increase the number of alarm annunciations over time, which could mask real alarm characteristics and distract operators from managing alarm floods properly. However, the alarm tags corresponding to the chattering alarms carry valuable information for plant operators. Thus, eliminating the alarm chatters while preserving the associated alarm tag information is essential prior to alarm flood detection to conduct an efficient alarm flood analysis. Multiple alarm suppression methods are available for removing chattering alarms, such as delay timers, deadbands, and filters. A detailed discussion of chattering alarm removal falls outside the scope of this study.

After eliminating the repeating information of chattering alarms, a pre-processed A&E log is used to detect alarm floods. Detecting alarm flood can be accomplished through a sliding time window, which counts the number of activated alarms within a certain period of time. It is then possible to define general conditions that determine when an alarm flood begins and ends as follows:

> **Start of an Alarm Flood:** An alarm flood begins when there are at least α alarm messages appearing in a time interval τ.
>
> **End of an Alarm Flood:** An alarm flood ends when there are less than $\delta\alpha$ alarm messages in a time interval τ, where $\delta \in (0, 1]$.

In the above conditions, τ represents the width of the sliding time window,

and parameters can be determined depending on the studied application. As an example, the industrial standard ANSI/ISA-18.2 assumes that alarms are generated at a rate of at least 10 per 10 minutes during an alarm flood [50]. The sliding time window is applied to the A&E log to identify the historical alarm floods, which are then stored as time-stamped alarm sequences in a historical alarm flood dataset.

3.2 MBoW Representation for Alarm Floods

For addressing alarm flood problems in the field of machine learning, a major challenge is making ML-based algorithms understand alarm flood sequences. In this section, an NLP-inspired vectorization strategy is developed to represent alarm floods as feature vectors that are well suited to ML techniques. The goal is to convert an alarm flood from a sequence of time-stamped alarms into a feature vector such that:

- The information regarding activated alarms and their triggering times is preserved.
- Alarms are ranked by importance to assist operators in finding alarms related to the underlying cause of an abnormal condition.
- The accuracy of early classification is taken into account, which is advantageous for online applications.

NLP is an area of Artificial Intelligence (AI) that gives computers the ability to understand human language and analyze text automatically. In NLP, text vectorization refers to the process of converting text data into numerical vectors that can be employed for tasks such as topic classification through ML algorithms. Bag-of-Words is a common model for representing text data as feature vectors, which indicates the words appearing within a specific document.

We propose a modified Bag-of-Words (MBoW) representation for alarm floods by assuming an alarm flood to be a document and each triggered alarm

as a term. Although alarm floods lack some of the complexities associated with text data, such as semantic analysis, triggered alarms are time-stamped, and the ability to properly analyze the temporal context of alarms is critical. With this in mind, and considering the above-mentioned objectives, we propose the following alarm weighting strategy for defining alarm flood vectors:

$$\mathrm{W}(a, F) = \mathrm{TF}(a, F) \times \mathrm{IDF}(a) \times \mathrm{TW}(a, F) \tag{3.1}$$

where $\mathrm{W}(a, F)$ denotes the weight defined for a unique alarm a activated in the flood $F \in \mathbb{F}$, where $\mathbb{F}$ is the set of historical alarm floods.

$\mathrm{TF}(a, F)$ and $\mathrm{IDF}(a)$ are defined based on the term frequency (TF) and inverse document frequency (IDF) concepts from NLP as

$$\mathrm{TF}(a, F) = \frac{|F_a|}{|F|} \tag{3.2}$$

$$\mathrm{IDF}(a) = \log_e \frac{|\mathbb{F}|}{|\{F \in \mathbb{F} | a \in F\}|} \tag{3.3}$$

Here $|F_a|$ denotes the number of annunciations of alarm a in flood F; $|F|$ represents the total number of alarm activations in F; $|\mathbb{F}|$ is the total number of alarm floods in historical dataset; the denominator in (3.3) is the number of historical alarm floods which include alarm a.

The third term in (3.1) is the time weight, denoted as $\mathrm{TW}(a, F)$, which is defined as follows:

$$\mathrm{TW}(a, F) = \log_e \frac{t_{\max}}{t_{a,F}} \tag{3.4}$$

where $t_{\max}$ denotes the maximum triggering time and $t_{a,F}$ is the measure of the time interval between the triggering time of alarm a, and the first annunciated alarm in the alarm flood F. To calculate $t_{a,F}$, the first annunciated alarm in the flood is taken as a baseline and the time distances between subsequent alarms are calculated from this baseline.

According to the measures defined in (3.2) and (3.3), alarms in an alarm flood are weighted depending on how frequent and discriminative they are

compared to the historical data. Generally, the frequency of alarm activations in an alarm flood can provide a representative idea of the nature of the alarm flood. However, some alarms appear frequently and are common among many alarm floods, but they may not be strongly related to the origin of the abnormality. In this case, IDF index adjusts the alarm weight by allocating a lower weight to this type of alarms.

Time weights are defined to embed the temporal information of alarm floods into alarm flood feature vectors, while considering the significance of earlier activated alarms. Through the proposed weights, it is possible to reflect the chronological order of alarms in feature vectors as a result of incorporating the temporal information of triggered alarms. This measure would avoid any possible ambiguity regarding the order in which alarms are recorded in A&E logs.

The proposed MBoW is a vector, denoted as $F^{MBoW} \in \mathbb{R}^n$ where n is the total number of unique alarms configured for the plant. In the alarm flood feature vectors, each entry indicates the alarm's weight, calculated according to the strategy proposed in (3.1). The proposed vector representation captures important features of alarm flood sequences and can be used to perform ML-based alarm flood similarity analysis. Additionally, by providing a ranking of alarms, it could help operators identify the alarms that might be more relevant to the abnormal situation.

3.3 Alarm Flood Similarity Analysis

Similarity analysis for alarm floods is considered in this section, which uses an ML-based clustering method in conjunction with an efficient similarity measurement to group similar alarm flood vectors. Using data clustering, similar alarm flood vectors in an unlabeled historical dataset can be grouped in the same clusters to obtain insights from historical data and help plant operators in managing alarm flood problem.

3.3.1 Similarity Measure

As part of clustering the MBoW alarm flood vectors, it is necessary to determine how to define the similarity between alarm flood data. Flood vector entries are valued based on the triggering time and frequency of the alarms associated with them. The presence of noise during real plant operation may influence recorded data. As a result, there would be two alarm floods within the same category, each with a different size and number of alarms activated. Additionally, the alarm triggering time might also be different between the two, even though alarms are activated in a similar chronological order. Consequently, two alarm floods in the same category may differ in size. An analogy for similarity analysis between feature vectors can be made by using the Euclidean distance, which is a dissimilarity index commonly used in ML algorithms. The Euclidean distance depends on the size of feature vectors, so based on this metric, two similar alarm floods could be considered far apart in our application. To reduce the sensitivity of similarity measurement to vector sizes and improve similarity analysis, we employ the cosine similarity.

Cosine similarity is a measure for assessing how similar the alarm flood vectors are regardless of their sizes. It shows the orientation of the MBoW vectors in the alarm flood vector space, where each dimension corresponds to an alarm in the monitoring system. For a historical dataset with m alarm flood vectors, denoted as $\mathbb{F} = \{F_1^{MBoW}, F_2^{MBoW}, \cdots, F_m^{MBoW}\}$, the cosine similarity for each pair of alarm floods is calculated as follows:

$$\cos_{sim}(F_i^{MBoW}, F_j^{MBoW}) = \frac{F_i^{MBoW} \cdot F_j^{MBoW}}{\|F_i^{MBoW}\| \|F_j^{MBoW}\|} \tag{3.5}$$

where, $\| \cdot \|$ is the Euclidean norm of the vector, and "$\cdot$" denotes the dot product of two vectors.

Cosine similarity is a measure of similarity between two vectors in multidimensional space using the cosine of the angle between the vectors. For MBoW alarm flood vectors, it can take values between zero and one, with zero representing no similarity and one representing identical matches. Therefore, a

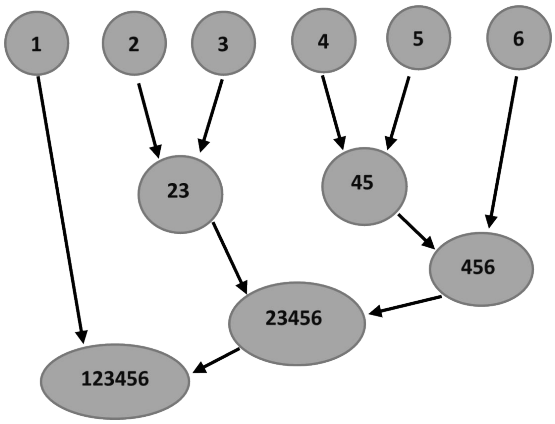

Figure 3.1: An example for AHC tree

smaller angle between vectors indicates a greater degree of similarity.

3.3.2 Similarity Analysis

Based on the cosine similarity measure, alarm flood similarity analysis is carried out using agglomerative hierarchical cluster (AHC) tree method. An AHC tree is a clustering algorithm that starts by assigning each observation to a distinct cluster. Afterward, it merges each pair of similar clusters gradually until all clusters are merged into one, or when the highest pairwise similarity measure for clusters falls below a predefined threshold. An illustrative example for the process of constructing an AHC tree is depicted in Figure 3.1. In this study, the similarity between the clusters is measured based on the average linkage criteria [22].

To provide plant operators with a visual tool, the pairwise similarity matrix for historical alarm floods is calculated and color-coded. As a result, cluster color maps are visualized by reordering the similarity scores based on the clustering order of the AHC tree. By cutting off the clustering process at a certain point, one would be able to get a clustering solution. In this study, alarm flood vectors with a pairwise cosine similarity index greater than 0.6 are grouped together.

The proposed alarm flood similarity analysis is intended to develop an operator assistance system using retrospective analysis of historical data. Group-

ing similar alarm floods in the historical data could facilitate the handling of large datasets by operators. Moreover, the provided scores for alarms could help operators in locating the cause of the abnormality associated with each alarm flood group. Therefore, annotated or labeled groups of alarm floods can be provided by considering the data of each cluster from the same category of abnormal situation. These categorized historical alarm floods can be also used as a valuable dataset for online alarm flood analysis.

3.4 Case Study

3.4.1 Data Description

For assessing the effectiveness of the proposed method, the TEP benchmark with a closed-loop plant simulator developed in [11] is employed. A total of 39 process variables are used to simulate the alarm system, where four states, namely "PVH", "PVL", "PVHH", and "PVLL" are defined for alarm configuration. There is not enough space to include more detail about process variables. A complete list of variables and alarm limits for the different states can be found in [86].

Over the simulation period of 10 hours, process measurements are sampled uniformly every 0.6 minutes. The alarm floods are generated by setting the parameters $\tau = 10$ min, $\alpha = 8$, and $\delta = 0.75$ according to the standard definition presented in Section 3.1. The dataset used for this study consists of 150 alarm flood sequences from 5 categories, each with 30 alarm floods. A list of faults corresponding to the root cause categories of alarm floods in historical data is provided in Table 3.1.

3.4.2 Similarity Analysis of Alarm Floods

All historical alarm flood sequences are first converted to the MBoW vector format. Using the dataset including historical flood vectors, pairwise cosine similarity between all flood vectors is calculated and used to generate an AHC tree. Cluster color maps are presented in Figure 3.2 as visualizations of clus-

Table 3.1: Description of five categories of alarm flood causes in TEP

Fault	Description	Type
C_1	A/C feed ratio, B composition constant (Stream 4)	Step
C_2	B composition, A/C feed constant (Stream 4)	Step
C_3	Reactor cooling water inlet temperature	Step
C_4	C header pressure loss (Stream 4)	Step
C_5	Reactor cooling water valve	Sticking

tering results. A darker color on the color map indicates greater similarity between sequences.

For comparison, three cases for similarity analysis are provided in Figure 3.2. The color map presented in Figure 3.2(a) corresponds to the method proposed in this study. Figure 3.2(b) shows the clustering result based on cosine similarity and MBoW feature vectors without temporal information. The similarity color map in Figure 3.2(c) shows the similarity analysis of MBoW vectors based on the Euclidean distance. These results confirm that the proposed method achieves better discrimination between alarm floods belonging to different categories compared with the other cases. As seen in Figure 3.2(b), discarding the time information in alarm flood feature vectors results in identifying more similarity between different alarm flood categories. It could be the result of ignoring the chronological order of alarm sequences in alarm floods, which leads to comparing alarm floods based only on their triggered alarms. Figure 3.2(c) also shows that when analyzing the similarity of MBoW vectors cosine similarity behaves better than the Euclidean distance and makes a better distinction. Using the cut off threshold of 0.6 for similarity measures in the AHC tree, the proposed method achieves 99.33% purity. Here purity is a cluster validity index and its definition can be found in (2.12).

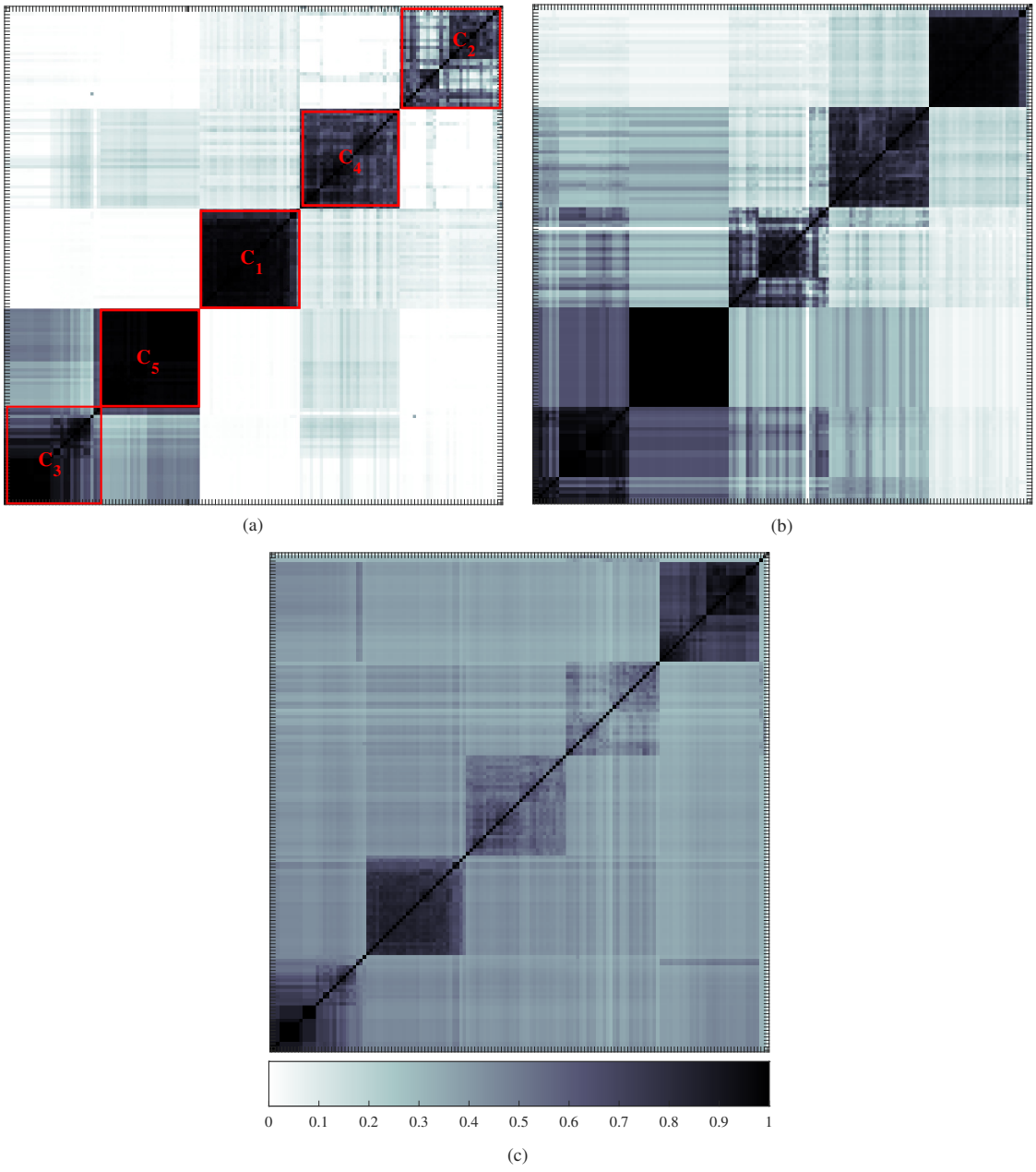

Figure 3.2: Cluster color maps derived from similarity analysis of alarm floods: (a) The proposed Method. (b) Bag-of-Words representation excluding temporal information [32]. (c) Similarity analysis of MBoW vectors based on Euclidean distance.

After similarity analysis and clustering historical alarm floods, our method provides operators with a list of scored alarms for each alarm flood. The ranks of alarms and their corresponding scores could facilitate finding the fault category for each alarm flood group. To demonstrate the effectiveness of the proposed alarm scoring method, Table 3.2 shows the list of alarms and their scores for an alarm flood from fault category C_5. Table 3.1 explains that this alarm flood is due to a failure of the reactor cooling water valve. According to the alarm scores and their descriptions, we can see that higher scores are assigned to the alarms that are more relevant to this fault category.

Table 3.2: Alarm scoring results for an alarm flood from category C_5

Alarm Rank	Alarm Index	Alarm Identifier	Alarm Scores	Description
1	137	LL	0.71	Reactor cooling water outlet temperature
2	59	L	0.67	Reactor cooling water outlet temperature
3	87	HH	0.49	Reactor temperature
4	126	LL	0.43	Reactor temperature
5	9	H	0.39	Reactor temperature
6	48	L	0.31	Reactor temperature
7	98	HH	0.04	Reactor cooling water outlet temperature
8	7	H	0.02	Reactor pressure
9	20	H	14×10^{-3}	Reactor cooling water outlet temperature
10	13	H	12×10^{-3}	Product separator pressure
11	46	L	10×10^{-3}	Reactor pressure
12	52	L	5×10^{-3}	Product separator pressure
13	55	L	3×10^{-3}	Stripper pressure
14	16	H	17×10^{-4}	Stripper pressure
15	50	L	10×10^{-4}	Product separator temperature
16	60	L	68×10^{-5}	separator cooling water outlet temperature
17	44	L	61×10^{-5}	Recycle flow (stream 8)
18	33	H	42×10^{-5}	Component F (stream 9)

3.5 Conclusion

In this chapter, a vector representation model inspired by NLP for industrial alarm floods was developed. This representation is capable of reflecting the important characteristics of the triggered alarms in an alarm flood sequence, including their chronological orders. Proposing alarm flood fea-

ture vectors is mainly intended to make them useful for ML-based similarity analysis algorithms. An advantage of using ML-based strategies is that they can be easily adapted for online applications. A weighting strategy was proposed to define alarm flood feature vectors so that a degree of importance for alarms can be provided to help operators recognize key alarms relevant to specific abnormal situations. Furthermore, the accuracy for early classification of alarm floods was taken into account when embedding temporal information into feature vectors to make them appropriate for online applications. Alarm flood similarity analysis was conducted by grouping similar alarm flood vectors through AHC trees and cluster color maps by using the cosine similarity measurement. To reduce sensitivity against sizes of feature vectors, the cosine similarity index was used to measure the similarity between alarm floods. Utilizing the TEP benchmark, an evaluation of the proposed approach was conducted and favorable results were obtained.

Chapter 4

Open Set Online Classification of Industrial Alarm Floods with Alarm Ranking [*]

In this chapter, an ML-based early classification of alarm floods is developed, which is capable of handling online alarm floods corresponding to new abnormal situations. Online alarm floods are modeled as feature vectors using an online extension of the weighting strategy developed in the previous chapter. Online feature vectors incorporate the triggering times of alarms associated with alarm floods such that early classification accuracy is enhanced. Besides reflecting important characteristics of alarm flood sequences, such as temporal information, the proposed vector representation provides ranking of alarms to help operators find key alarms that are relevant to abnormal situations. Modeling alarm floods as feature vectors makes them suitable for ML-based methods, which have been found to be effective in a variety of applications. Although the problem of online alarm flood classification was studied in the literature, dealing with previously unseen alarm flood scenarios has not been fully investigated. ML-based classifiers are classically constructed under the closed set assumption, where all upcoming data must be from categories that previously appeared in the historical dataset. This assumption

[*]The material in this chapter has been published as: Haniyeh Seyed Alinezhad, Jun Shang, and Tongwen Chen, "Open set online classification of industrial alarm floods with alarm ranking," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. article 3500811, 2023.

may be valid in many situations, but it is generally a restrictive assumption. As previously unseen situations can occur in practice, a more realistic scenario needs to be considered. This chapter proposes an ML-based open set classification method based on a systematic similarity threshold estimation to deal with previously unseen situations. A classifier with an open set classification capability can avoid incorrect classifications by excluding samples with low classification confidence via incorporating a reject option. The survey paper presented in [34] provided a comprehensive review of existing open set classification methods. The proposed classification method along with alarm ranking can help operators in making timely decisions to handle both previously seen and new alarm flood scenarios. A case study by using the Tennessee Eastman benchmark is illustrated to assess the effectiveness of the proposed method.

This chapter is organized as follows. In Section 4.1, alarm flood detection and the developed alarm flood feature vectors are presented. Section 4.2 discusses the proposed open set classification and alarm ranking. A case study is presented in Section 4.3 to evaluate the efficiency of the proposed method. Finally, Section 4.4 concludes the chapter.

4.1 Alarm Floods

This section demonstrates how alarm floods can be detected using both historical and real-time alarm data. Then an alarm weighting strategy is proposed to convert alarm floods from alarm sequences to feature vectors.

4.1.1 Alarm Flood Detection

Alarms are set up in industrial alarm systems to detect process deviations from predetermined normal operating ranges, which serve to warn plant operators. Let $A = \{a_i, i = 1, 2, \ldots, n\}$ denote the set of unique alarm variables configured in an alarm system. The alarm occurrence signal for a_i can be

defined as follows:

$$\mathcal{A}_i(t) = \begin{cases} 1, & \text{if } \mathcal{X}_i(t-s) \in \mathcal{N}_i \ \& \ \mathcal{X}_i(t) \notin \mathcal{N}_i \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

where $\mathcal{X}_i$ indicates a process variable whose normal operating characteristics are represented by $\mathcal{N}_i$, and s denotes the sampling size.

As the name implies, an alarm flood is a situation in which plant operators are overwhelmed with many alarms within a short period of time. Thus, general conditions can be derived for detecting the start and the end of an alarm flood based on the alarm rate. At time instant t, the alarm rate for a past time interval of size T can be calculated as follows:

$$\mathcal{R}(t) = \sum_{i=1}^{n} \sum_{j \in [t-T,t]} \mathcal{A}_i(j). \tag{4.2}$$

Here a T-width sliding window is used to count the number of triggered alarms. An alarm flood can be detected by comparing the calculated alarm rate with predetermined thresholds. This can be accomplished by defining an indicator for the occurance of alarm floods as

$$\lambda_t = \begin{cases} 1, & \text{if } \lambda_{t-\delta} = 0, \ \mathcal{R}(t) \geq \gamma \\ 1, & \text{if } \lambda_{t-\delta} = 1, \ \mathcal{R}(t) \geq \epsilon\gamma \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

where $\epsilon \in (0, 1]$ and δ is the step size of the sliding window. When the indicator is one, the sequence of alarms detected by the sliding window is considered to be an alarm flood. It is easy to verify the start and end time of an alarm flood using this indicator. An alarm flood starts at time instant $t_s = t_d - T$ if $\lambda_{t_d-\delta} = 0$ and $\mathcal{R}(t_d) \geq \gamma$. Also, an alarm flood ends at time instant t_e when $\lambda_{t_e-\delta} = 1$ and $\mathcal{R}(t_e) < \epsilon\gamma$. For instance, according to the ANSI/ISA-18.2 standards [50], the values of the parameters T, ϵ, and γ are 10 min, 0.5 and 10 respectively. This indicates that an alarm flood begins when the rate of activated alarms reaches the threshold of 10 alarms per 10 minutes and ends when less than 5 alarms are triggered within a 10-minute period. The described strategy can be used to identify the occurrence of alarm floods in both offline and online scenarios.

It should be noted that there are some alarm tags that frequently switch between normal and abnormal states in a short period of time, known as chattering alarms. The presence of chattering alarms increases the alarm rate and can lead to nuisance alarm flood detection. Thus, it is essential to remove chattering alarms before alarm flood diagnosis. Multiple well-established methods are available for reducing chattering alarms in both offline and online applications [45, 62]. It is beyond the scope of this study to discuss reduction of chattering alarms. Hence, in the remainder of this chapter, it is assumed that chattering alarms have been reduced, at least in the alarm data collected.

Offline Detection

The offline detection of alarm floods is possible through the A&E logs that contain historical alarm data. An A&E log includes several attributes including the alarm tag and triggering time stamp for alarms that are activated during plant operation. By applying the sliding time window to the A&E log, alarm floods are detected as time-stamped alarm sequences and stored in a historical alarm flood dataset that can be used for further analysis. Each historical alarm flood sequence is recorded as follows:

$$f_k = \{(\alpha_1^k, t_{\alpha_1}^k), (\alpha_2^k, t_{\alpha_2}^k), \ldots, (\alpha_{|f_k|}^k, t_{\alpha_{|f_k|}}^k)\}. \tag{4.4}$$

Here $|\cdot|$ indicates the number of elements in a set or sequence; $f_k \in \mathcal{F}$ is the kth alarm flood of the historical alarm flood dataset $\mathcal{F}$, $k = 1, 2, \ldots, |\mathcal{F}|$; $\alpha_m^k \in A$ represents the mth triggered alarm tag in f_k, $m = 1, 2, \ldots, |f_k|$; $t_{\alpha_m}^k$ denotes the triggering time stamp corresponding to α_m^k that belongs to the time interval $[t_s^k, t_e^k]$; t_s^k and t_e^k are the start time and end time of f_k.

Online Detection

A T-width sliding time window can also be used for real-time calculation of the alarm rate, which can be utilized for identifying online alarm floods. Figure 4.1 demonstrates how (4.3) can be used to detect an ongoing alarm flood using online alarm events. An online alarm flood f_o^1 is detected at time

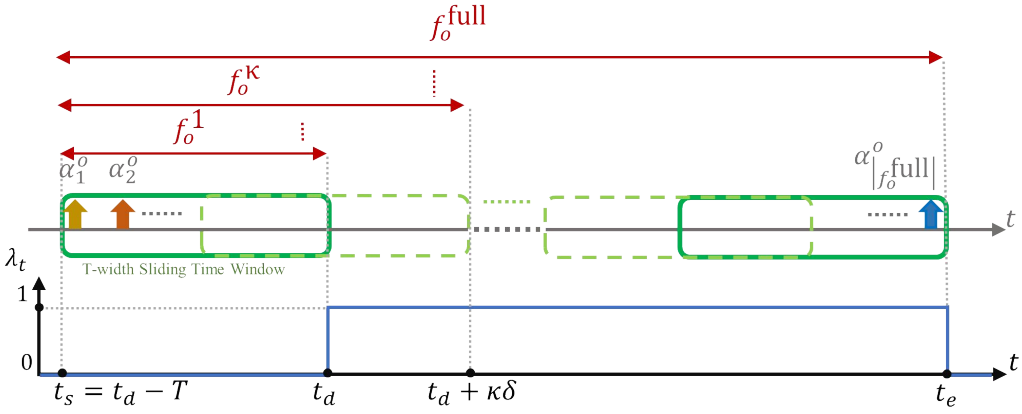

Figure 4.1: Online alarm flood detection.

t_d when λ_t changes from zero to one. The record of the T-width time window is updated during online plant operation. In each update, alarm records beyond the time span T are removed and new alarms are added to calculate the alarm rate. As long as $\lambda_t = 1$ the alarm record in the online alarm flood is updated. Once λ_t changes from one to zero the end of the alarm flood is detected. Different updates of an online alarm flood can be recorded as follows using the corresponding alarms and time stamps:

$$
\begin{aligned}
& f_o^1 = \{(\alpha_1^o, t_s),\ (\alpha_2^o, t^o_{\alpha_2}), \ldots,\ (\alpha^o_{|f_o^1|}, t^o_{\alpha_{|f_o^1|}})\} \\
& \vdots \\
& f_o^{\kappa} = \{(\alpha_1^o, t_s),\ (\alpha_2^o, t^o_{\alpha_2}), \ldots, (\alpha^o_{|f_o^1|}, t^o_{\alpha_{|f_o^1|}}), \\
& \qquad \ldots,\ (\alpha^o_{|f_o^{\kappa}|}, t^o_{\alpha_{|f_o^{\kappa}|}})\} \\
& \vdots \\
& f_o^{\text{full}} = \{(\alpha_1^o, t_s),\ (\alpha_2^o, t^o_{\alpha_2}), \ldots, (\alpha^o_{|f_o^1|}, t^o_{\alpha_{|f_o^1|}}), \\
& \qquad \ldots,\ (\alpha^o_{|f_o^{\kappa}|}, t^o_{\alpha_{|f_o^{\kappa}|}}),\ \ldots,\ (\alpha^o_{\left|f_o^{\text{full}}\right|}, t_e)\}.
\end{aligned}
\tag{4.5}
$$

Here f_o^{κ} and f_o^{full} represent the κth and last update of the alarm flood respectively; α_r^o denotes the rth triggered alarm tag in the online alarm flood, $r = 1, 2, \ldots, |f_o^{\text{full}}|$; $t^o_{\alpha_r}$ is the triggering time stamp corresponding to α_r^o; t_s and t_e are the start time and end time of f_o^{full} respectively.

4.1.2 Alarm Flood Feature Vectors

ML techniques can be used to investigate the problem of alarm floods efficiently in both offline and online applications. In applying ML-based methods to alarm flood analysis, one of the most challenging parts is to make the ML algorithms understand the alarm floods, which are sequences of time-stamped alarms. In the following, a vectorization method inspired by NLP is proposed to represent alarm floods as feature vectors, which can be used as suitable inputs for ML-based methods.

Alarm Weighting Strategy

NLP is a branch of artificial intelligence that enables computers to understand and analyze text automatically. As a part of NLP, text vectorization aims at representing a text as a numerical vector, which can be used to accomplish tasks such as topic classification with ML algorithms. Bag-of-Words is a widely-accepted numerical vector model used for text vectorization that shows whether a text contains or excludes certain words [52]. We propose a modified Bag-of-Words (MBoW) model for alarm flood vectorization by considering each alarm flood as a document and the alarms associated with it as terms.

Alarm flood data is not subject to the same challenges as text data, e.g., semantic complexities. However, considering the temporal context of alarm sequences is an important factor in the vectorization of alarm floods. To achieve effective alarm flood vectorization, we aim to come up with an alarm weighting strategy, which 1) preserves information about triggered alarms and their chronological order; 2) offers a ranking of alarms reflecting their relevance to the underlying abnormal condition; 3) considers the early classification accuracy for online situations. Accordingly, the weight of the alarm α in the alarm flood f, which is denoted as $\mathrm{W}(\alpha, f)$, is defined as follows:

$$\mathrm{W}(\alpha, f) = \mathrm{TF}(\alpha, f) \times \mathrm{IDF}(\alpha) \times \mathrm{TW}(\alpha, f). \tag{4.6}$$

The first two terms, $\mathrm{TF}(\alpha, f)$ and $\mathrm{IDF}(\alpha)$, are defined based on two NLP con-

cepts, namely, the term frequency (TF) and the inverse document frequency (IDF) as follows:

$$\mathrm{TF}(\alpha, f) = \frac{N(\alpha, f)}{|f|} \tag{4.7}$$

$$\mathrm{IDF}(\alpha) = \log_e \frac{|\mathcal{F}|}{|\mathcal{F}^\alpha|}. \tag{4.8}$$

Here $N(\alpha, f)$ indicates the number of activations of alarm α in flood f; $|f|$ is the total number of triggered alarms in f; $|\mathcal{F}|$ represents the number of alarm floods recorded in historical dataset; $\mathcal{F}^\alpha$ denotes the set of historical alarm floods that contain alarm α and $|\mathcal{F}^\alpha|$ represents the total number of elements in this set.

The term $\mathrm{TW}(\alpha, f)$ in (4.6) denotes the time weight which is defined as

$$\mathrm{TW}(\alpha, f) = \log_e \frac{\tau_{\max}}{\tau_{\alpha,f}} \tag{4.9}$$

where $\tau_{\alpha,f}$ is the time distance between the start time of the alarm flood f and the time instant when the alarm α is activated in f, and $\tau_{\max}$ denotes the maximum alarm flood duration in historical data.

The terms defined in (4.7) and (4.8) are employed in the weighting strategy to capture the frequency and significance of each alarm in an alarm flood. The TF is intended to reflect the general characteristics of an alarm flood by calculating the frequency of each unique alarm. There can be alarm activations that are frequent and common to several alarm floods, which are not necessarily related to the source of the abnormality. IDF is utilized in this case for weight adjustments by assigning lower weights to this type of alarms. The TW defined in (4.9) is included in the weighting strategy to incorporate the temporal characteristics of the alarm sequences into the alarm flood feature vectors. It aims at capturing the chronological order of alarms while preserving the importance of earlier activated alarms. It would also avoid potential ambiguity in the order that alarms appear in the A&E log.

MBoW Feature Vectors

The proposed alarm weighting strategy is used to convert alarm floods from alarm sequences to MBoW feature vectors. An MBoW alarm flood vector is an n-dimensional vector, where n is the number of unique alarm tags configured for the plant. Each entry of this feature vector represents the weight of its corresponding alarm tag calculated by (4.6)–(4.9).

To build an MBoW vector for an alarm flood f, it is required to calculate the weights of the unique alarms $a_i \in A$, $i = 1, 2, \ldots, n$, configured for the plant. To accomplish this, $\mathrm{IDF}(a_i)$ and $\tau_{\max}$ can be derived by using historical data, and after the calculation of $\mathrm{TF}(a_i, f)$, $\mathrm{TW}(a_i, f)$ is obtained by computing the time distance corresponding to each unique alarm tag as

$$\tau_{a_i,f} = \begin{cases} \min\limits_{l}(t_{a_i,l} - t_s^f) & \text{if } \mathrm{TF}(a_i, f) \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.10}$$

where l is the number of activations of a_i in f and $t_{a_i,l}$ denotes the time stamp associated with each activation. In (4.10), $\tau_{a_i,f}$ is calculated by taking the start time of the f, denoted as t_s^f as a baseline.

Following the above calculations, the MBoW vector for alarm flood f, denoted as $\boldsymbol{F}$, can be constructed as follows:

$$\boldsymbol{F} = [\mathrm{W}(a_1, f), \mathrm{W}(a_2, f), \ldots, \mathrm{W}(a_n, f)]^T. \tag{4.11}$$

The MBoW vector model can represent important characteristics of alarm flood sequences and can be used as a suitable input to ML-based analysis of alarm floods. In addition, the proposed alarm weighting can provide plant operators with a ranking of alarms, which can be helpful in identifying alarms that are more likely to be related to underlying abnormal conditions.

4.2 Classification of Online Alarm Floods

The classification of online alarm floods into groups of similar historical alarm floods can be viewed as an operator assistance mechanism for safe operation. This can facilitate handling the abnormal situations corresponding

to the classified alarm floods by providing information about similar historical alarm floods and their causes. Early classification of an ongoing alarm flood is important for enabling plant operators to take timely and efficient safety measures without waiting for the alarm flood to end, especially in the case of long-duration alarm floods.

4.2.1 Problem Definition

Alarm flood classification can be accomplished by developing a method based on machine learning, which has been found effective in various applications. Let $\mathcal{H}$ denote a training dataset defined as

$$\mathcal{H} = \{(\digamma_1, L_1), (\digamma_2, L_2), \ldots, (\digamma_{|\mathcal{F}|}, L_{|\mathcal{F}|})\} \tag{4.12}$$

where $\digamma_k = [\mathrm{W}(a_1, f_k), \mathrm{W}(a_2, f_k), \ldots, \mathrm{W}(a_n, f_k)]^T$ is the MBoW feature vector corresponding to the kth historical alarm flood f_k, and $L_k \in \mathcal{C}$ represents the class label of the kth alarm flood belonging to the set of labels corresponding to alarm flood categories denoted by $\mathcal{C} = \{c_1, c_2, \ldots, c_\rho\}$. The category information in $\mathcal{C}$ can be arbitrarily selected and used to train and implement the classifier based on the training dataset defined in (4.12). Regardless of what categorical label is selected for each alarm flood class, each category is characterized by some descriptions or annotations related to its corresponding abnormal condition. When classification is performed, the information associated with the predicted category for the classified alarm flood is provided for on-site operators to help them diagnose the root cause. It is possible to label and annotate the categories in the training dataset with the help of plant experts and historical records. The preparation of the training dataset can be facilitated using offline alarm flood management approaches, such as the method presented in the previous chapter, which help handle large datasets by providing insight into similar historical alarm floods.

An ML-based model for online alarm flood classification can be developed based on the training dataset represented in (4.12). A classic ML-based classifier works under the closed set assumption, where all the upcoming data being

classified must be from classes that appeared in the historical dataset. While this can be true in many cases, it is generally a restrictive assumption. Because obtaining historical samples that cover all possible scenarios is difficult and previously unseen situations can occur in practice. Thus, a more realistic scenario needs to be considered.

We aim to address the problem of online alarm flood classification by building a classifier that can accurately classify previously observed alarm floods into the categories in the training dataset and identify unseen alarm floods caused by new abnormal situations. The open set classification refers to such a scenario, where new situations can be handled by incorporating a reject option that allows the classifier to avoid incorrect classifications by excluding samples with low classification confidence [34].

In online classification, an immediate classification should be initiated upon detection of the ongoing alarm flood f_o. As new alarms are triggered in the ongoing alarm sequence, the classification result should be updated until the alarm flood ends. The input of the classifier at every update requires to be in the form of a MBoW feature vector as defined in (4.11). Thus, alarm sequences identified in each update of the online alarm flood detection process shown in (4.5) need to be converted to MBoW vectors as follows:

$$\begin{aligned}
&\boldsymbol{F}_o^1 = [\mathrm{W}(a_1, f_o^1),\ \mathrm{W}(a_2, f_o^1), \ldots,\ \mathrm{W}(a_n, f_o^1)]^T \\
&\vdots \\
&\boldsymbol{F}_o^\kappa = [\mathrm{W}(a_1, f_o^\kappa),\ \mathrm{W}(a_2, f_o^\kappa), \ldots,\ \mathrm{W}(a_n, f_o^\kappa)]^T \\
&\vdots \\
&\boldsymbol{F}_o^{\mathrm{full}} = [\mathrm{W}(a_1, f_o^{\mathrm{full}}),\ \mathrm{W}(a_2, f_o^{\mathrm{full}}), \ldots,\ \mathrm{W}(a_n, f_o^{\mathrm{full}})]^T.
\end{aligned} \tag{4.13}$$

With the proposed alarm weighting strategy, classification of ongoing alarm floods can serve as an online decision support tool for plant operators by providing early alarm flood classification and alarm ranking.

4.2.2 Open Set Classification

The following proposes a multi-class open set classification model aiming at classifying each online alarm flood to one of the historical categories in $\mathcal{C}$

or rejecting it to indicate that it is from an unseen or new class, i.e., it is not from any of the ρ seen classes. This is accomplished by using the concept of logistic regression (LR) for classification based on thresholding the output class probability.

Logistic Regression

LR is a probabilistic classification method that can make binary decisions about input observations. It performs the classification task based on the class probabilities, which are calculated with a sigmoid function as $h_\theta(x) = \text{sigmoid}(z)$, where $z = w^T x + b$. For an input vector x, $h_\theta(x)$ represents the probability of belonging to a class that is characterized by parameter $\theta = (w, b)$. The parameter θ is obtained from a training process, which is based on solving an optimization problem using historical data with the aim of maximizing the likelihood of the class label of the training observations being correct [53].

When the optimal value of θ (denoted as θ^*) is obtained, the LR classifier can predict a class probability for a newly observed input observation. The output of an LR classifier, denoted by y, can be either 1 indicating that the given input observation belongs to the class or 0 indicating that it does not belong to the class. Thus, the probability of x_{new} being a member of the class (being in the positive class) is $\Pr(y = 1|x_{\text{new}}) = h_{\theta^*}(x_{\text{new}})$, and the probability of not being a member of the class (being in the negative class) is $\Pr(y = 0|x_{\text{new}}) = 1 - h_{\theta^*}(x_{\text{new}})$. An input observation is classified into a positive or negative class using a decision boundary for class probability, which is commonly set at 0.5. As a result, a new input instance x_{new} is classified into the positive class when $h_{\theta^*}(x_{\text{new}}) > 0.5$ and into the negative class otherwise.

Multi-Class Classification

As real industrial processes typically involve multiple categories of abnormal situations, the historical data of alarm floods may contain multiple classes. Therefore, developing a multi-class classification strategy is required for ad-

dressing the problem of alarm flood classification. To this end, the one-vs-rest approach is employed to develop a multi-class classification based on the LR algorithm. With this strategy, a binary classifier is fitted for each class, so that each class is compared against all other classes.

With ρ classes of historical alarm floods, a one-vs-rest LR model is trained separately for each class, such that the rth sigmoid function takes all the training observations with $y = c_r$ as positive examples, and all the rest with $y \neq c_r$ as negative examples. Thus, based on training data $\mathcal{H}$ defined in (4.12), the parameter θ_r for the rth one-vs-rest classifier is learned by utilizing the following data:

$$\mathbf{P_r} = \{(x, y) = (\digamma_q, L_q) | q \in \{1, 2, \ldots, |\mathcal{F}|\}, L_q = c_r\}$$
$$\mathbf{N_r} = \{(x, y) = (\digamma_q, L_q) | q \in \{1, 2, \ldots, |\mathcal{F}|\}, L_q \neq c_r\}$$

where $\mathbf{P_r}$ and $\mathbf{N_r}$ are the sets of positive and negative examples respectively, and $r = \{1, 2, \ldots, \rho\}$. For $\mathbf{P_r}$, q denotes the index of historical alarm floods with labels equal to the rth alarm flood category label c_r in the training dataset $\mathcal{H}$, and for $\mathbf{N_r}$, q denotes the index of historical alarm floods with labels other than c_r. Alternatively, $\mathbf{P_r}$ is a subset of training dataset $\mathcal{H}$ including historical alarm floods that belong to the rth class. In contrast, $\mathbf{N_r}$ is a subset of training dataset $\mathcal{H}$ including historical alarm floods that are not of class r. When the multi-class classifier is built using the obtained parameters for each class (denoted as $\theta_r^* = (w_r^*, b_r^*)$), the rth class probability for an online alarm flood $\digamma_o$ can be obtained through the corresponding sigmoid function as

$$h_{\theta_r^*}(\digamma_o) = \text{sigmoid}(z_r) \tag{4.14}$$

where $z_r = w_r^{*T} \digamma_o + b_r^*$. For classification, a decision boundary is set for each trained class, so that if the predicted class probabilities fall below the corresponding threshold, the online alarm flood is rejected; otherwise, it is assigned to the class with the highest probability. The common value for the decision boundary is 0.5, as previously mentioned.

This strategy provides the advantage of training each class independently and enables the classifier to add a reject option by thresholding the class

probability. Since the sigmoid function corresponding to each class considers its input value separately, inspecting trained classifiers can provide insight into each class. The default probability threshold of 0.5 may not be a proper choice for decision making in alarm flood classification. This can result in incorrect classifications in case of previously unseen alarm floods. As a result of the interpretability feature of this classification strategy, it is possible to develop a threshold estimation strategy based on historical data to improve the decision boundaries for each class.

Open Set Classification Decision Boundaries

In the case of online alarm flood classification, the class probability value for a previously unseen example may exceed the default threshold of 0.5 due to some similar alarm tags. As a result, the unseen example is classified incorrectly into one of the seen categories. When training sigmoid functions, positive examples tend to be associated with higher class probabilities, i.e., closer to "1". Hence, the risk of incorrect classification of new observations can be reduced by increasing the probability threshold.

It is possible to estimate the appropriate decision boundaries for each class by using the class probabilities corresponding to training data. To accomplish this, for the rth class, it is assumed that predicted probabilities follow one half of a Gaussian distribution, with a mean of "1". Using the class probabilities as $\Pr(L_k = c_r|\digamma_k)$, the other half of the Gaussian distribution can be generated by calculating probability values as $1 + (1 - \Pr(L_k = c_r|\digamma_k))$. A Gaussian distribution describes the probability of a variable taking on different values. Variables closer to the mean of a Gaussian distribution have a higher probability density, whereas variables away from the mean have a decreasing probability density. We know that the classifier is trained such that the likelihood of the class label of the training observations being correct is maximized. As a result, positive examples are more likely to have class probabilities closer to "1", while negative examples take probability values away from "1". Thus, by fitting a Gaussian distribution with a mean of "1" on class probabilities, a

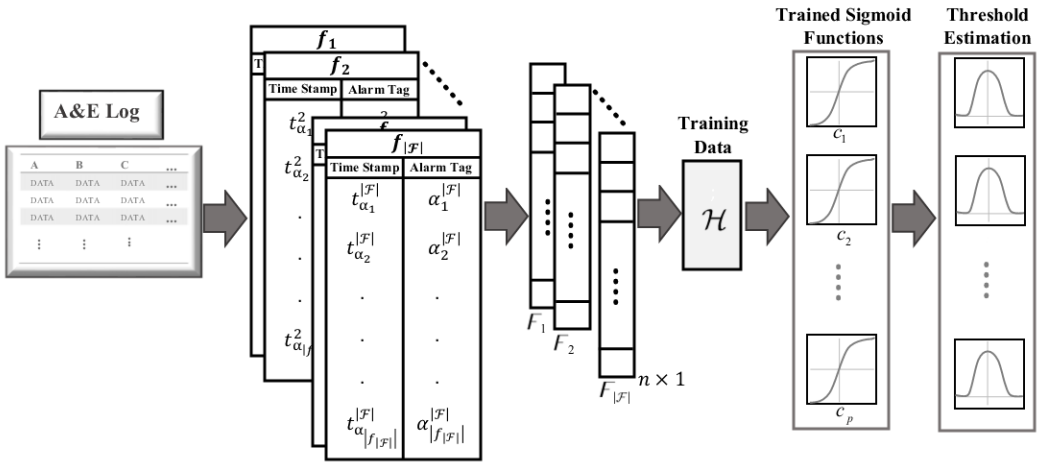

Figure 4.2: The process of training the open set classifier for online alarm flood clasification.

classification threshold can be estimated based on the 95% confidence interval of the Gaussian distribution. It can be calculated based on the standard deviation of the resulting Gaussian distribution, denoted as σ_r, as

$$\xi_r = \max(0.5, 1.96\sigma_r). \tag{4.15}$$

By using the estimated decision boundaries of all ρ classes, the following strategy is defined for the open set classification of online alarm floods, which enables the classifier to reject unseen alarm floods and classify seen alarm floods efficiently:

$$OS(\digamma_o) = \begin{cases} \text{reject} \;\; \text{if} \;\; \text{for} \; r = \{1, 2, \ldots, \rho\}, \; h_{\theta_r^*}(\digamma_o) < \xi_r \\ \arg\max_r h_{\theta_r^*}(\digamma_o), \quad \text{otherwise.} \end{cases}$$

The steps involved in training the proposed open set classifier are shown in Figure 4.2. As shown in this figure, first historical alarm floods are detected from the A&E log and recorded as sequences of time-stamped alarms in the form of (4.4). Then alarm floods are converted from sequences of alarms to MBoW vectors according to the model proposed in (4.11). Historical alarm flood vectors are used to build the training data defined in (4.12) to be used for training the classifier. The training data are then utilized to train the sigmoid functions and their corresponding classification thresholds.

Algorithm 3: Open set classification of alarm floods

Input: Online alarm sequence as defined in (4.5), trained one-vs-rest classifiers, estimated decision boundaries;
Output: Classification decision for the ongoing alarm flood f_o
begin
  while $\lambda_t = 1$ **do**
    Record the time-stamped triggered alarms from t_s to t;
    Calculate the time distance for each alarm by using (4.10);
    Covnert the current update of f_o to MBoW vector $\digamma_o$ according to (4.13);
    for $r = 1$ **to** ρ **do**
      Calculate $h_{\theta_r^*}(\digamma_o)$ by (4.14);
    if $h_{\theta_r^*}(\digamma_o) \leq \xi_r$ **then**
      Identify the online alam flood as a new abnormal situation;
    else
      Make a classification decision according to $OS(\digamma_o) = \arg\max_r h_{\theta_r^*}(\digamma_o)$;
    Update f_o;

The summary of the proposed open set classification for online alarm floods is presented in Algorithm 3. When an online alarm flood from a previously seen category is classified in each update of its occurrence, plant operators are offered online assistance based on the information regarding classified abnormal categories, which can be confirmed further using the provided alarm rankings. If an online alarm flood is discovered as a new abnormal situation, alarm ranking can help plant operators in coping with high alarm rates and taking safety precautions in a timely manner. It can serve as an online decision support tool for plant operators, allowing them to identify the most urgent alarms and handle abnormal situations accordingly.

4.3 Case Study

In this section, the proposed method is evaluated by using the TEP with a closed-loop plant simulator, which is a well-established benchmark [11]. In this process, alarms are generated for measured process variables sampled over a

Table 4.1: Description of five alarm flood categories in TEP

Class index	Label	Fault description	Type
1	c_1	A/C feed ratio, B composition constant (Stream 4)	Step
2	c_2	B composition, A/C feed constant (Stream 4)	Step
3	c_3	Reactor cooling water inlet temperature	Step
4	c_4	C header pressure loss (Stream 4)	Step
5	c_5	Reactor cooling water valve	Sticking
6	New_1	Condenser cooling water inlet temperature	Step
7	New_2	A feed loss (Stream 1)	Step

10-hour simulation period with a 0.6-min sampling interval. This study examines seven types of abnormal situations caused by different faults. As shown in Table 4.1, five abnormal conditions associated with the lables $c_1, c_2, \ldots, c_5$ are considered to be previously seen, while two categories labeled New_1 and New_2 are assumed to be new abnormal situations.

The alarm floods associated with each abnormal condition are detected based on the strategy presented in (4.3) by using the parameters $T = 10$, $\epsilon = 0.75$, and $\gamma = 8$. During the alarm configuration process, alarms are set for 39 measured process variables based on four different thresholds. As a result, four alarm tags are defined for each process variable as "XMEAS_pv_H", "XMEAS_pv_L", "XMEAS_pv_HH", and "XMEAS_pv_LL". In this format "pv" indicates the process variable index, "H" and "HH" represent "High" and "High High" alarms, and "L" and "LL" represent "Low" and "Low Low" alarms, respectively. For example, alarm tag "XMEAS_01_H" is annunciated when process variable number one exceeds its corresponding threshold for the "High" alarm. The list of all process variables and their corresponding thresholds can be found in [86].

As an example of simulated alarm data, the A&E log of an alarm flood belonging to category c_4 is provided in Table 4.2. The list of alarm tags triggered during this alarm flood is presented in the first column. The second

column shows the triggering time of alarm tags, and the last column includes a description for each alarm.

Table 4.2: An alarm flood from c_4

Alarm	Time	Description
XMEAS_20_H	6:03:00	Compressor work
XMEAS_04_L	6:03:00	A and C feed (stream 4)
XMEAS_07_L	6:03:00	Reactor pressure
XMEAS_09_L	6:03:00	Reactor temperature
XMEAS_13_L	6:03:00	Product separator pressure
XMEAS_16_L	6:03:00	Stripper pressure
XMEAS_20_HH	6:03:00	Compressor work
XMEAS_04_LL	6:03:00	A and C feed (stream 4)
XMEAS_07_LL	6:03:00	Reactor pressure
XMEAS_13_LL	6:03:00	Product separator pressure
XMEAS_16_LL	6:03:00	Stripper pressure
XMEAS_18_H	6:03:36	Stripper temperature
XMEAS_21_H	6:03:36	Reactor cooling Water outlet temperature
XMEAS_06_L	6:03:36	Reactor feed rate (stream 6)
XMEAS_21_HH	6:03:36	Reactor cooling Water outlet temperature
XMEAS_09_LL	6:03:36	Reactor temperature
XMEAS_22_H	6:04:12	Separator cooling Water outlet temperature
XMEAS_18_HH	6:04:12	Stripper temperature
XMEAS_10_L	6:05:24	Purge rate (stream 9)
XMEAS_11_H	6:06:36	Product separator temperature
XMEAS_10_LL	6:07:48	Purge rate (stream 9)
XMEAS_04_L	6:09:00	A and C feed (stream 4)
XMEAS_06_L	6:10:48	Reactor feed rate (stream 6)
XMEAS_21_H	6:11:24	Reactor cooling Water outlet temperature
XMEAS_21_HH	6:11:24	Reactor cooling Water outlet temperature
XMEAS_34_H	6:13:12	Component F (stream 9)

Every abnormal condition is associated with a total of 40 alarm floods. 75% of the alarm floods caused by seen categories are used to train the classifier, and 25% of them in addition to 20 alarm floods corresponding to new classes are used to test the online open set classification. The training dataset is built using (4.12) and utilized to train five one-vs-rest classifiers. Figure 4.3 shows the trained sigmoid functions for each class, which are used to generate class

probabilities.

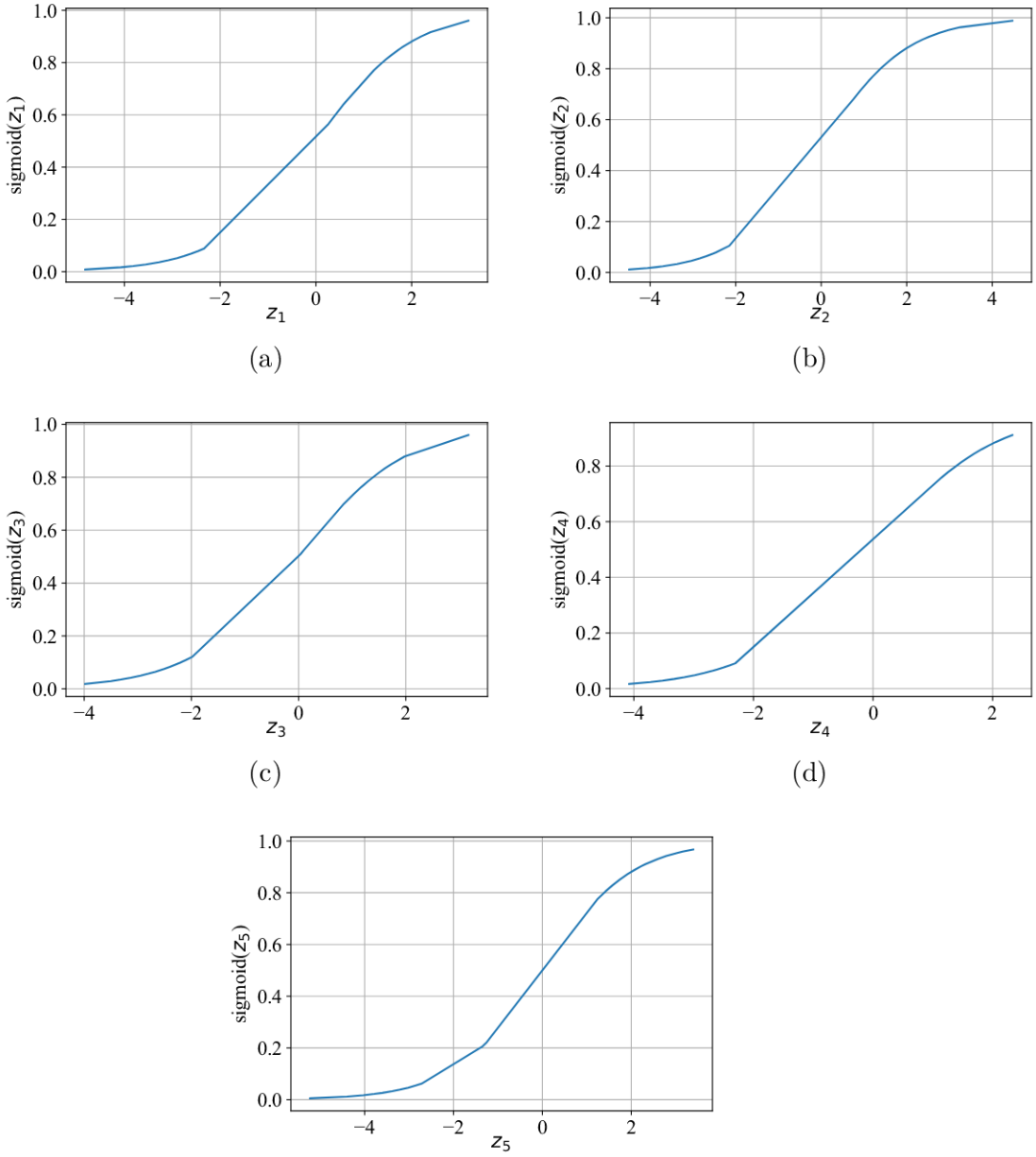

Figure 4.3: Trained sigmoid functions: (a) Class 1. (b) Class 2, (c) Class 3. (d) Class 4. (e) Class 5.

The PDFs generated by class probabilities are depicted in Figure 4.4. As shown in this figure, the open set decision boundaries estimated by (4.15) are higher than the default value.

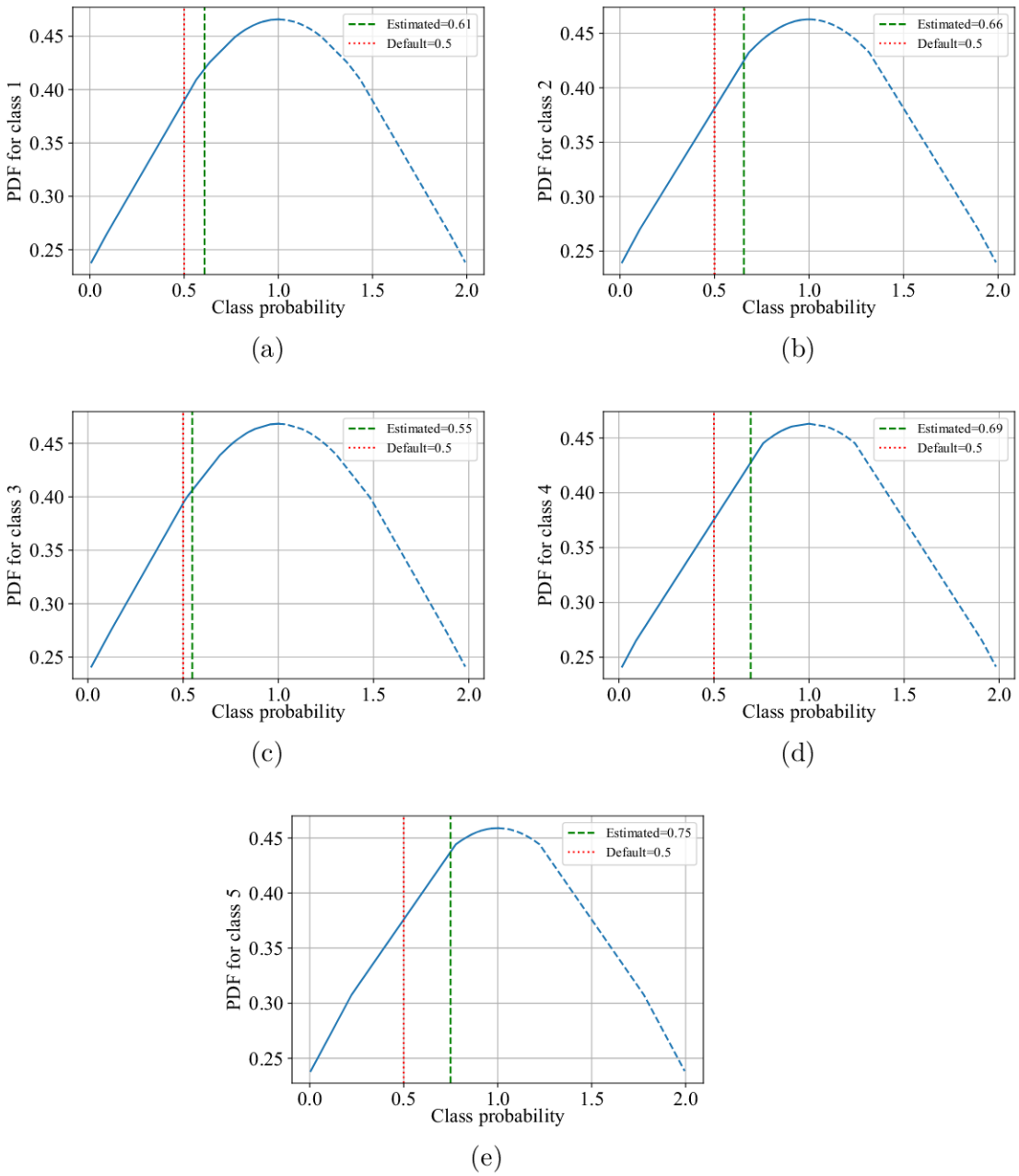

Figure 4.4: PDFs generated for class probabilities and decision boundaries (dotted red vertical lines represent the default thresholds and dashed green vertical lines represent the estimated thresholds): (a) Class 1. (b) Class 2, (c) Class 3. (d) Class 4. (e) Class 5.

After training the proposed classifier, Algorithm 3 is implemented for test data. The open set F-measure proposed in [72] is employed to evaluate the

open set classification performance. This metric measures how well a classifier can classify previously seen classes and reject new ones. It ranges from 0 to 1, with a higher value indicating better open set classification performance. Figure 4.5 shows the average open set F-measures calculated for test data at each time update of the online classification. We can see that the proposed classifier performs well even at the early stages of detecting online alarm floods. Also, the classifier with the estimated decision boundaries outperforms the one that uses the default threshold.

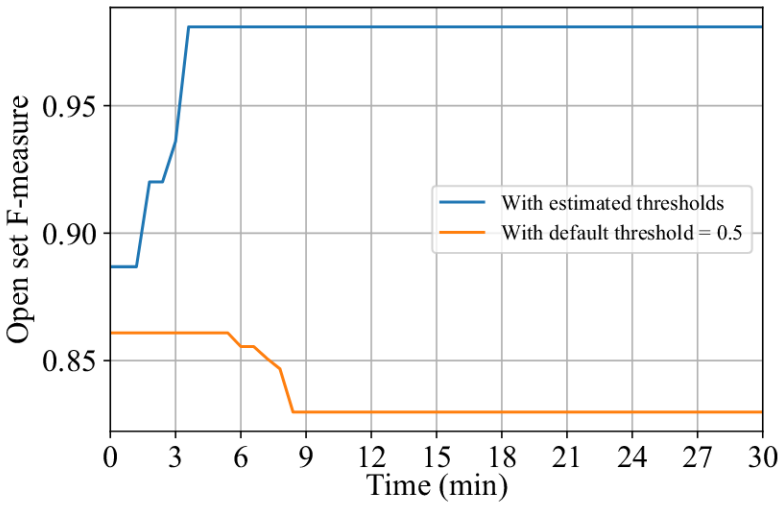

Figure 4.5: Average open set F-measure.

A comparison of the classification results of the proposed method with those of state-of-the-art alarm flood classification methodologies is provided in Figure 4.6. The F-measure is used to evaluate the effectiveness of the proposed method in dealing with previously unseen alarm floods. At each time update, the F-measures are calculated based on the results of classifying test data including both seen and unseen samples. The results confirm the effectiveness of the proposed method as compared to existing online alarm flood classification approaches based on alarm data.

The avarage accuracies for online classification of seen data with the proposed method and the method in [32] are shown in Figure 4.7. It confirms

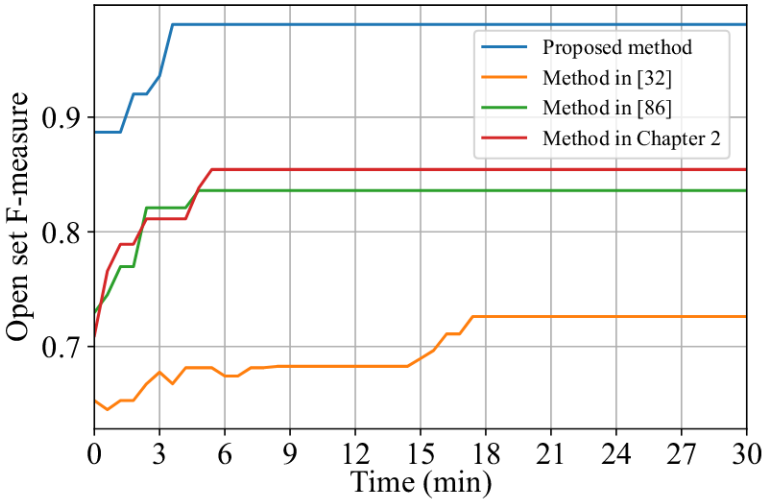

Figure 4.6: Average open set F-measure.

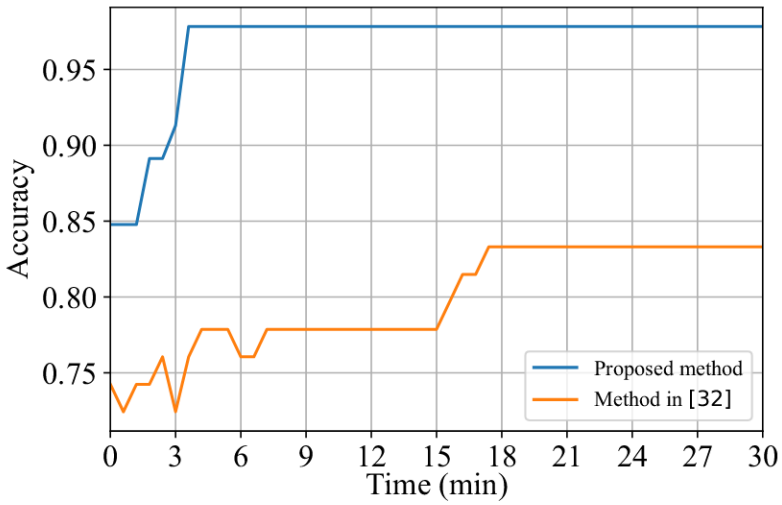

Figure 4.7: Average accuracy for the classification of seen data.

that the proposed weighting strategy effectively reflects the key characteristics of alarm floods in MBoW vectors. In [32], an alarm flood vector was proposed without considering the time information of the alarm sequences. According to the results shown in Figure 4.6, the developed method outperforms the other method at all stages of classification. This indicates the contribution of the term TW to embed the temporal information of alarm floods into the

alarm flood feature vectors.

An alarm flood from the unseen class New_2 is used to conduct an experimental analysis of the proposed weighting strategy. The weights of the top three alarms in different updates of the online alarm flood are listed in Table 4.3. This is to demonstrate the effectiveness of the online alarm ranking offered by the proposed vectorization method. The records in this table correspond to updates when new alarms are triggered in the online alarm flood. Table 4.3 shows that the two top alarm tags in all updates correspond to the process measurement related to A feed, which take values considerably higher than the lower ranked tags. These alarms have been configured as "LL", and "L" for the associated process variable, indicating that the process measurement falls below the alarm limits of "Low-Low" and "Low". Additionally, most updates have reactor pressure and reactor feed rate alarms as the third highest ranked alarms. According to Table 4.1, the fault description corresponding to class New_2 states that this category of alarm floods is caused by A feed loss. In [40], the entire TEP was divided into multiple sub-blocks using the P&IDs and expert knowledge to provide a clear insight into the process. It is evident from the information provided in [40] that the feed and reactor can be combined into one block as they are connected through physical structures and control loops. Thus, it can be verified that at every update, reasonable rankings of alarms are provided, which correspond to abnormal conditions.

4.4 Conclusion

In this chapter, an online operator-assistance mechanism proposed that relies on early classification of alarm floods and alarm ranking. A vectorization model was developed to represent alarm floods as feature vectors considering key alarm characteristics, including the temporal information. A strategy for weighting alarms based on their relevance to the abnormal situation was developed, while taking early classification accuracy into account. A classification method based on decision boundary estimation was proposed to handle the

Table 4.3: An example of online alarm ranking for a new alarm flood

Time	Alarm	Description	Score
Start of the alarm flood	XMEAS_01_LL	A feed	2.26
	XMEAS_01_L	A feed	1.95
	XMEAS_07_LL	Reactor pressure	0.31
1.8 min later	XMEAS_01_LL	A feed	2.10
	XMEAS_01_L	A feed	1.80
	XMEAS_07_LL	Reactor pressure	0.30
3 min later	XMEAS_01_LL	A feed	1.96
	XMEAS_01_L	A feed	1.69
	XMEAS_07_LL	Reactor pressure	0.28
4.2 min later	XMEAS_01_LL	A feed	1.73
	XMEAS_01_L	A feed	1.49
	XMEAS_07_LL	Reactor pressure	0.25
5.4 min later	XMEAS_01_LL	A feed	1.63
	XMEAS_01_L	A feed	1.40
	XMEAS_20_LL	Compressor work	0.36
7.8 min later	XMEAS_01_LL	A feed	1.54
	XMEAS_01_L	A feed	1.33
	XMEAS_06_L	Reactor feed rate	0.38
10.8 min later	XMEAS_01_LL	A feed	1.47
	XMEAS_01_L	A feed	1.26
	XMEAS_06_L	Reactor feed rate	0.55
11.4 min later	XMEAS_01_LL	A feed	1.40
	XMEAS_01_L	A feed	1.20
	XMEAS_06_L	Reactor feed rate	0.52
13.2 min later	XMEAS_01_LL	A feed	1.33
	XMEAS_01_L	A feed	1.15
	XMEAS_06_L	Reactor feed rate	0.50
13.8 min later (End of the alarm flood)	XMEAS_01_LL	A feed	1.33
	XMEAS_01_L	A feed	1.15
	XMEAS_06_L	Reactor feed rate	0.50

new alarm flood scenarios. The benchmark TEP validates the performance of the proposed method.

Chapter 5

A Probabilistic Framework for Online Analysis of Alarm Floods Using Convolutional Neural Networks [*]

In this chapter, a novel alarm flood representation based on a probabilistic framework is proposed, which is capable of tolerating alarm order ambiguities and the effect of irrelevant alarms. The proposed representation incorporates historical alarm tags and their triggering time to define conditional probability distribution functions (CPDF)s for temporal information inherent in alarm floods. Using the resulting CPDFs, time-stamped alarm sequences associated with alarm flood data of different sizes are transformed into fixed order matrices that can be fed into convolutional neural networks (CNNs). An online alarm flood analysis is then carried out through training a CNN, which predicts upcoming fault scenarios by observing online alarm floods. By using a modified training process, online alarm floods with alarm order ambiguity can be classified accurately even at early stages of occurrence. Based on this strategy, online operator assistance mechanisms can be implemented to provide early decision support and ensure safe and efficient plant operations. The

[*]The material in this chapter has been submitted for publication as: Haniyeh Seyed Alinezhad, Jun Shang, Tongwen Chen, and Sirish L. Shah, "A probabilistic framework for online analysis of alarm floods using convolutional neural networks," *IEEE Transactions on Instrumentation and Measurement*.

proposed method is evaluated through case studies using alarm datasets from the well-established TEP benchmark and a Vinyl Acetate Monomer (VAM) process. Comparative studies with state-of-the-art alarm flood analysis methods are also provided to show the effectiveness of the proposed approach.

This chapter is organized as follows. Section 5.1 provides the definition of an alarm flood and a summary of challenges in alarm flood monitoring. In Section 5.2, the proposed ML-based online alarm flood monitoring technique is introduced. Two case studies evaluating the effectiveness of the proposed method are presented in Section 5.3. The chapter is finally concluded in Section 5.4.

5.1 Alarm Floods

Alarm floods occur when plant operators receive too many alarms within a relatively short period of time, making it difficult for them to deal with the data and respond appropriately. An alarm flood can be detected using alarm rate as a metric for defining its start and end. The calculation of the alarm rate can be implemented through a sliding time window, which counts the number of triggered alarms within a predefined time frame. The detection of alarm floods can then be accomplished by setting alarm rate thresholds corresponding to the start and end conditions. For example, according to the industrial standard ANSI/ISA-18.2, an alarm flood occurs when alarms are raised at a rate of at least 10 alarms per 10 minutes and ends when no more than five alarms are raised in a 10-min period [50]. This strategy can be applied both to historical alarm logs and upcoming real-time alarm data to identify alarm floods. A detailed explanation of the application of a sliding time window to online and offline alarm flood detection has been presented in Chapter 4.

In the field of alarm management, the development of alarm flood analysis techniques can contribute to assisting plant operators in addressing alarm flood issues. Analyzing similarity between alarm sequences associated with

alarm floods is a useful way to handle alarm floods. Similar alarm sequences are often indicative of alarm floods caused by the same abnormality. Therefore, data from similar historical alarm floods can provide useful information for operators to manage alarm overloading situations. To conduct an effective similarity analysis, it is imperative to take into account certain aspects of alarm data corresponding to alarm floods.

1. *Alarm Order Ambiguity*: The ambiguity of alarm orders during an alarm flood is one of the critical features to consider. Alarms in similar alarm floods are expected to follow certain patterns revealing a sequential relationship between specific alarm variables. It is however possible that an alarm may appear after a subsequent alarm due to noise and random detection delays. For example, highly correlated alarms that almost happen simultaneously can be subject to uncertainty regarding their order of occurrence.

2. *Similar Alarm Floods with Different Sizes and Alarm Tags*: Due to noise in real plant operations, the number of triggered alarms during similar alarm floods may vary, which results in alarm sequences differing in size. There is also a possibility that similar alarm floods may contain a number of non-identical alarm tags in their alarm sequences, that may not be relevant to their root cause category.

3. *Triggering Time*: An alarm flood is a sequence of time-stamped alarm tags. For similar alarm floods, the triggering time of common alarms may vary, even when alarm sequences have similar chronological orders.

Two alarm floods from the same root cause category that comprise of the aforementioned characteristics are shown in Table 5.1. The data were collected from alarm datasets of the plant simulator for a VAM process [110]. In this example, the red alarm tags indicate alarms that are not shared by two alarm sequences, while the black alarm tags indicate alarms that are the same in both alarm sequences.

Table 5.1: Alarm flood examples

Alarm Flood 1		Alarm Flood 2	
Alarm Tag	Triggering Time	Alarm Tag	Triggering Time
C FI131.PV.Low	17:00:00	FI131.PV.Low	18:33:00
FI132.PV.Low	17:00:00	FI132.PV.Low	18:33:00
FI502.PV.Low	17:00:00	FI502.PV.Low	18:33:00
QI175.PV.High	17:00:00	QI176.PV.High	18:33:00
QI176.PV.High	17:00:00	QI404.PV.Low	18:33:00
QI175.PV.High	17:00:00	QI531.PV.Low	18:33:00
QI401.PV.High	17:00:00	QI561.PV.Low	18:33:00
QI404.PV.Low	17:00:00	EI330.PV.Low	18:33:00
QI531.PV.High	17:00:00	EI560.PV.Low	18:33:00
QI531.PV.Low	17:00:00	QI173.PV.High	18:33:10
QI561.PV.Low	17:00:00	FI131.PV.High	18:33:28
EI330.PV.Low	17:00:00	QI460.PV.High	18:33:30
EI560.PV.Low	17:00:00	PI131.PV.High	18:33:32
QI173.PV.High	17:00:30	FI401.PV.Low	18:33:40
FI401.PV.Low	17:00:32	QI206.PV.High	18:33:42
FI131.PV.High	17:00:37	QI174.PV.Low	18:34:20
PI131.PV.High	17:01:03	QI400.PV.Low	18:34:32
QI206.PV.High	17:01:05	QI412.PV.High	18:34:50
QI174.PV.Low	17:01:30	QI402.PV.High	18:36:02
QI400.PV.Low	17:01:40	QI411.PV.Low	18:36:07

For alarm floods originating from the same cause, investigating the alarm patterns directly without considering the above features can lead to false similarity analysis. The problem arises when two alarm sequences from the same source have small patterns in common, which leads to categorizing them into different alarm flood types. This study addresses the issues listed above to conduct an efficient similarity analysis of alarm floods and to design a reliable operator assistance mechanism.

There is a class of alarms known as chattering alarms, which are characterized by the frequent transition of a single alarm tag between normal and abnormal states [102]. There is a possibility of false alarm flood detection due to chattering alarms causing an increase in the alarm rate. The removal of

chattering alarms is thus necessary before conducting alarm flood detection. In both offline and online applications, several effective techniques have been developed to address the problem of chattering alarms [45, 62]. Since dealing with chattering alarms is outside the scope of this study, the rest of the chapter assumes that chattering alarms have been reduced in the alarm data being examined.

5.2 ML-Based Alarm Flood Monitoring

ML provides methods for analyzing newly generated data by leveraging historical data. An A&E log contains the history of alarms that have been triggered during plant operation. Historically recorded alarms contain valuable information about abnormal circumstances and can be used to provide insights into such situations. Thus, it is possible to develop an ML-based online alarm flood monitoring strategy through the use of historical A&E logs. For ML-based alarm flood analysis, it is necessary to make alarm floods compatible with ML algorithms. Thus, our research primarily focuses on the investigation of a method for transforming alarm floods from time-stamped alarm sequences into inputs suitable for ML algorithms.

5.2.1 A Probabilistic Alarm Flood Matrix

Here a framework is developed to model time-stamped alarm sequences associated with alarm floods as feature matrices that can be applied to a CNN. The proposed alarm flood representation is based on a probabilistic incorporation of temporal information of alarm floods aiming to increase tolerance for alarm order ambiguity across similar alarm floods.

Time data corresponding to historical alarm floods from A&E logs are employed to achieve an alarm flood representation that addresses the three concerns described in the previous section. To accomplish this, temporal datasets including the relative triggering time of alarms in historical alarm floods are

created as follows:

$$T_{ij} = \{t_{a_{ij}} | a_{ij} \in \mathbb{F}_j\}. \tag{5.1}$$

Here the index i indicates the unique alarm tag such that $i \in \{1, 2, \ldots, n\}$; n denotes the total number of unique alarms configured for the plant; j is the index of the alarm flood category, where $j = 1, 2, \ldots, c$ and c denotes the total number of alarm flood categories observed in the historical alarm floods; $\mathbb{F}_j = \{F^{kj}\}_{k=1}^{m_j}$ denotes the set of alarm sequences corresponding to historical alarm floods from the jth category; m_j is the number of alarm flood samples in $\mathbb{F}_j$; a_{ij} represents the ith alarm tag triggered in alarm floods from set $\mathbb{F}_j$; T_{ij} is the set including historical time distances corresponding to the alarm a_{ij}, denoted by $t_{a_{ij}}$, which is defined as

$$t_{a_{ij}} = t^t_{a_{ij}} - t^s_{F^{kj}, a_{ij}}, k \in \{1, \ldots, m_j\} \tag{5.2}$$

where $t^t_{a_{ij}}$ is the triggering time corresponding to a_{ij}, and $t^s_{F^{kj}, a_{ij}}$ is the start time of the kth alarm flood from the jth category, which includes the ith unique alarm tag in its alarm sequence.

To build datasets defined in (5.1), the triggering times of unique alarm tags that satisfy the following assumption are collected:

$$\mathcal{N}_{\mathbb{F}_j}(a_{ij}) \geq \sigma m_j \tag{5.3}$$

where $\mathcal{N}_{\mathbb{F}_j}(a_{ij})$ is the number of alarm floods from the jth historical category that include the ith unique alarm tag in their alarm sequences, and $0 < \sigma \leq 1$.

According to (5.3), alarm tags that are observed in at least a certain number of historical alarm floods of a category are used to define (5.1). This assumption is made to filter out irrelevant or less important alarm tags in each category. Therefore, a value greater than 0.5 would be a reasonable choice for σ to reflect triggering times corresponding to alarm tags activated in the majority of alarm floods. The generated historical time datasets are used to estimate a CPDF for each unique alarm tag activated in each alarm flood category. The CPDF of historical data belonging to the dataset T_{ij}

Algorithm 4: Estimation of CPDFs

Input: Alarm floods detected from historical A&E logs, n unique alarm tags configured for the plant;
Output: Estimated CPDFs
begin
  for $j = 1$ **to** c **do**
    for $i = 1$ **to** n **do**
      if $\mathcal{N}_{\mathbb{F}_j}(a_{ij}) \geq \sigma m_j$ **then**
        for $k = 1$ **to** m_j **do**
          Calculate $t_{a_{ij}}$ by (5.2);
      else
        Ignore the effect of ith unique alarm tag in the jth alarm flood category;
    Build the dataset T_{ij} based on (5.1);
    Estimate the CPDF for T_{ij} by (5.4);

is denoted as $P_{t_{a_i}}(t|C_j)$, which is estimated using the well-established kernel density estimation method as

$$P_{t_{a_i}}(t|C_j) = \frac{1}{n_{ij}h} \sum_{l=1}^{n_{ij}} \mathcal{K}(\frac{t - t_{a_{ij}}}{h}) \tag{5.4}$$

where $\mathcal{K}(\cdot)$ is the kernel function, which is a Gaussian kernel here, and h is the kernel bandwidth that is selected as the optimal choice proposed by [90]. The process of estimating CPDFs from historical temporal data is summarized in Algorithm 4.

The functions estimated based on (5.4) represent the probability distribution of the temporal data corresponding to the ith unique alarm a_i when it is triggered in the jth alarm flood category C_j. The resulting CDFs are employed to define a probabilistic matrix representing an alarm sequence associated with an alarm flood as follows:

$$\mathcal{F} = [f_{ji}]_{c \times n} \tag{5.5}$$

where

$$f_{ji} = \begin{cases} \max P_{t_{a_i}}(t_{a_i}|C_j), & \text{if } a_i \in \mathbb{F}_j \\ 0, \text{ otherwise.} \end{cases} \tag{5.6}$$

The proposed representation can be used to convert alarm sequences of any size into a fixed order matrix, which can be used as input to a CNN. Activated alarm tags and temporal characteristics related to alarm floods can be reflected in this matrix representation. This probabilistic model aims to capture the stochastic nature of temporal information in alarm floods. Using probability-based temporal data rather than exact time stamp values helps to mitigate alarm order uncertainty when performing alarm flood similarity analysis.

5.2.2 CNN-Based Early Classification

Similar alarm sequences are likely to be associated with alarm floods caused by the same abnormality. Online operator assistance for managing alarm floods can be viewed as an ML-based classification method based on similarity analysis between upcoming alarm floods and historical alarm floods. Providing early online assistance for plant operators is crucial to minimize the risk of hazardous incidents and process failures. Therefore we seek to develop a classification process that is capable of providing accurate early predictions for the categories of online alarm floods.

Convolutional Neural Networks

The superior ability of CNNs to deal with complicated problems has led to their successful implementation in many fields. Due to their powerful self-tuning and learning capabilities, CNNs can capture complex features from high-dimensional inputs efficiently. In general, CNNs consist of three basic components, convolutional layer, pooling layer, and fully connected layer.

Convolution is the core operation of a CNN that is carried out through a convolutional layer. In a convolutional layer, a set of learnable kernels are employed to extract local features from input data increasing the generalization capability of the CNN to handle new inputs. The convolved results or produced feature maps are then sent through an activation function to incorporate nonlinearity. As a result, the CNN is able to learn more complex

representations and becomes more expressive. If the flood matrix $\mathcal{F}$ is used as the input to a convolutional layer, the resulting output feature map is defined as follows:

$$\mathcal{F}^{\text{conv}} = g_{\text{conv}}(W * \mathcal{F}^{\text{in}}) \tag{5.7}$$

where $g_{\text{conv}}(\cdot)$ denotes a nonlinear activation function; W represents a $w \times w$ convolutional kernel, and "$*$" indicates the 2-dimensional convolutional operator, which calculates the inner product of the kernel matrix at each location of the input by moving kernel across the input data.

Typically, a pooling layer is placed after a convolutional layer, which applies a downsampling operation to feature maps aiming to reduce the input dimensionality, the computational cost, and the possibility of over-fitting. The pooling operation for a feature map extracted from an alarm flood matrix can be defined as follows:

$$\mathcal{F}^{\text{pool}} = g_{\text{pool}}(\mathcal{F}^{\text{conv}}). \tag{5.8}$$

Here $g_{\text{pool}}(\cdot)$ is the pooling function. The most common pooling strategies in CNNs are average-pooling and max-pooling. This work uses the max-pooling operation to generate feature maps including prominent attributes.

Following feature extraction layers, the resulting feature maps are flattened into a one-dimensional vector for feeding into a fully connected layer as

$$\mathcal{O}_{fc} = g_{fc}(w_{fc}\mathcal{I}fc + b_{fc}). \tag{5.9}$$

Here a weighted sum of the input $\mathcal{I}fc$ is performed, which is then put through an activation function $g_{fc}(\cdot)$. This layer is a key component of the CNN architecture, which helps to conduct classification tasks via providing a probability distribution over different classes based on the output $\mathcal{O}_{fc}$.

A CNN can be constructed by using multiple feature extraction layers with kernels of different sizes and multiple fully connected layers. This enables the extraction of deep features from input data and helps improve classification accuracy. Weights and biases in the fully connected layer and convolutional kernels are learned during the training process of a CNN by processing a

set of historical data. The trained CNN can then be used to perform the classification task on the incoming input data [80].

CNN-Based Online Alarm Flood Monitoring

By using the probabilistic model proposed in (5.5) and (5.6), a training dataset including historical alarm flood matrices can be built, which can be used to train a CNN. A classification-based mechanism for monitoring online floods can then be implemented using the trained CNN.

In this study, training data are modified so that early classification accuracy is enhanced. As a result, each historical alarm flood sequence F^{kj} is broken down into sub-sequences of varying lengths as follows:

$$F_s^{kj} = F^{kj}_{t_s^{kj}:t_d^{kj}+(s-1)\Delta}. \tag{5.10}$$

Here F_s^{kj} is a sub-sequence of F^{kj} including the alarms that have been triggered within the time interval $t_s^{kj} : t_d^{kj} + (s-1)\Delta$; t_s^{kj} is the start time of the alarm flood; t_d^{kj} is the time at which the alarm flood was detected; $s = 1, 2, \ldots, s_{e_{kj}}$, such that for $s = s_{e_{kj}}$, $t_d^{kj} + (s-1)\Delta = t_e^{kj}$, where t_e^{kj} is the end time of the alarm flood; and Δ is a predefined step size to increase the length of generated sub-sequence.

To improve the early classification accuracy, a modified training dataset can be created by converting the alarm sub-sequences corresponding to each historical alarm flood into the proposed probability matrices and labeling them identically. Using this dataset rather than a set of full-length historical alarm flood data allows the classifier to learn different updates of alarm floods from each fault category. As a result, the trained CNN can handle the classification of online alarm floods even at early stages of occurrence more effectively. Algorithm 5 summarizes the proposed data prepossessing that is used to prepare the training data for the CNN.

As a result of the proposed CNN-based strategy, a classification-based mechanism for online analysis of alarm floods is achieved, which is capable

Algorithm 5: Training CNN

Input: Historical alarm floods, estimated CPDFs;
Output: Trained CNN
begin
  for $j = 1$ **to** c **do**
    for $k = 1$ **to** m_j **do**
      while $s \leq s_{e_{kj}}$ **do**
        Extract the sub-sequence F_s^{kj} by using (5.10);
        Convert F_s^{kj} to the matrix form (5.5) using (5.6);
        Label the resulting alarm flood matrix as category j;
        Save labeled data in a training dataset;
  Train the CNN using the built training dataset;

of identifying the fault category for an online alarm sequence. To establish early classification and provide timely decision support for plant operators, it is essential to initiate classification once an ongoing alarm flood is detected. To perform online classification, the CPDFs estimated by (5.4) are assumed to be a set of generative models representing temporal information about historical alarm floods. The probabilistic representation of the temporal data corresponding to newly detected alarm flood samples can be drawn from these generative models. The result is utilized to generate the alarm flood feature matrix for the online alarm flood, which is then used as an input to the trained neural network that predicts the ongoing fault category. Whenever the online alarm flood is updated over time the alarm flood matrix and subsequent classification results should also be updated until the alarm flood ends. By defining the online alarm flood in (5.11), the summary of the proposed alarm flood monitoring methodology can be represented by Algorithm 6.

$$F_u^o = F_{t_s^o : t_d^o + (u-1)\delta}^o \tag{5.11}$$

Here F_u^o indicates the uth update of the online alarm flood, denoted as F^o, which includes the alarms that have been triggered within the time period $t_s^o : t_d^o + (u-1)\delta$; $t_s^o = t_d^o - \mathcal{T}_o$ is the start time of the online alarm flood; t_d^o is the time at which the online alarm flood is detected; δ and $\mathcal{T}_o$ denote

Algorithm 6: CNN-based alarm flood monitoring

Input: Online alarm sequence F^o, estimated CPDFs, and trained CNN;
Output: Classification decision for the ongoing alarm flood
begin
  while $u \leq u_e$ **do**
    Convert F^o to the matrix form (5.5) using (5.6);
    Feed the resulting alarm flood matrix to the CNN trained by Algorithm 5;
    Make a classification decision about ongoing fault category to offer online assistance to plant operators;
    Update F^o;

the step size and the width of the sliding time window used to detect alarm floods during online operation, respectively; and $u = 1, 2, \ldots, u_e$, such that for $u = u_e$, $t_d^o + (u-1)\delta = t_e^o$, where t_e^o is the end time of the online alarm flood. With the proposed probabilistic framework and modified training scheme, we can achieve high accuracy when classifying online alarm floods despite possible ambiguities in alarm orders.

5.3 Case Studies

In this section, an evaluation of the proposed method is presented through case studies utilizing alarm data from the TE process and a VAM process [11, 110]. Moreover, to demonstrate the effectiveness of the developed methodology, comparisons are made with state-of-the-art online alarm flood analysis methods.

5.3.1 TE Process Alarm Data

Here, alarm flood data generated by TE process with a closed-loop plant simulator, a well-established benchmark, are employed to assess the performance of the proposed technique [11]. Alarm data for this process are collected through simulated process variables with a 0.6 min sampling interval. Detailed information regarding all process variables and their configured alarm thresh-

Table 5.2: Description of four alarm flood categories in TE process

Class index	Fault description
1	A/C feed ratio, B composition constant (Stream 4)
2	B composition, A/C feed constant (Stream 4)
3	A feed loss (Stream 1)
4	C header pressure loss (Stream 4)

olds can be found in [86]. This study analyzes alarm floods resulting from four different abnormal situations listed in Table 5.2. A total of 40 alarm floods accompany every abnormal condition, where average duration of alarm floods is 21.6 minutes. We use 75% of the alarm floods as historical data, and the remaining 25% as test data for online alarm flood analysis.

Historical time stamps of alarm tags in each alarm flood category are used to estimate CPDFs by using Algorithm 4, which are shown in Figure 5.1. By using historical alarm flood data, Algorithm 5 prepares the training data, which is then used to train a CNN. Adjusting CNN parameters, such as network structure, is a data-dependent task, which is here achieved based on the common practice adopted by the deep learning community [36]. We use a CNN with three convolutional layers, two pooling layers, and a fully connected layer.

The number of convolution kernels for the three convolution layers is set to 8, 16, and 32, respectively, with 3×3 kernels and the rectified linear unit (ReLU) as the activation function g_{conv}. The function g_{pool} implements max-pooling with a window size of 2×2, and the fully connected layer performs classification through a softmax activation function.

Through the implementation of Algorithm 6 on test data, the classification performance of the trained CNN is evaluated. Figure 5.2 illustrates the average classification accuracy of test data calculated for each online classification update. Considering the average duration of the employed alarm flood data,

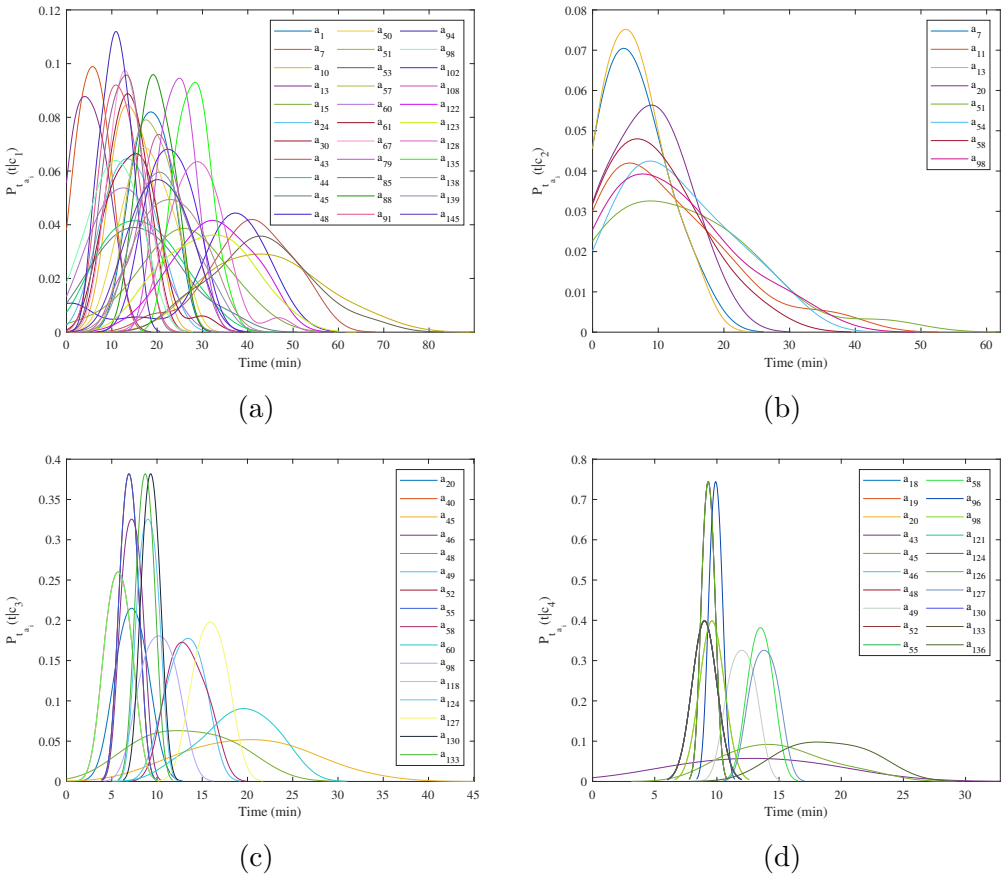

Figure 5.1: Estimated CPDFs for TE process alarm flood data: (a) Class 1. (b) Class 2, (c) Class 3. (d) Class 4.

it is evident that the developed classifier works well even at the initial stages of online alarm floods.

5.3.2 VAM Process Alarm Data

In this part, alarm data from a VAM process is used to conduct a comparative study between the proposed approach and state-of-the-art online alarm flood analysis methods. As compared to the alarm flood data collected through the TE process, alarm floods from the VAM process include a higher degree of alarm order ambiguity. This makes it suitable for evaluating whether the proposed method is effective in handling alarm order ambiguity when compared with existing online methods. A VAM process simulator is used to model dif-

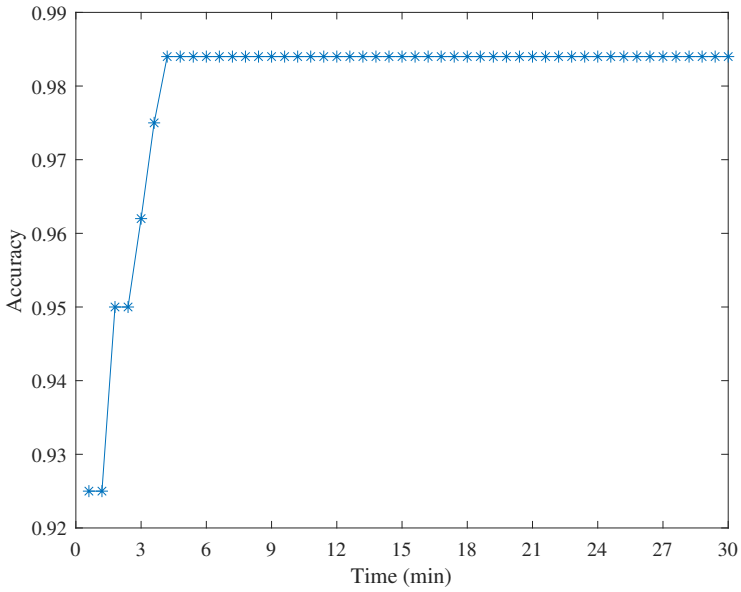

Figure 5.2: Average classification accuracy using TE process alarm flood data.

Table 5.3: Description of four alarm flood categories in VAM process

Class index	Fault description
1	Absorber circulation pump fail
2	Column bottom pump fail
3	Separator bottom valve fail
4	Compressor surging

ferent faulty conditions and generate alarm data for process variables based on approximately 10 seconds sampling intervals [110]. Here we examine alarm floods resulting from four different abnormal conditions listed in Table 5.3. For each abnormal situation, 30 alarm flood samples are generated, and the average duration of alarm floods is about 10.05 minutes. To perform online alarm flood analysis, we use about 75% of the alarm floods as historical data and the remaining 25% as test data.

Algorithm 4 is utilized to estimate the CPDFs, which are shown in Figure 5.3. Consequently, Algorithm 5 is used to prepare the dataset for CNN

training. We use a network structure with three convolution layers, two max-pooling layers with 2×2 pooling windows, and a fully connected layer. The three convolution layers utilize 16, 32, and 64 convolution kernels, respectively, and the classification is carried out with the softmax activation function.

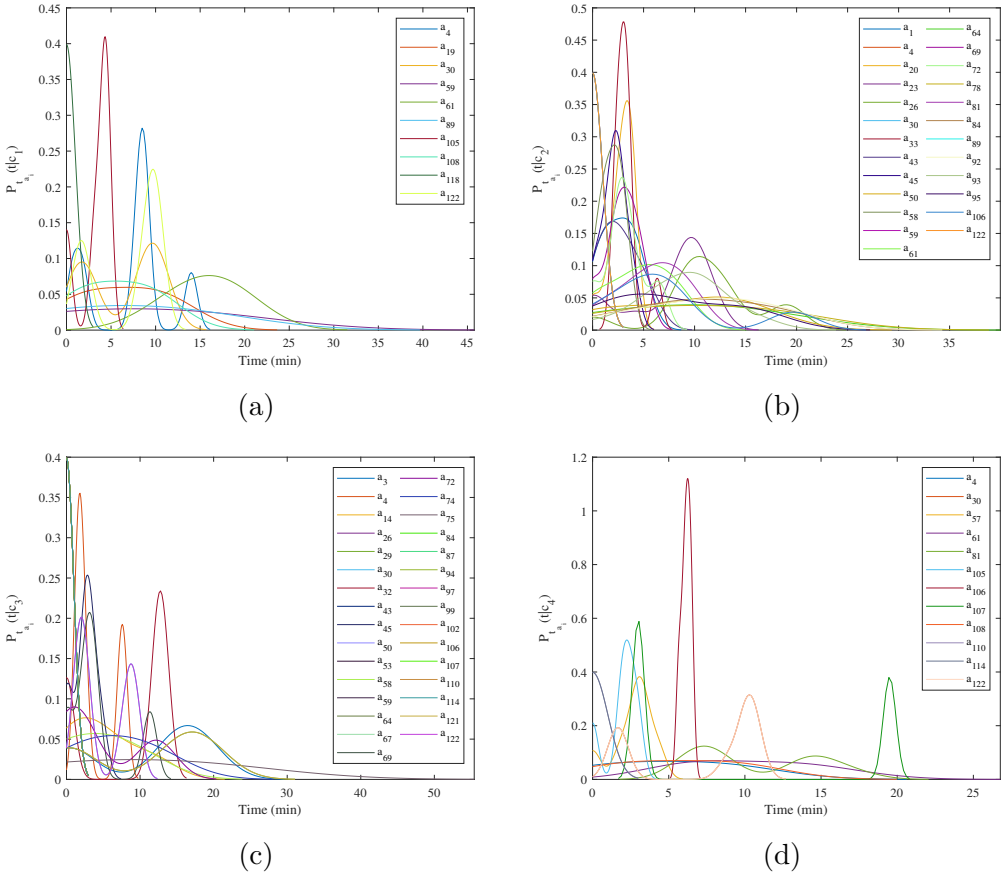

Figure 5.3: Estimated CPDFs for VAM process alarm flood data: (a) Class 1. (b) Class 2, (c) Class 3. (d) Class 4.

Algorithm 6 is implemented to perform online classification on test data. A comparison of the classification results with those of the state-of-the-art online alarm flood classification methodologies is conducted. Average classification accuracy of test data at each time update is used to evaluate the effectiveness of the proposed method. According to the results shown in Figure 5.4, the developed method outperforms the others in all classification updates. This shows how the proposed probabilistic framework contributes to resolving

uncertainty in alarm orders when conducting online similarity analysis.

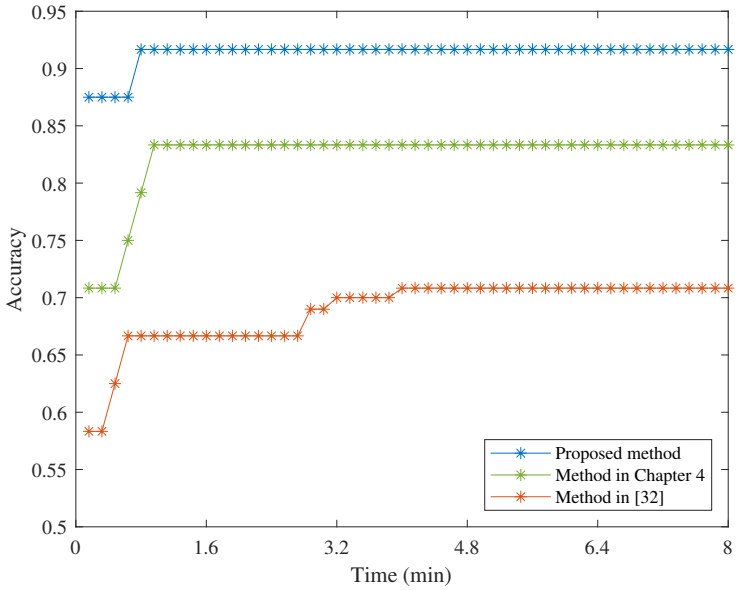

Figure 5.4: Average classification accuracy using VAM process alarm flood data.

5.4 Conclusion

In this chapter, a probabilistic framework based on alarm data was proposed to convert alarm floods into suitable inputs for CNNs. Using historical alarm tags and their triggering times, the proposed framework transformed alarm sequences associated with alarm flood data of different sizes into fixed order probabilistic matrices. We developed ML-based alarm flood analysis using historical alarm flood matrices, in which a CNN was trained to predict upcoming fault categories from online alarm flood data. As a result of an improved training process, online alarm floods could be classified accurately even in their early stages of development. Moreover, this method was capable of tolerating irrelevant alarms and ambiguous alarm orders in the similarity analysis of alarm floods due to its probabilistic nature. We evaluated the effectiveness of the proposed approach using two alarm datasets from the TE process and a VAM process.

Chapter 6

Conclusions and Future Work

This chapter concludes the thesis by providing some remarks on the main findings. It also points out some problems and potential research directions for future work.

6.1 Conclusions

This thesis proposes advanced ML-based methods for online alarm flood monitoring using alarm data, with the ultimate goal of supporting plant operators in analyzing failure root causes. The outcomes of the investigations in this thesis are summarized as follows:

1. First, an EAC analysis was used to model alarm floods as feature vectors by utilizing the time stamps associated with their corresponding alarm sequences. Based on the concept of GMM, an approach incorporating offline labeling and online classification was developed using unlabeled historical alarm floods. In the offline phase, GMM-based data clustering was performed, which was automated in terms of choosing the optimal number of clusters by applying an efficient cluster validity index. Partially labeled historical alarm flood clusters were then employed for early classification of ongoing alarm floods.

2. Second, a novel alarm flood vector representation called MBoW was developed inspired by NLP. This representation could reflect the key

features of time-stamped alarm sequences associated with alarm floods. Alarm flood similarity analysis was carried out by grouping similar vectors through an ML-based clustering technique based on an efficient similarity measurement. A weighting strategy was devised to rank alarms according to their relevance to specific abnormal conditions. This method could provide insights from historical data and facilitate the handling of large datasets.

3. Third, an online operator assistance mechanism was developed through the extension of the MBoW model. Aiming at dealing with previously unseen abnormal situations, an ML-based open set classification technique using systematic similarity threshold estimation was devised. The proposed classifier could reduce the risk of incorrect classifications by rejecting samples with low classification confidence. Plant operators could benefit from this approach to make timely decisions regarding both previously observed and unseen alarm flood scenarios.

4. Finally, a probabilistic framework was developed to address the problem of alarm order ambiguity in alarm floods. A set of CPDFs were estimated for the temporal information associated with historical alarm floods, which were then utilized to convert alarm floods into suitable inputs for CNNs. As a result, time-stamped alarm sequences associated with alarm flood data of different sizes were transformed into fixed order matrices. The proposed alarm flood matrices were used to establish an ML-based online operator assistance mechanism, where a CNN was trained to predict upcoming fault scenarios by observing online alarm floods. By utilizing a modified CNN training process, it was possible to classify online alarm floods accurately even at the early stages of their occurrence.

6.2 Future Work

Aiming at the development of operator assistance mechanisms, the thesis proposed effective methods for managing abnormal conditions based on online alarm flood monitoring. Alarm flood management during online plant operation is a critical and imperative topic of research which deserves further investigation. Some possible future directions for alarm flood monitoring are summarized as follows.

1. **Benefiting from Alarm Data:**

   The use of data-based approaches to manage alarm floods associated with abnormal situations has received considerable attention due to the large amount of data available in modern computerized processes. This class of methods aims to reduce human effort in process monitoring, and ensure safe and efficient plant operation. The majority of existing research in abnormality management and process safety is based on process data. As an alternative to process data, alarm data related to abnormal conditions can provide useful information for the development of operator assistance systems. Alarms are raised only during fault occurrence, but process data are generated regularly by measuring process variables during normal process operation. As a result, alarm data have lower volume than process data, while provide valuable insights into abnormal conditions. This makes alarm data an advantageous candidate for developing online operator assistance mechanisms with lower implementation and computational complexity. There have been several methods for analyzing alarm flood data in offline applications. However, utilizing alarm data to provide operator decision support during online abnormal situations has not been extensively studied in the literature, and further research is still required to fill the existing gap.

2. **Minimizing Human Effort:**

   In this thesis operator assistance mechanisms were proposed on the basis

of similarity analysis of alarm flood data. Data classification strategies using labeled alarm floods were developed to predict fault categories corresponding to online alarm floods. Labeling alarm floods with information regarding process variables that contribute to their root causes is essential to facilitate the identification of possible solutions to handle abnormal situations efficiently. However, expert evaluation and process knowledge are required to categorize alarm flood data by their root causes. It is therefore imperative to develop root cause identification methods to minimize the reliance on expert knowledge and thus save a considerable amount of time. For instance, integrating causality inference strategies reviewed in the thesis with ML-based strategies can improve the online alarm flood monitoring. This can provide further support via incorporating root cause identification capabilities into the online operator assistance mechanism.

3. **Dealing with Situations Involving Multiple Faults:**
   In this thesis, the study is based on the assumption that alarm floods are the result of a single fault. However, this may not always be the case when dealing with real-world industrial applications, and alarm floods may be caused by a combination of multiple faults during a single incident. In the literature, the situation of multiple faults has not been fully addressed. Thus, it is important to take this condition into consideration when investigating alarm flood analysis methods.

Bibliography

[1] L. Abele, M. Anic, T. Gutmann, J. Folmer, M. Kleinsteuber, and B. Vogel-Heuser. Combining knowledge modeling and machine learning for alarm root cause analysis. *IFAC Proceedings Volumes*, 46(9):1843–1848, 2013.

[2] K. Ahmed, I. Izadi, T. Chen, D. Joe, and T. Burton. Similarity analysis of industrial alarm flood data. *IEEE Trans. Autom. Sci. Eng.*, 10(2):452–457, 2013.

[3] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr.*, 12(4):461–486, 2009.

[4] M. T. Amin, F. Khan, and S. Imtiaz. Dynamic availability assessment of safety critical systems using a dynamic Bayesian network. *Reliability Engineering & System Safety*, 178:108–117, 2018.

[5] M. T. Amin, F. Khan, and S. Imtiaz. Fault detection and pathway analysis using a dynamic Bayesian network. *Chemical Engineering Science*, 195:777–790, 2019.

[6] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. PéRez, and I. Perona. An extensive comparative study of cluster validity indices. *Pattern Recognit.*, 46(1):243–256, 2013.

[7] J. Ariamuthu Venkidasalapathy and C. Kravaris. Hidden markov model based approach for diagnosing cause of alarm signals. *AIChE Journal*, 67(10):e17297, 2021.

[8] R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz. Autonomous fault diagnosis and root cause analysis for the processing system using one-class SVM and NN permutation algorithm. *Industrial & Engineering Chemistry Research*, 61(3):1408–1422, 2022.

[9] K. Balzereit, A. Maier, B. Barig, T. Hutschenreuther, and O. Niggemann. Data-driven identification of causal dependencies in cyber-physical production systems. In *ICAART (2)*, pages 592–601, 2019.

[10] L. Barnett, A. B. Barrett, and A. K. Seth. Granger causality and transfer entropy are equivalent for gaussian variables. *Physical review letters*, 103(23):238701, 2009.

[11] A. Bathelt, N. L. Ricker, and M. Jelali. Revision of the Tennessee Eastman process model. *IFAC-PapersOnLine*, 48(8):309–314, 2015.

[12] M. Bauer, J. W. Cox, M. H. Caveness, J. J. Downs, and N. F. Thornhill. Finding the direction of disturbance propagation in a chemical process using transfer entropy. *IEEE Transactions on Control Systems Technology*, 15(1):12–21, 2007.

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

[14] M. Bauer and N. F. Thornhill. A practical method for identifying the propagation path of plant-wide disturbances. *Journal of process control*, 18(7-8):707–719, 2008.

[15] Z. Chai and C. Zhao. Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification. *IEEE Transactions on Industrial Informatics*, 16(1):54–66, 2019.

[16] H. S. Chen, Z. Yan, Y. Yao, T. B. Huang, and Y. S. Wong. Systematic procedure for Granger-causality-based root cause diagnosis of chemical process faults. *Industrial & Engineering Chemistry Research*, 57(29):9500–9512, 2018.

[17] H. S. Chen, Z. Yan, X. Zhang, Y. Liu, and Y. Yao. Root cause diagnosis of process faults using conditional Granger causality analysis and maximum spanning tree. *IFAC-PapersOnLine*, 51(18):381–386, 2018.

[18] Q. Chen, X. Lang, S. Lu, N. ur Rehman, L. Xie, and H. Su. Detection and root cause analysis of multiple plant-wide oscillations using multivariate nonlinear chirp mode decomposition and multivariate Granger causality. *Computers & Chemical Engineering*, 147:107231, 2021.

[19] X. Chen, J. Wang, and J. Zhou. Process monitoring based on multivariate causality analysis and probability inference. *IEEE Access*, 6:6360–6369, 2018.

[20] F. Cheng and J. Zhao. A novel method for real-time alarm root cause analysis. In *Computer Aided Chemical Engineering*, volume 44, pages 2323–2328, 2018.

[21] X. Cheng, W. Hu, W. Cao, and M. Wu. Identification of root cause alarms by detecting correlations and time delays between alarm variables. In *2021 China Automation Congress (CAC)*, pages 4485–4490, 2021.

[22] Y. Cheng, I. Izadi, and T. Chen. Pattern matching of alarm flood sequences by a modified Smith–Waterman algorithm. *Chem. Eng. Res. Des.*, 91(6):1085–1094, 2013.

[23] L. H. Chiang and R. D. Braatz. Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics and intelligent laboratory systems*, 65(2):159–178, 2003.

[24] D. M. Chickering. A transformational characterization of equivalent Bayesian network structures. In *Conference on Uncertainty in Artificial Intelligence*, pages 87–98, 1995.

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

[26] R. S. de Abreu, Y. T. Nunes, L. A. Guedes, and I. Silva. A method for detecting causal relationships between industrial alarm variables using transfer entropy and k2 algorithm. *Journal of Process Control*, 106:142–154, 2021.

[27] P. Duan, F. Yang, T. Chen, and S. L. Shah. Direct causality detection via the transfer entropy approach. *IEEE transactions on control systems technology*, 21(6):2052–2066, 2013.

[28] P. Duan, F. Yang, S. L. Shah, and T. Chen. Transfer zero-entropy and its application for capturing cause and effect relationship between variables. *IEEE Transactions on Control Systems Technology*, 23(3):855–867, 2014.

[29] M. Fahimipirehgalin, I. Weiss, and B. Vogel Heuser. Causal inference in industrial alarm data by timely clustered alarms and transfer entropy. In *2020 European Control Conference (ECC)*, pages 2056–2061, 2020.

[30] H. Fei, W. Chaojun, and F. Shu Kai S. Fault detection and root cause analysis of a batch process via novel nonlinear dissimilarity and comparative Granger causality analysis. *Industrial & Engineering Chemistry Research*, 58(47):21842–21854, 2019.

[31] J. Folmer, F. Schuricht, and B. Vogel-Heuser. Detection of temporal dependencies in alarm time series of industrial plants. *IFAC Proceedings Volumes*, 47(3):1802–1807, 2014.

[32] M. Fullen, P. Schüller, and O. Niggemann. Semi-supervised case-based reasoning approach to alarm flood analysis. In *Machine Learning for Cyber Physical Systems*, pages 53–61, 2020. Springer.

[33] M. Fullen, P. Schüller, and O. Niggemann. Validation of similarity measures for industrial alarm flood analysis. In *IMPROVE-Innovative Modelling Approaches for Production Systems to Raise Validatable Efficiency*, pages 93–109, 2018. Springer Vieweg, Berlin, Heidelberg.

[34] C. Geng, S.-j. Huang, and S. Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3614–3631, 2020.

[35] H. Gharahbagheri, S. Imtiaz, and F. Khan. Root cause diagnosis of process fault using KPCA and Bayesian network. *Industrial & Engineering Chemistry Research*, 56(8):2054–2070, 2017.

[36] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[37] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, 37(3):424–438, 1969.

[38] C. Guo, W. Hu, S. Lai, F. Yang, and T. Chen. An accelerated alignment method for analyzing time sequences of industrial alarm floods. *Journal of Process Control*, 57:102–115, 2017.

[39] C. Guo, F. Yang, and W. Yu. A causality capturing method for diagnosis based on transfer entropy by analyzing trends of time series. *IFAC-PapersOnLine*, 48(21):778–783, 2015.

[40] R. He, G. Chen, S. Sun, C. Dong, and S. Jiang. Attention-based long short-term memory method for alarm root-cause diagnosis in chemical processes. *Industrial & Engineering Chemistry Research*, 59(25):11559–11569, 2020.

[41] Y. L. He, Y. Zhao, Q. X. Zhu, and Y. Xu. Online distributed process monitoring and alarm analysis using novel canonical variate analysis with multicorrelation blocks and enhanced contribution plot. *Industrial & Engineering Chemistry Research*, 59(45):20045–20057, 2020.

[42] J. Hu and Y. Yi. A two-level intelligent alarm management framework for process safety. *Safety science*, 82:432–444, 2016.

[43] J. Hu, L. Zhang, Z. Cai, Y. Wang, and A. Wang. Fault propagation behavior study and root cause reasoning with dynamic Bayesian network based framework. *Process Safety and Environmental Protection*, 97:25–36, 2015.

[44] J. Hu, L. Zhang, A. Wang, and S. Li. Accident prevention by fault propagation analysis and causal fault diagnosis based on Granger causality test. In *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pages 1554–1558, 2017.

[45] W. Hu, T. Chen, and S. L. Shah. Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods. *IEEE Trans. Ind. Electron.*, 65(9):7290–7300, 2018.

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

[47] W. Hu, J. Wang, and T. Chen. A local alignment approach to similarity analysis of industrial alarm flood sequences. *Control Engineering Practice*, 55:13–25, 2016.

[48] W. Hu, J. Wang, T. Chen, and S. L. Shah. Cause-effect analysis of industrial alarm variables using transfer entropies. *Control Engineering Practice*, 64:205–214, 2017.

[49] K. Huang, S. Wu, F. Li, C. Yang, and W. Gui. Fault diagnosis of hydraulic systems based on deep learning model with multirate data samples. *IEEE Transactions on neural networks and learning systems*, 33(11):6789–6801, 2021.

[50] A. ISA. ISA-18.2: Management of alarm systems for the process industries. *International Society of Automation. Durham, NC, USA*, 2009.

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

[52] D. Jurafsky and J. H. Martin. Naive bayes and sentiment classification. *Speech and language processing*, pages 74–91, 2017.

[53] D. Jurafsky and J. H. Martin. Logistic regression. *Speech and Language Processing*, pages 75–92, 2019.

[54] M. A. Khalil, A. Ahmad, T. A. T. Abdullah, and A. Al-shanini. Failure analysis using functional model and Bayesian network. *Chemical Product and Process Modeling*, 11(4):265–272, 2016.

[55] D. Kirchhübel and T. M. Jørgensen. Generating diagnostic Bayesian networks from qualitative causal models. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1239–1242, 2019.

[56] D. Kirchhübel, X. Zhang, M. Lind, and O. Ravn. Identifying causality from alarm observations. In *International Symposium on Future Instrumentation and Control for Nuclear Power Plants*, pages 1–6, 2017.

[57] D. Koks. *Explorations in mathematical physics: the concepts behind an elegant language*. Springer Science & Business Media, 2006.

[58] X. Kong, Z. Yang, J. Luo, H. Li, and X. Yang. Extraction of reduced fault subspace based on kdica and its application in fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 71:1–12, 2022.

[59] C. Kühnert and J. Beyerer. Data-driven methods for the detection of causal structures in process technology. *Machines*, 2(4):255–274, 2014.

[60] P. Kumari, B. Bhadriraju, Q. Wang, and J. S. I. Kwon. A modified Bayesian network to handle cyclic loops in root cause diagnosis of process faults in the chemical process industry. *Journal of Process Control*, 110:84–98, 2022.

[61] S. Lai and T. Chen. A method for pattern mining in multiple alarm flood sequences. *Chem. Eng. Res. Des.*, 117:831–839, 2017.

[62] S. Lai, F. Yang, and T. Chen. Online pattern matching and prediction of incoming alarm floods. *J. Process Control*, 56:69–78, 2017.

[63] K. Leahy, C. Gallagher, P. O'Donovan, K. Bruton, and D. T. O'Sullivan. A robust prescriptive framework and performance metric for diagnosing and predicting wind turbine faults based on scada and alarms data with case study. *Energies*, 11(7):1738, 2018.

[64] D. Leung and J. Romagnoli. Dynamic probabilistic model-based expert system for fault diagnosis. *Computers & Chemical Engineering*, 24(11):2473–2492, 2000.

[65] G. Li, S. J. Qin, and T. Yuan. Data-driven root cause diagnosis of faults in process industries. *Chemometrics and Intelligent Laboratory Systems*, 159:1–11, 2016.

[66] C. Liu, K. G. Lore, Z. Jiang, and S. Sarkar. Root-cause analysis for time-series anomalies via spatiotemporal graphical modeling in distributed complex systems. *Knowledge-Based Systems*, 211:106527, 2021.

[67] Y. Liu and Z. Ge. Weighted random forests for fault classification in industrial processes with hierarchical clustering model selection. *Journal of Process Control*, 64:62–70, 2018.

[68] M. Lucke, M. Chioua, C. Grimholt, M. Hollender, and N. F. Thornhill. Advances in alarm data analysis with a practical application to online alarm flood classification. *J. Process Control*, 79:56–71, 2019.

[69] Y. Luo, B. Gopaluni, Y. Xu, L. Cao, and Q. X. Zhu. A novel approach to alarm causality analysis using active dynamic transfer entropy. *Industrial & Engineering Chemistry Research*, 59(18):8661–8673, 2020.

[70] X. Ma, Y. Si, Z. Yuan, Y. Qin, and Y. Wang. Multistep dynamic slow feature analysis for industrial process monitoring. *IEEE Transactions on Instrumentation and Measurement*, 69(12):9535–9548, 2020.

[71] G. J. McLachlan, S. X. Lee, and S. I. Rathnayake. Finite mixture models. *Annu. Rev. Stat. Appl.*, 6:355–378, 2019.

[72] P. R. Mendes Júnior, R. M. De Souza, R. d. O. Werneck, B. V. Stein, D. V. Pazinato, W. R. de Almeida, O. A. Penatti, R. d. S. Torres, and A. Rocha. Nearest neighbors distance ratio open-set classifier. *Machine Learning*, 106(3):359–386, 2017.

[73] Q. Q. Meng, Q. X. Zhu, H. H. Gao, Y. L. He, and Y. Xu. A novel scoring function based on family transfer entropy for Bayesian networks learning and its application to industrial alarm systems. *Journal of Process Control*, 76:122–132, 2019.

[74] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[75] T. Niyazmand and I. Izadi. Pattern mining in alarm flood sequences using a modified Prefixspan algorithm. *ISA Trans.*, 90:287–293, 2019.

[76] A. Noroozifar and I. Izadi. Root cause analysis of process faults using alarm data. In *Iranian Conference on Electrical Engineering*, pages 1118–1122, 2019.

[77] M. R. Parvez, W. Hu, and T. Chen. Real-time pattern matching and ranking for early prediction of industrial alarm floods. *Control Engineering Practice*, 120:105004, 2022.

[78] H. Pyun, K. Kim, D. Ha, C. J. Lee, and W. B. Lee. Root causality analysis at early abnormal stage using principal component analysis and multivariate Granger causality. *Process Safety and Environmental Protection*, 135:113–125, 2020.

[79] B. Rashidi, D. S. Singh, and Q. Zhao. Data-driven root-cause fault diagnosis for multivariate non-linear processes. *Control Engineering Practice*, 70:134–147, 2018.

[80] W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[81] M. H. Roohi, P. Ramazi, and T. Chen. Towards accurate root-alarm identification: The causal Bayesian network approach. In *International Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 169–174, 2021.

[82] M. Schleburg, L. Christiansen, N. F. Thornhill, and A. Fay. A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system. *Journal of process control*, 23(6):839–851, 2013.

[83] T. Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461–464, 2000.

[84] M. Scutari. Bayesian network constraint-based structure learning algorithms: Parallel and optimized implementations in the bnlearn R package. *Journal of Statistical Software*, 77(2):1–20, 2017.

[85] M. Scutari, C. E. Graafland, and J. M. Gutiérrez. Who learns better Bayesian network structures: Constraint-based, score-based or hybrid algorithms? In *International Conference on Probabilistic Graphical Models*, pages 416–427, 2018. PMLR.

[86] J. Shang and T. Chen. Early classification of alarm floods via exponentially attenuated component analysis. *IEEE Trans. Ind. Electron.*, 67(10):8702–8712, 2020.

[87] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[88] H. M. Shao, J. G. Wang, and Y. Yao. A copula-based Granger causality analysis method for root cause diagnosis of plant-wide oscillation. In *2020 International Conference on Image, Video Processing and Artificial Intelligence*, volume 11584, pages 1158426, 2020.

[89] Y. Shu and J. Zhao. Data-driven causal inference based on a modified transfer entropy. *Computers & Chemical Engineering*, 57:173–180, 2013.

[90] B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[91] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman. *Causation, prediction, and search*. MIT press, 2000.

[92] J. Su, D. Wang, Y. Zhang, F. Yang, Y. Zhao, and X. Pang. Capturing causality for fault diagnosis based on multi-valued alarm series using transfer entropy. *Entropy*, 19(12):663, 2017.

[93] J. Taheri-Kalani, G. Latif-Shabgahi, and M. A. Shooredeli. On the use of penalty approach for design and analysis of univariate alarm systems. *J. Process Control*, 69:103–113, 2018.

[94] J. Thambirajah, L. Benabbas, M. Bauer, and N. 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.

[95] N. F. Thornhill. Finding the source of nonlinearity in a process with plant-wide oscillation. *IEEE Transactions on Control Systems Technology*, 13(3):434–443, 2005.

[96] C. Tian and C. Zhao. Single model-based analysis of relative causal changes for root-cause diagnosis in complex industrial processes. *Industrial & Engineering Chemistry Research*, 60(34):12602–12613, 2021.

[97] T. Verma and J. Pearl. *Equivalence and synthesis of causal models*. UCLA, Computer Science Department, 1991.

[98] Y. Wan, F. Yang, N. Lv, H. Xu, H. Ye, W. Li, P. Xu, L. Song, and A. K. Usadi. Statistical root cause analysis of novel faults based on digraph models. *Chemical Engineering Research and Design*, 91(1):87–99, 2013.

[99] J. Wang and K. Chen. Selection of root-cause process variables based on qualitative trends in historical data samples. *IEEE Access*, 7:138637–138644, 2019.

[100] J. Wang and T. Chen. An online method to remove chattering and repeating alarms based on alarm durations and intervals. *Comput. Chem. Eng.*, 67:43–52, 2014.

[101] J. Wang, H. Li, J. Huang, and C. Su. Association rules mining based analysis of consequential alarm sequences in chemical processes. *Journal of Loss Prevention in the Process Industries*, 41:178–185, 2016.

[102] 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, 2015.

[103] J. Wang, Z. Yang, J. Su, Y. Zhao, S. Gao, X. Pang, and D. Zhou. Root-cause analysis of occurring alarms in thermal power plants based on Bayesian networks. *International Journal of Electrical Power & Energy Systems*, 103:67–74, 2018.

[104] Y. Wang, H. Yang, X. Yuan, Y. A. Shardt, C. Yang, and W. Gui. Deep learning for fault-relevant feature extraction and fault classification with stacked supervised auto-encoder. *Journal of Process Control*, 92:79–89, 2020.

[105] Z. Wang, X. Bai, J. Wang, and Z. Yang. Indexing and designing deadbands for industrial alarm signals. *IEEE Trans. Ind. Electron.*, 66(10):8093–8103, 2018.

[106] G. Weidl, A. L. Madsen, and S. Israelson. Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes. *Computers & chemical engineering*, 29(9):1996–2009, 2005.

[107] P. Wunderlich and N. Hranisavljevic. Comparison of different probabilistic graphical models as causal models in alarm flood reduction. In *International Conference on Industrial Informatics (INDIN)*, volume 1, pages 1285–1290, 2019.

[108] P. Wunderlich and O. Niggemann. Structure learning methods for Bayesian networks to reduce alarm floods by identifying the root cause. In *IEEE International Conference on Emerging Technologies and Factory Automation*, pages 1–8, 2017.

[109] P. Wunderlich and O.Niggemann. Inference methods for detecting the root cause of alarm floods in causal models. In *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*, pages 893–898, 2018.

[110] G. Yang, W. Hu, W. Cao, and M. Wu. Simulating industrial alarm systems by extending the public model of a vinyl acetate monomer process. In *2020 39th Chinese Control Conference (CCC)*, pages 6093–6098. IEEE, 2020.

[111] J. Yu and M. M. Rashid. A novel dynamic Bayesian network-based networked process monitoring approach for fault detection, propagation identification, and root cause diagnosis. *AIChE Journal*, 59(7):2348–2365, 2013.

[112] W. Yu and F. Yang. Detection of causality between process variables based on industrial alarm data using transfer entropy. *Entropy*, 17(8):5868–5887, 2015.

[113] T. Yuan and S. J. Qin. Root cause diagnosis of plant-wide oscillations using Granger causality. *Journal of Process Control*, 24(2):450–459, 2014.

[114] H. Zhang, C. Jia, and M. Chen. Remaining useful life prediction for degradation processes with dependent and nonstationary increments. *IEEE Transactions on Instrumentation and Measurement*, 70:1–12, 2021.

[115] H. Zhang, J. Shang, J. Zhang, and C. Yang. Nonstationary process monitoring for blast furnaces based on consistent trend feature analysis. *IEEE Transactions on Control Systems Technology*, 30(3):1257–1267, 2021.

[116] X. Zhang, W. Hu, and F. Yang. Detection of cause-effect relations based on information granulation and transfer entropy. *Entropy*, 24(2):212, 2022.

[117] Z. Zhang and F. Dong. Fault detection and diagnosis for missing data systems with a three time-slice dynamic Bayesian network approach. *Chemometrics and Intelligent Laboratory Systems*, 138:30–40, 2014.

[118] B. Zhou, W. Hu, and T. Chen. Pattern extraction from industrial alarm flood sequences by a modified clofast algorithm. *IEEE Transactions on Industrial Informatics*, 18(1):288–296, 2021.

[119] Q. X. Zhu, W. J. Ding, and Y. L. He. Novel multimodule Bayesian network with cyclic structures for root cause analysis: Application to complex chemical processes. *Industrial & Engineering Chemistry Research*, 59(28):12812–12821, 2020.

[120] Q. X. Zhu, Y. Luo, and Y. L. He. Novel distributed alarm visual analysis using multicorrelation block-based PLS and its application to online root cause analysis. *Industrial & Engineering Chemistry Research*, 58(45):20655–20666, 2019.

[121] Q. X. Zhu, Y. Luo, and Y. L. He. Novel multiblock transfer entropy based Bayesian network and its application to root cause analysis. *Industrial & Engineering Chemistry Research*, 58(12):4936–4945, 2019.