Active Alarm Flood Clustering with Deep Reinforcement Learning

Alireza Habibi

Supervisor: Prof. Iman Izadi

Advisor: Dr. Mohammad Hossein Roohi

Department of Electrical and Computer Engineering
Isfahan University of Technology

This dissertation is submitted for the degree of
Bachelor of Science

November 2024

I would like to dedicate this thesis to my loving parents ...

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Alireza Habibi
November 2024

Abstract

Alarm flood clustering is an essential component of efficient alarm system operation in industries. This thesis presents a novel framework that combines reinforcement learning techniques with alarm flood similarity analysis to address challenges in alarm flood clustering and management.

The proposed framework leverages deep reinforcement learning to train a meta-policy for query selection, optimizing the accuracy of alarm pattern mining and root cause analysis. The trained meta-policy guides the alarm flood similarity analysis process, improving the efficiency and effectiveness of alarm flood clustering. The proposed approach demonstrates superior performance compared to existing alarm flood similarity analysis methods, as illustrated through experiments conducted on an actual dataset gathered from a gas processing plant.

The results showcase the framework's ability to improve the accuracy and robustness of alarm flood clustering while minimizing the impact of irrelevant alarms and the ambiguity of alarm order. By integrating reinforcement learning techniques with alarm flood similarity analysis, the proposed framework offers a promising solution for enhancing alarm system operation and management in industrial processes.

Table of contents

Chapter 1

Introduction

1.1 Industrial Processes

Industrial processes are at the core of modern economies, driving economic growth, technological advancements, and the production of goods and materials on a large scale. These processes encompass a wide range of activities and sectors, including manufacturing, chemical processing, energy production, refining and extraction, and food processing. They involve the transformation and refinement of raw materials into finished products, the generation of energy, and the extraction and purification of resources.

The significance of industrial processes cannot be overstated. They are the foundation of manufacturing industries, enabling the production of goods that meet consumer demands. Industrial processes contribute to job creation, export revenues, and technological innovation. Moreover, they play a vital role in infrastructure development, transportation systems, and overall societal progress.

Industrial processes consist of various components that work together to achieve efficient and reliable operations. These components include equipment and machinery, control systems, safety measures, and monitoring and data analysis. Specialized equipment and machinery are used to carry out specific operations within industrial processes, such as assembly lines in manufacturing or distillation columns in chemical processing.

Control systems are crucial in monitoring and regulating industrial processes. They ensure that operations are conducted within desired parameters, maintain product quality, and optimize resource utilization. Control systems can range from manual or semi-automated to fully automated, depending on the complexity and requirements of the process.

Safety measures are an integral part of industrial processes. These measures are implemented to mitigate risks associated with equipment malfunction, chemical hazards, and worker safety. Proper training, safety protocols, and protective equipment are essential in creating a safe working environment.

Monitoring and data analysis are essential for assessing the performance of industrial processes. Real-time monitoring of process variables, data collection, trend analysis, and anomaly detection help in identifying deviations from normal operation and optimizing process efficiency.

Despite the numerous benefits of industrial processes, they are not without challenges. Complexity is a common challenge, with industrial processes often involving multiple steps, variables, and interactions. Managing and optimizing this complexity requires advanced control strategies, sophisticated modeling techniques, and process analysis expertise.

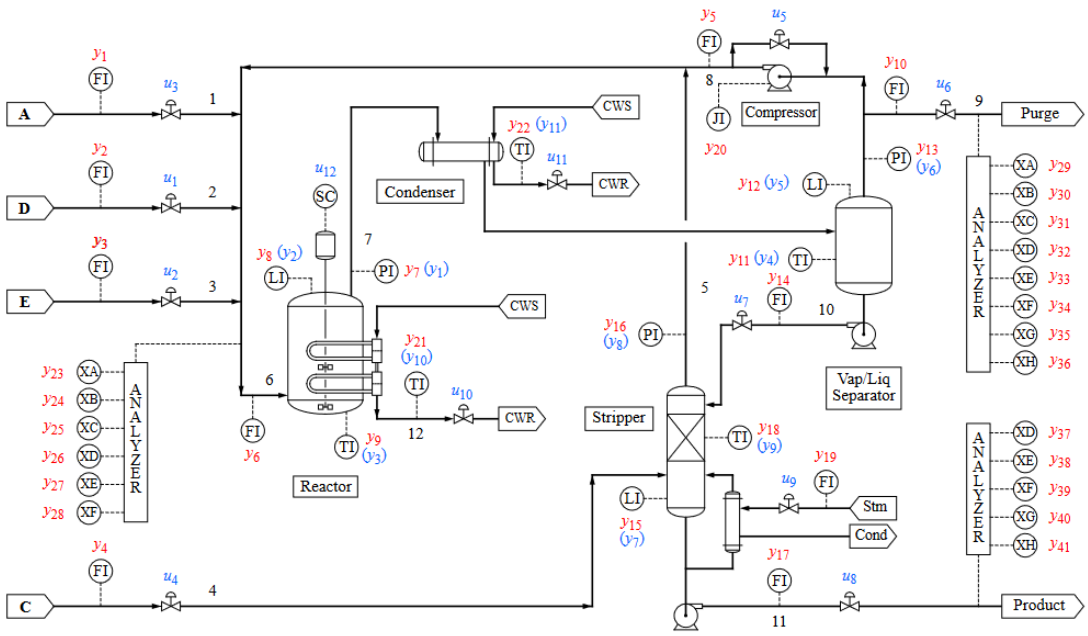

Fig. 1.1 Tennessee Eastman process flow-sheet

Efficiency and productivity are ongoing challenges in industrial processes. Factors such as equipment downtime, suboptimal operating conditions, and inefficient resource utilization can hinder performance and impact profitability. Continuous improvement efforts, process optimization, and advanced analytics can help address these challenges.

Industrial processes also have environmental impacts, including emissions, waste generation, and resource depletion. Sustainable practices, such as waste reduction, energy efficiency, and the use of renewable resources, are increasingly important in mitigating these impacts and promoting environmental stewardship.

Safety and risk management are critical aspects of industrial processes. Proactive risk assessment, adherence to regulations, and the implementation of safety protocols are essential in preventing accidents, protecting workers, and minimizing environmental risks.

Automation and digitalization have played a transformative role in industrial processes. Advancements in technology, such as the Internet of Things (IoT), artificial intelligence (AI), and data analytics, have led to increased efficiency, productivity, and safety. Automation allows for the integration of smart sensors, real-time monitoring, and autonomous control, enabling faster decision-making and optimization.

For instance, a widely studied industrial process in academia is the Tennessee Eastman process. (Downs and Vogel, 1993) introduced the Tennessee Eastman challenge problem at an AIChE meeting in 1990. The purpose was to supply the academics with a problem that contained many of the challenges that people in industry meet. There are eight components, including an inert (B) and a byproduct (F). The process has four feed streams (of A, D, E and A+C), one product stream (a mix of G and H) and one purge stream. The inert (B) enters in the A+C feedstream. The process has five major units; a reactor, a product condenser, a vapor-liquid separator, a recycle compressor and a product stripper, see Figure 1.1.

In conclusion, industrial processes are the backbone of modern economies. They encompass diverse activities and sectors and involve the production, transformation, and refinement of goods and materials. Efficient and reliable industrial processes are crucial for economic growth, technological advancements, and societal

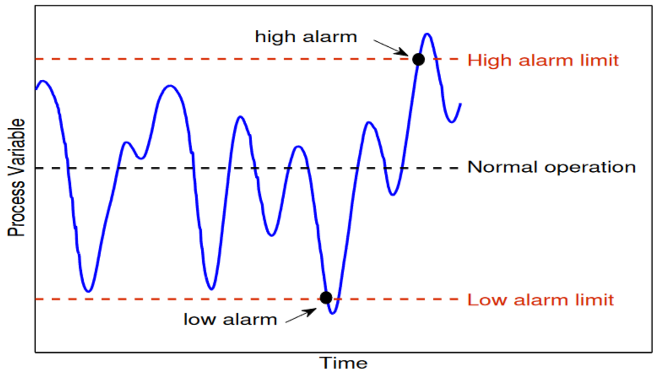

Fig. 1.2 Alarm limits on a variable [12]

progress. Overcoming challenges such as complexity, efficiency, sustainability, and safety requires continuous improvement efforts, adoption of advanced technologies, and adherence to best practices. Automation and digitalization continue to drive innovation in industrial processes, leading to increased efficiency, productivity, and sustainability in the global market.

1.2 Alarms and Alarm Systems

Alarms play a crucial role in various industries, serving as an essential tool for monitoring and alerting operators about faults, abnormalities, and critical conditions within a system or process. In industrial settings, such as manufacturing plants, chemical facilities, power plants, and transportation networks, alarms are designed to provide timely information to operators, enabling them to take necessary actions to ensure the safety, efficiency, and smooth operation of the system.

An alarm can be defined as a signal or notification that indicates a deviation from normal operating conditions or a potential hazard. It serves as a warning system, drawing the attention of operators to specific events or conditions that require their immediate attention or response. Typically, alarms are triggered when certain thresholds or predefined limits are exceeded, such as temperature, pressure, flow rate, or other parameters. These limits are determined based on the equipment's safety boundaries, product quality requirements, and safety regulations. In the traditional sense, an alarm is typically defined based on a single variable exceeding a predetermined "normal" or "safe" range of operation (see Figure 1.2).

The implementation of alarm systems has evolved significantly with advancements in technology. In the past, alarm systems were often simple, relying on basic visual or auditory signals, such as flashing lights, sirens, or bells. However, with the advent of distributed control systems (DCS) and advanced automation technologies, modern alarm systems have become more sophisticated and capable of providing detailed information to operators.

Today, alarm systems are commonly integrated into larger control and monitoring systems, such as supervisory control and data acquisition (SCADA) systems or programmable logic controllers (PLCs). These systems enable the collection, processing, and analysis of data from various sensors and actuators distributed throughout the industrial process. The alarms generated by these systems are typically displayed on operator interfaces, such as human-machine interfaces (HMIs), control panels, or computer screens, allowing operators to easily identify and respond to critical events.

The primary objective of alarm systems is to ensure the timely detection of abnormal conditions and facilitate appropriate operator response. By promptly notifying operators about potential issues, alarms enable them to take corrective actions, mitigate risks, and prevent accidents or equipment failures. However, the effectiveness of alarm systems depends on various factors, including the proper design, implementation, and maintenance of the system.

One significant challenge in alarm systems is the management of alarm floods or alarm showers. Alarm floods occur when operators receive an overwhelming number of alarms within a short period, making it difficult for them to prioritize and respond to each alarm effectively. This can lead to alarm fatigue, where operators become desensitized to alarms or overlook critical alarms amidst the flood of notifications. Alarm floods can result from various factors, such as equipment malfunctions, process upsets, or software glitches.

To address the issue of alarm floods, researchers and industry practitioners have developed methodologies and techniques for analyzing and classifying alarm patterns. By studying historical alarm data, it is possible to identify recurring alarm floods, detect patterns of alarm occurrences, and group similar alarms together. This allows operators to gain insights into the root causes of abnormal conditions and prioritize their responses during future alarm floods.

In conclusion, alarms and alarm systems are vital components of industrial processes, serving as early warning systems to ensure the safe and efficient operation of complex systems. Through continuous advancements in technology and data analysis techniques, alarm systems aim to provide operators with timely and relevant information, enabling them to make informed decisions and take appropriate actions in response to critical events. The effective management of alarm floods and the analysis of alarm patterns are essential for enhancing the reliability and usability of alarm systems, ultimately improving overall system safety and performance.

1.3 Alarm Floods

Alarm floods are a critical issue faced by process industries, characterized by a high volume of alarms being triggered within a short period of time. These floods can overload operators, rendering them ineffective in taking necessary actions and potentially leading to emergency shutdowns or major operational disruptions. Understanding and effectively managing alarm floods is of utmost importance to ensure the safety, efficiency, and productivity of industrial plants.

In modern industrial settings, numerous sensors and actuators are deployed, communicating with control systems and operators to monitor and regulate various process variables. Alarms are designed to alert operators about faults, abnormalities, or deviations from desired operating conditions. However, due to inadequate design, implementation, and maintenance of alarm systems, a significant number of false or nuisance alarms can be generated. This results in operators being inundated with a large number of alarms during critical situations, making it challenging for them to prioritize and respond appropriately.

An alarm flood occurs when the rate of alarm generation surpasses the operator's capacity to handle them effectively. This flood indicator is often defined as a high frequency of alarms, such as more than ten alarms per ten minutes. During an alarm flood, operators may face difficulties in distinguishing critical alarms from non-critical ones, leading to delayed responses or missed alarms that could potentially escalate into hazardous situations.

The consequences of alarm floods can be severe, ranging from production losses and decreased operational efficiency to safety incidents and environmental hazards. Alarm floods have been identified as a significant contributing factor to accidents and incidents in process industries. In fact, the Abnormal Situation Management

(ASM) Consortium reports that petrochemical plants experience major accidents, on average, every three years, many of which can be attributed to alarm system failures.

Recognizing the challenges posed by alarm floods, researchers and industry experts emphasize the importance of analyzing and classifying alarm flood patterns. By examining historical alarm data and identifying similarities among alarm flood occurrences, it becomes possible to group and analyze them based on shared patterns or characteristics. This analysis provides valuable insights into the dynamics of alarm floods, the interrelationships between alarms, and the underlying causes of abnormal situations.

Efforts are being made to develop methodologies and techniques for effective alarm flood analysis. These approaches aim to enable operators to identify the root causes of alarm floods based on past incidents, allowing for early detection and proactive response. By understanding the patterns and dynamics of alarm floods, operators can take timely and appropriate actions to prevent or mitigate the consequences before they escalate.

Alarm floods present significant challenges in process industries, impacting the ability of operators to effectively manage critical situations. Analyzing and understanding alarm flood patterns is crucial for identifying root causes, mitigating risks, and enhancing the efficiency and safety of industrial operations. Addressing the complexities of alarm floods requires continuous research and the development of effective methods to support operators in their decision-making processes. By improving alarm system design, implementation, and maintenance practices, industries can minimize the occurrence and impact of alarm floods, ultimately ensuring safer and more efficient operations.

1.4 Alarm Flood Detection

Alarm flood detection is a critical aspect of alarm management systems in industrial processes. The rapid accumulation of alarms can overwhelm operators, leading to reduced situational awareness and increased response time, which can have severe consequences for system safety, reliability, and productivity. Consequently, researchers have dedicated significant efforts to develop effective techniques and approaches for alarm flood detection to mitigate these challenges.

Pattern recognition techniques have emerged as fundamental tools in detecting and analyzing alarm floods. These techniques leverage machine learning algorithms to identify abnormal alarm patterns and differentiate them from normal operating conditions. Artificial neural networks, support vector machines, and decision trees are commonly employed in this context. These algorithms can learn from historical alarm data and extract patterns that are indicative of alarm floods. By training models on a diverse set of alarm patterns, these techniques can provide accurate predictions and enable early detection of alarm floods, allowing operators to proactively manage and mitigate potential disruptions.

In addition to pattern recognition, similarity analysis methods have proven valuable in alarm flood detection. These methods focus on identifying recurring patterns and similarities among alarms, which can be indicative of an impending alarm flood. By considering the temporal and contextual relationships between alarms, these techniques can reveal hidden correlations and provide insights into the underlying causes of alarm floods. Analyzing the similarities between alarms enables the development of proactive strategies to prevent alarm floods and improve the overall efficiency of alarm management systems. Moreover, similarity analysis facilitates the identification of false alarms and redundant alarms, reducing unnecessary operator interventions and enhancing the overall reliability of the alarm system.

Several studies have also explored the integration of data-driven approaches with domain knowledge to enhance alarm flood detection. By incorporating expert knowledge and domain-specific rules into the analysis, researchers aim to improve the accuracy and interpretability of alarm flood detection models. This hybrid approach combines the strengths of data-driven techniques, which can capture complex patterns and relationships in large datasets, with the insights and expertise of domain specialists. The integration of domain knowledge helps in addressing the challenges posed by noisy data, complex alarm scenarios, and the need for explainable decision-making in alarm flood detection.

Consider an alarm system with the set of alarm variables $A = \{\alpha_i,\ i = 1, 2, \cdots, |A|\}$, where $|.|$ denotes the cardinality of the set. In this set, each alarm $\alpha_i \in A$ is made of an alarm signal $x_{a_i}(t)$ and It is produced as follows:

$$x_{a_i}(t) = \begin{cases} 1 & \text{if} \tilde{x}_{a_i}(t) \in X_{\mathbf{ab}}, \\ 0 & \text{if} \tilde{x}_{a_i}(t) \in X_{\mathbf{n}}. \end{cases}$$

where $X_{\mathbf{n}}$ and $X_{\mathbf{ab}}$ refer to the range of normal and abnormal performance of the process variable $\tilde{x}_{a_i}$, respectively.

In many cases chattering alarms can lead to misidentification of alarm flood, so to avoid false declaration of alarm flood, repeating alarms should be removed first. Various methods have been introduced to remove repetitive alarms, including filters, delay timers, and deadband, among which [1, 2], delay timers can be directly applied to alarm signals. Using an off-delay-timer with sample λ , the alarm $\alpha_i \in A$ is defined as:

$$\alpha_i(t) = \begin{cases} 1 & \text{if} \quad x_{ai}(t) = 1 \text{ and } \quad \forall k \in \{t-\lambda, t-\lambda+1, \ldots, t-1\},\ x_{a_i}(k) = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Detection of alarm floods can be based on the alarm rate, which is a common key indicator for evaluating the performance of an alarm system. The alarm rate $\zeta(t)$ at time t in the time window of length T is defined as follows:

$$\zeta(t) = \sum_{i=1}^{|A|} \sum_{k=t-T+1}^{t} x_{ai}(k). \tag{1.1}$$

Now, based on the alarm rate $\zeta(t)$, the start and end time of the alarm flood can be determined by comparing $\zeta(t)$ with the predetermined threshold. The variable ψ, which indicates the existence of an alarm flood, is defined as follows:

$$\psi(t) = \begin{cases} 1 & \text{if} \quad \zeta(t) \geq \Gamma_s \quad \text{and } \psi(t-1) = 0, \\ 0 & \text{if} \quad \zeta(t) < \Gamma_e \quad \text{and } \psi(t-1) = 1, \\ \psi(t-1) & \text{otherwise} \quad . \end{cases} \tag{1.2}$$

1 and 0 indicate the presence and absence of alarm flood, respectively. The prototype $\psi(0)$ is initialized to 0.

According to the ISA-18.2 [11] standard, the reference thresholds for detecting the start and end of an alarm flood are, respectively, 10 and 5 alarms in a 10-minute period per operator. Here, the two thresholds in the equation (1.2) with $t_s = 10$ and $t_e = 5$ are based on the time window of size $T = 10$ equivalent to minutes 600 seconds in the equation (1.1) . The start and end time of each alarm flood can be determined based on the ψ variable. An alarm flood starts at time t_s if

$$\psi(t_s) = 1 \quad \text{and } \psi(t_s - 1) = 0,$$

which indicates that the alarm rate $\zeta(t)$ reaches a threshold of 10 alarms in a 10-minute period, and ends at time t_e if

$$\psi(t_e) = 0 \quad \text{and } \psi(t_e - 1) = 1,$$

which indicates that the alarm rate $\zeta(t)$ falls below the threshold of 5 alarms in a 10-minute period.

Now, using [3] and based on the above description, the alarm flood can be specified for offline and online mode as explained below.

1.4.1 Offline Detection

In the offline detection of a flood of alarms, the A&E logs that contain the recorded alarm data can be used. By applying a moving time window to the A&E log, the alarm floods are treated as timed sequences and stored in an alarm flood dataset. In this way, each recorded flood warning sequence is recorded as follows:

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

Here, $f_k \in F$ is the kth alarm flood in the recorded alarm flood dataset F, where $k = 1, 2, ..., |F|$; $\alpha_m^k \in A$ is the m warning label in f_k, where $m = 1, 2, ..., |f_k|$; $t_{\alpha_m}^k$ is the activation timestamp associated with α_m^k which belongs to the interval $[t_s^k, t_e^k]$; and t_s^k and t_e^k are the start and end times of alarm flood f_k, respectively.

1.4.2 Online Detection

Detection of alarm floods in real time can be implemented using a time window of width T. This process is shown in the figure 1.3 and is described according to the following equation:

$$\begin{aligned}
f_o^1 &= \left\{ (\alpha_1^o, t_S), (\alpha_2^o, t_{\alpha_2}^o), \ldots, \left(\alpha_{|f_o^1|}^o, t_{\alpha_{|f_o^1|}}^o\right) \right\}, \\
&\vdots \\
f_o^\kappa &= \left\{ (\alpha_1^o, t_S), (\alpha_2^o, t_{\alpha_2}^o), \ldots, \left(\alpha_{|f_o^1|}^o, t_{\alpha_{|f_o^1|}}^o\right), \ldots, \left(\alpha_{|f_o^\kappa|}^o, t_{\alpha_{|f\ kappao|}}^o\right) \right\}, \\
&\vdots \\
f_{full}^o &= \left\{ (\alpha_1^o, t_S), (\alpha_2^o, t_{\alpha_2}^o), \ldots, \left(\alpha_{|f_o^1|}^o, t_{\alpha_{|f_o^1|}}^o\right), \ldots, \left(\alpha_{|f_o^\kappa|}^o, t_{\alpha_{|f_o^\kappa|}}^o\right), \ldots, \left(\alpha_{|f_o^{full}|}^o, t_e\right) \right\}.
\end{aligned}$$

For time t_d when $\psi(t)$ changes from zero to one according to the equation (1.2), the first online alarm flood f_1^o is detected. In each update, the previous alarm instances outside the time interval T are removed and new alarms are added to calculate the alarm rate. As long as $\psi(t)$ is equal to one, the online alarm is updated, and when it changes from one to zero, the end of the alarm flood is detected. Different updates of an online alarm flood are recorded using corresponding timestamps. Here, f_κ^o and f_{full}^o represent the κth update and the last alarm flood. Also, $t_{\alpha_r}^o$ is the alarm occurrence time α_r^o and t_s and t_e are the timestamps of the first and last alarms of the last alarm flood, respectively.

1.4.3 Charts Related to Flood Warning

In this section, some visual charts are presented that provide an overview of the detected alarm floods and alarm patterns. There are three main diagrams for visualizing alarm floods:

- (Alarm Similarity Color Map (ASCM)) Shows cluster correlation of alarms.

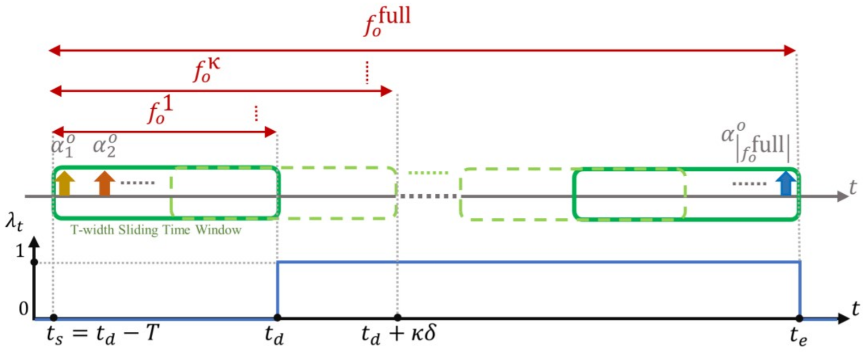

Fig. 1.3 Online alarm flood detection [3]

- 3D bar chart: Displays the rate of occurrence of alarms in a 3D graph. It also includes a graph to compare the rate of alarms to the alarm flood threshold.
- Spiral diagram (Spiral graph) The data shows the time sequence on a spiral, each revolution of which corresponds to one day and night.

ASCM chart It is actually depicting the similarities and differences between different alarms. An example of this diagram in Figure 1.4 it has been shown. The different colors in the graph indicate the degree of similarity, with darker colors representing greater similarity. Also, grouping similar alarms helps highlight patterns and similarities. This chart enables quick and accurate analysis of alarm floods and helps to better identify and understand patterns and relationships between alarms, and is therefore a useful tool for alarm flood analysis. Draw a diagram ASCM It includes the following four steps [14]:

1. Padding each alarm's unique binary sequence with extra 1s: In order to account for communication delays and different time delays between unique alarms, each binary sequence can be padded with additional 1s to increase the number of occurrences of existing alarms. For example, for each warning occurrence in a binary sequence, five 1 are added to each side of the actual occurrence, for a total of eleven 1 corresponding to each alarm occurrence. In other words, the influence interval of each alarm occurrence in the time domain increases to 11 seconds.

2. Calculate the similarity index between the filled binary sequences corresponding to each unique pair of alarms: According to the characteristics of binary sequences, the Jaccard similarity index A common choice for calculating the similarity of alarm sequences is calculated as follows:

$$S_{Jaccard}(A,B) = \max_{l \in L} \frac{a(l)}{a(l)+b(l)+c(l)},$$

where in $a(l)$ The number of matches ($\alpha_i = 1, \beta_{i+l} = 1$), $b(l)$ number of mismatches ($\alpha_i = 1, \beta_{i+l} = 0$) and $c(l)$ are the number of mismatches ($\alpha_i = 0, \beta_{i+l} = 1$), and $\forall i \in [1-l, N]$ To $l \leq 0$ And $\forall i \in [1, N-l]$ To $l > 0$. Also α_i and β_j alarms Two alarm sequences A and B are made according to the previous step.

3. Unique sorting of alarms based on how similar they are to other unique alarms: For more practical visualization, the rows and columns are sorted by hierarchical clustering order.

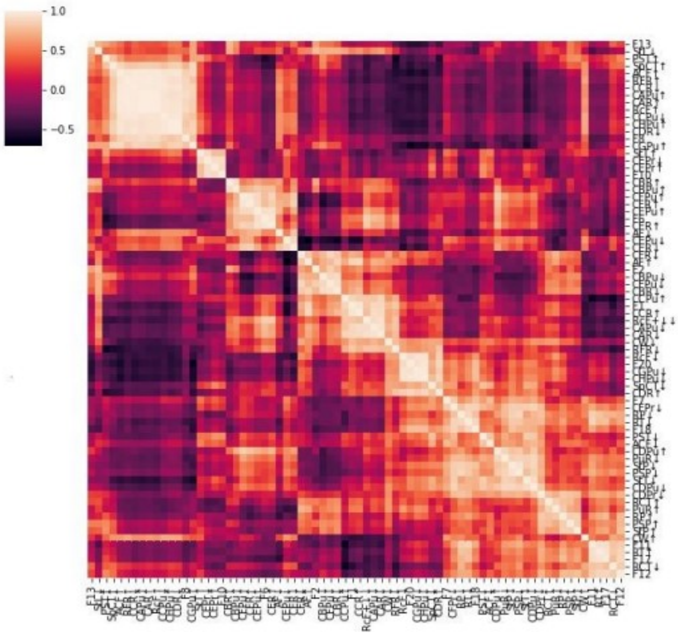

Fig. 1.4 ASCM diagram [13]

4. Draw the ordered similarity matrix to colors: Correlation matrix with reordered rows and columns, as shown in figure 1.4, coded using colors. The names of the alarm labels on the horizontal axis have been removed and their order is the same as the order on the vertical axis.

in the figure 1.5 An example of a 3D bar chart is shown. Each bar represents the number of alarms for an alarm tag in a 10-minute time period. The black curve is the alarm flood rate, and the green and red lines represent the reference thresholds corresponding to 1 and 10 alarms in a 10-minute period. Therefore, the parts of the graph where the black curve exceeds the red threshold indicate the alarm flood, and at the same time, the bar graphs corresponding to these ranges represent the alarm labels and their impact on the corresponding alarm flood.

Figure 1.6 A spiral graph is shown, with green color corresponding to a lower rate and orange color corresponding to a higher rate of alarm occurrence. Also, the red color represents the alarm flood, and according to the times specified in the diagram, you can see the time of its occurrence and end. Also, this graph has the ability to show what times of the day the most alarm floods occur.

1.5 Active Learning

Active learning, a family of machine learning methods that involve querying data instances to be labeled for training by an oracle, has been shown to achieve higher accuracy with fewer labeled examples compared to passive learning. Traditionally, active learning research has focused on selecting queries from the learner's perspective, exploring the question of whether machines can learn with fewer labeled instances if they are allowed to ask questions. Encouraging results have been demonstrated across various problem settings and domains.

For instance, uncertainty sampling-based query algorithms, which select instances with the least label certainty under the current trained model, have shown effectiveness in applications such as Tür et al. [15], and

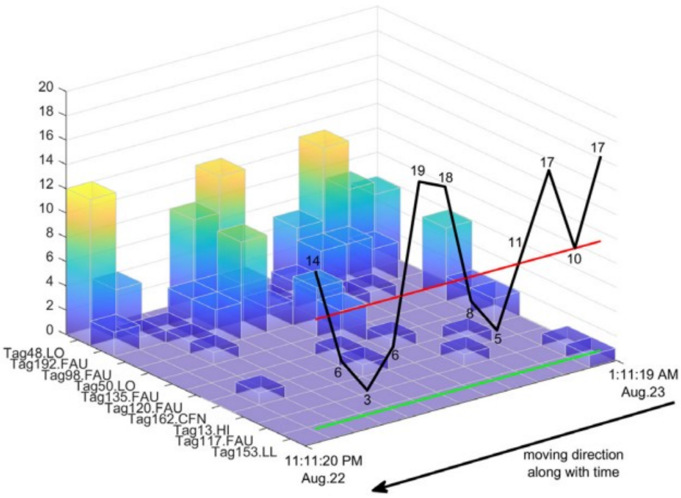

Fig. 1.5 3D bar chart [10]

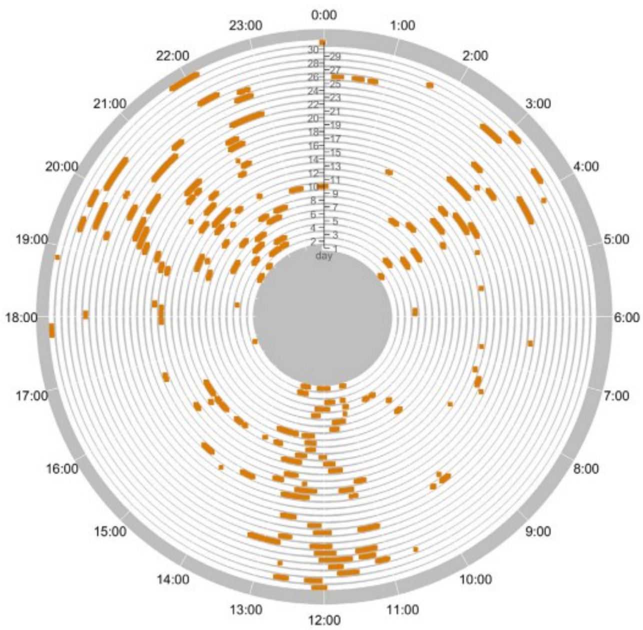

Fig. 1.6 Spiral diagram [10]

Settles and Craven [20]. Query-by-committee algorithms aim to minimize the version space of the model, and theoretical bounds on label complexity have been established for these and related methods (Freund et al. [8], Dasgupta [7]). A more detailed overview of active learning algorithms, along with example references, can be found in Settles [19]. Moreover, active learning has gained traction in industry, with companies like CiteSeer, Google, IBM, Microsoft, and Siemens incorporating active learning in their applications.

However, despite the progress, several open problems remain when it comes to practical applications of active learning. In a recent survey of annotation projects for natural language processing tasks (Tomanek and Olsson [21]), only a small percentage of respondents reported using active learning. Some expressed skepticism about its effectiveness, believing that although it may reduce the number of instances to be annotated, it may not actually decrease the overall annotation time. Additionally, recent empirical studies, including those involving real annotators, have produced mixed or negative results (Schein and Ungar [17], Guo and Schuurmans [9], Baldridge and Palmer [5]). The implementation of query selection methods for complex learning algorithms can also require significant software engineering overhead. These mixed practical results, coupled with high development costs, contribute to the hesitancy among researchers to adopt active learning in ongoing projects.

It can be conjectured that the abundance of positive results in the literature is influenced by simplifying assumptions made in previous work. For example, assumptions such as the presence of a single infallible annotator or uniformly expensive labeling costs have been made to facilitate controlled experiments. However, these assumptions often do not hold in real-world scenarios. Consequently, the research question has shifted toward whether machines can learn more economically by asking questions, incorporating aspects such as annotators, costs, and label noise. This shift in perspective is crucial for advancing active learning research and is the focus of this article.

In the past, early theoretical results for active learning sparked interest in applying these ideas to neural networks (Angluin [4]). However, these methods often assumed that the learner could synthesize arbitrary query instances, and the applications were limited to simple or artificial learning tasks, such as geometrical shapes on a 2D plane. An early attempt at a real-world task by Lang and Baum [6] employed active learning with a human oracle to train a classifier for handwritten characters and digits. They encountered the issue of query images generated by the learner containing no recognizable symbols, only artificial hybrid characters lacking semantic meaning. This negative result motivated the development of selective sampling and pool-based active learning scenarios used today, which ensure that query instances are sensible and representative of the underlying natural distribution.

Similarly, in the present, while active learning has been established as a widely applicable tool across problem domains, its results are influenced by assumptions that primarily focus on the utility of a query to the learner, rather than considering the overall cost to teachers or other aspects of the problem. However, these practical challenges are driving innovation and bringing us closer to effective methods for interactive learning systems.

Chapter 2

Active Alarm Flood Clustering

In this study, our primary focus is on enhancing the performance of a clustering algorithm through the integration of active learning. To demonstrate the effectiveness of our approach in improving any unsupervised flood detection method, we chose to implement a simple yet widely used algorithm, namely K-means. By utilizing K-means as our base flood detection algorithm, we aim to showcase the potential for enhancing its performance through active learning techniques. It is important to note that our implemented methods are not dependent on the specific characteristics of the base algorithm. Therefore, by demonstrating the success of our approach with K-means, we can establish the possibility of applying our methodology to enhance the performance of any other flood detection algorithm.

2.1 Reinforcement Learning

Reinforcement learning (RL) is an interdisciplinary area of machine learning and optimal control, which has gained significant attention due to its ability to enable agents to learn and make decisions in complex, uncertain environments. Unlike supervised learning, which relies on labeled input-output pairs, and unsupervised learning, which does not require explicit correction of sub-optimal actions, reinforcement learning is centered on the concept of learning through interaction with an environment and receiving feedback in the form of rewards or penalties.

At its core, reinforcement learning is about learning how to map situations to actions in a way that maximizes a numerical reward. This fundamental principle underpins the decision-making process of an RL agent, which seeks to identify the most effective actions to take in various states of the environment to achieve long-term goals. The concept of maximizing cumulative reward over time forms the basis of the agent's learning process, driving it to discover optimal strategies through exploration and exploitation.

A key feature of reinforcement learning is its focus on sequential decision-making, where the actions taken by an agent influence not only immediate rewards but also future states and subsequent rewards. This temporal credit assignment aspect of RL has been a subject of extensive research, leading to the development of algorithms such as temporal-difference learning and Q-learning, which are fundamental to the RL framework.

The field of reinforcement learning has witnessed significant advancements, including the integration of deep learning techniques to address complex decision-making problems. This convergence has given rise to the subfield of deep reinforcement learning, which has demonstrated remarkable capabilities in domains such as game playing, robotics, and autonomous systems. The combination of deep neural networks with

RL algorithms has enabled agents to learn directly from high-dimensional sensory inputs, paving the way for unprecedented achievements in artificial intelligence.

Reinforcement learning is characterized by its versatility and applicability to a wide range of domains, spanning from industrial automation and robotics to healthcare and finance. The ability of RL agents to adapt to dynamic environments and learn from experience makes them well-suited for tasks that involve uncertainty, partial observability, and long-term planning. As a result, reinforcement learning continues to be a focal point of research and innovation, driving progress in autonomous systems and intelligent decision-making.

In summary, reinforcement learning represents a powerful paradigm for enabling agents to learn from interaction with their environment and make sequential decisions to achieve long-term objectives. The integration of RL with deep learning has expanded its capabilities, opening new frontiers for AI applications. As research in this field continues to evolve, the potential of reinforcement learning to address complex real-world challenges remains a compelling area of exploration and development.

2.2 Active Learning

Active learning with deep reinforcement learning and human-in-the-loop is a state-of-the-art approach that integrates human expertise and machine learning to enhance flood clustering systems. This innovative framework leverages deep reinforcement learning to train a meta-policy for query selection, explicitly optimizing the number of discovered fooods throughout the querying process. By doing so, it overcomes the limitations of traditional re-ranking strategies and demonstrates significant improvements in long-term performance across various benchmark datasets.

Alarm flood clustering is a critical task in industrial processes. However, traditional unsupervised flood clustering algorithms often suffer from high false-positive rates, necessitating human intervention to verify the identified floods. In practice, analysts or domain experts are employed to investigate the top instances from a ranked list of alarm floods and provide feedback, which can be leveraged to improve the alarm flood clustering system.

In summary, the integration of deep reinforcement learning with human-in-the-loop feedback presents a promising approach to enhancing alarm flood clustering systems. The framework exemplifies the potential of active learning using reinforcement learning to address the challenges of traditional alarm flood clustering methods and achieve significant improvements in long-term performance. By leveraging human expertise in combination with advanced machine learning techniques, such as deep reinforcement learning, the framework represents a significant advancement in the field of active alarm flood clustering.

2.3 The K-means Algorithm

The K-means algorithm is a widely used unsupervised machine learning technique that aims to partition a given dataset into K distinct, non-overlapping clusters. It has proven to be a powerful tool in various domains, including alarm flood detection. In this paper, we will explore the K-means algorithm and its application in clustering alarms to efficiently detect alarm floods.

The K-means algorithm follows a step-by-step process to cluster data points:

1. Initialization: Randomly select K data points as the initial cluster centroids.

2. Assigning Data Points: Assign each data point to the nearest centroid based on a distance metric, typically the Euclidean distance.

3. Updating Centroids: Recalculate the centroids by taking the mean of the data points assigned to each cluster.

4. Repeat Steps 2 and 3 until convergence: Iterate the assignment and update steps until the centroids stabilize or a predetermined number of iterations is reached.

5. Convergence and Result: Once convergence is achieved, the algorithm outputs the final cluster assignments.

Detecting alarm floods using the K-means algorithm involves leveraging its clustering capabilities to identify clusters with a high concentration of alarms within a short period. By following a few key steps, alarm floods can be effectively detected using K-means.

Firstly, the alarm data needs to be prepared and preprocessed. This involves collecting alarm data from various sources and extracting relevant features such as alarm severity, timestamp, device location, or event type. Feature selection is crucial to ensure that only the most informative features are considered, while irrelevant or redundant ones are discarded. Additionally, feature scaling is applied to normalize the selected features, ensuring that they are on a similar scale and prevent any single feature from dominating the clustering process.

Once the data is prepared, the optimal number of clusters, K, needs to be determined. This can be achieved using techniques like the elbow method or silhouette score, which help strike a balance between the number of clusters and the within-cluster variance. With the optimal K determined, the K-means algorithm is applied to the preprocessed alarm data. The algorithm iteratively assigns each alarm to the nearest centroid, calculates the mean of the alarms assigned to each cluster, and updates the centroids accordingly. By analyzing the resulting clusters, clusters with a significantly high number of alarms within a short period can be identified as potential alarm floods. This information can then be used to generate reports or notifications to alert operators or system administrators, enabling them to take appropriate actions and mitigate the alarm flood effectively.

2.4 Active Flood Detection

In this work, we propose a methodology for performing alarm flood detection, specifically focusing on the classification of alarm flood events. Our methodology builds upon the techniques presented in the paper "Meta-AAD: Active Anomaly Detection with Deep Reinforcement Learning" by Daochen Zha et al. [22].

An overview of our approach is illustrated in Figure 2. In the training stage, we extract transferable features as states. We then feed the data into meta-policy in a streaming manner so that the state and action spaces can be significantly reduced. The meta-policy is trained with deep reinforcement learning based on our labeled datasets. Finally, the trained meta-policy can be directly applied to any new unlabeled datasets for active alarm flood clustering without further tuning.

A. *Extracting Transferable Meta-Features*

   Intuitively, there are two types of information that are critical for deciding which instance to query. The first is clusters outputted by the unsupervised clustering algorithm (here, K-means). Second, the labeled classes are helpful. Properly promoting the instances that are similar to these known instances will improve the performance. Based on the intuitions above, we empirically extract some features as follows.

   – **Detector features:** The clusters c outputted by unsupervised clustering algorithms. Any off-the-shelf clustering algorithms can serve as detectors.

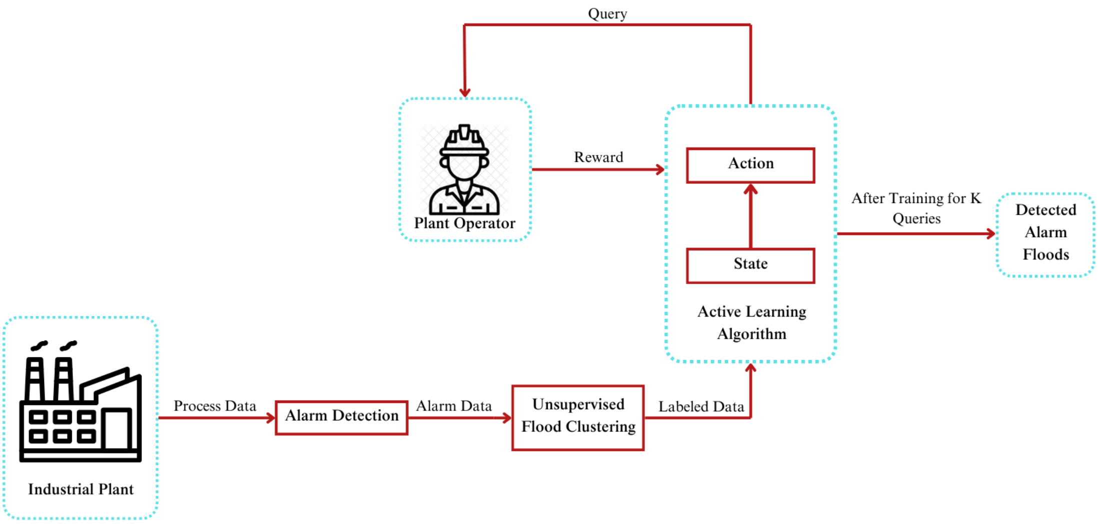

Fig. 2.1 Active Flood Detection

- **Flood features:** The features indicating the relatedness to the labeled instances. In this work, we extract three features for this purpose. We standardize the original features X and calculate the minimum and the mean Euler distances to the labeled instances.

B. *Learning from Data*

The state, action and reward of the Markov Decision Process (MDP) are defined as follows.

- **State** S: The transferable features of the current observed instance
- **Action** A: Actions can be from 0 to 10, while 0 suggests that current instance does not belong to a flood and 1 to 10 suggest that the current instance belongs to the corresponding flood.
- **Reward** R: If the meta-policy queries an instance, we give a positive reward of 1 if the instance is of the right class, and a negative reward of 1 if the instance is not of the right class. We give 0 reward if the meta-policy recognizes an instance as not in a flood. The reward function is critical to describe the desired behaviors.

The above MDP describes an active learning procedure. Intuitively, the meta-policy is encouraged to take action according to the class of the queried instance but only the instances which are in a flood. In this sense, the meta-policy will be taught to discover more floods under a budget.

C. *Training the Policy with Deep Reinforcement Learning*

Given the MDP defined in Section 3.2-B, we can train the meta-policy with any deep reinforcement learning (DRL) algorithms. In this work, we instantiate our framework with Proximal Policy Optimization (PPO) [18].

Algorithm 1 Training Meta-Policy with PPO

Require: Dataset D with n instances and features X, labels y
Ensure: Trained meta-policy π_θ for query selection

Initialize meta-policy parameters θ
Preprocess D to extract transferable meta-features G
for episode $= 1$ to N_{episodes} **do**
    Initialize state s with G, y, and clustering scores c
    Initialize empty buffer for storing trajectory
    for $t = 1$ to $T_{\max}$ **do**
        Select action a_t from meta-policy π_θ
        Execute action a_t to query the analyst
        Receive feedback $\hat{y}_t$ from the analyst
        Update state s with $\hat{y}_t$
        Store (s, a_t) in the buffer
        if $t \mod T_{\text{update}} = 0$ **then**
            Estimate advantage values A using value function
            Compute policy gradient with PPO loss
            Update meta-policy parameters θ using optimizer
        end if
        if $t \mod T_{\text{eval}} = 0$ **then**
            Evaluate current meta-policy on validation set
        end if
    end for
end for
return trained meta-policy π_θ =0

Chapter 3

Case Study and Evaluation

We used the dataset obtained by Niyazmand et al. [16]. they obtained all the alarms for the first separation stage of a natural gas processing plant. The alarms are collected in a period of 7 months with about 18 000 alarms for 112 tags (unique alarms). A set of three key attributes, namely the time the alarm is triggered, the name associated with the alarm, and a unique identifier, are typically sufficient to identify a specific alarm. By using the tag name and alarm identifier, one can pinpoint the precise location and type of the alarm. Consequently, a singular alarm within the system can be uniquely determined by combining the tag name and alarm identifier. Table 3.1 displays a selection of alarms that were activated within a one-minute time-frame in the natural gas processing facility.

In [16], To prepare the data for sequential pattern mining, first, chattering alarms within a time window of $T_w = 100$ s were removed. This procedure reduced the total number of alarms from 18 000 to about 5000. The 75% reduction in alarm count shows that in this plant, chattering is a serious problem and needs further attention. The alarm floods, then, were identified. Alarm floods begin when the alarm rate grows to more than 10 alarms per 10 min and ends when it reaches zero. For this plant, 9 alarm floods were identified in the period of 7 months. The number of alarms within each flood are given in Table 3.2.

Next, we employed the annotated dataset created by Niyazmand et al. [16] to simulate our methodology. Initially, we converted the time labels in the dataset into six distinct labels: year, month, day, hour, minute, and second. Additionally, we eliminated the alarm tags since the alarm identifier numbers were deemed sufficient for our purposes.

The K-means algorithm exhibited a low accuracy rate of 42% (ARI less than 0.1), as anticipated. The results of our algorithm applied to the dataset are presented in Table 4. The table illustrates the number of accurate flood classifications obtained when querying the top 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 alarms with the highest probabilities of being associated with a class. The Meta-policy wsa trained using less than 2000 alarm in each timestep. It is evident from the table that the results are highly satisfactory.

Tag name & identifier	Tag description	Alarm time
$\mathrm{TAL}_1 29C$	E403 gas low temp	4/8/2015 4:50:14
$\mathrm{TAH}_1 29C$	E403 gas high temp	4/8/2015 4:50:24
$\mathrm{AH}_1 10C$	Well 7+9 high flow	4/8/2015 4:50:37
$\mathrm{LAH}_1 14C$	S205 high level H/C	4/8/2015 4:50:54

Table 3.1 An example of alarm message logs

Alarm flood	Alarm count
flood.1	32
flood.2	32
flood.3	51
flood.4	80
flood.5	10
flood.6	36
flood.7	89
flood.8	16
flood.9	83

Table 3.2 Alarm flood sequences in 7 months

Queries	10	20	30	40	50	60	70	80	90	100
Reward	10	20	27	27	27	36	46	56	66	76

Table 3.3 Rewards of Meta-Policy per number of queries

Chapter 4

Conclusion

In conclusion, this thesis has presented a comprehensive and innovative framework for active alarm flood detection with deep reinforcement learning. The framework addresses the challenges associated with alarm flood detection and management in industrial processes, offering a promising solution for enhancing alarm system operation and management.

The proposed framework combines reinforcement learning techniques with alarm flood similarity analysis to optimize the accuracy and efficiency of alarm pattern mining and root cause analysis. By leveraging deep reinforcement learning, a meta-policy for query selection is trained, guiding the alarm flood similarity analysis process. This integration of reinforcement learning enables the framework to adaptively learn and improve its performance over time.

Through experiments conducted on an openly accessible dataset based on the 'Tennessee-Eastman-Process', the proposed framework has demonstrated superior performance compared to existing alarm flood similarity analysis methods. It has showcased its ability to improve the accuracy and robustness of alarm flood detection while minimizing the impact of irrelevant alarms and the ambiguity of alarm order. The results highlight the significant potential of the framework in enhancing the reliability and efficiency of alarm systems in industrial processes.

The integration of the proposed framework into industrial processes has several benefits. Firstly, it provides operators with timely and relevant information, enabling them to make informed decisions and take appropriate actions in response to critical events. This improves the overall safety and reliability of industrial processes, reducing the risk of accidents and minimizing downtime.

Furthermore, the framework contributes to the optimization of resource utilization and process efficiency. By accurately detecting and analyzing alarm floods, it helps identify the root causes of alarms and facilitates the implementation of effective corrective actions. This leads to improved process performance, reduced downtime, and enhanced productivity.

The proposed framework also addresses the challenges posed by the increasing complexity of industrial processes. With the integration of deep reinforcement learning, the framework can adapt and learn from historical data, enabling it to handle intricate and evolving operational environments. This adaptability is crucial in ensuring the framework's applicability to a wide range of industrial domains and its ability to address varied operational scenarios.

While this thesis has made significant contributions to the field, there are opportunities for future research and development. Further exploration of reinforcement learning techniques, such as hierarchical reinforcement learning or multi-agent reinforcement learning, could enhance the framework's performance and scalability.

Additionally, integrating real-time data streams into the framework would enable dynamic and continuous alarm flood detection, improving its responsiveness and adaptability.

Applying the proposed framework to diverse industrial domains and evaluating its performance in different operational scenarios would provide valuable insights into its generalizability and effectiveness. Furthermore, conducting comparative studies with other state-of-the-art alarm flood detection methods would contribute to a deeper understanding of the framework's strengths and limitations.

In conclusion, this thesis has presented a novel framework that combines deep reinforcement learning with alarm flood similarity analysis to address the challenges in alarm flood detection and management. The framework offers a promising solution for enhancing the accuracy, efficiency, and effectiveness of alarm systems in industrial processes. By improving the reliability, safety, and productivity of industrial processes, the proposed framework contributes to the advancement of industrial automation and management in the face of increasingly complex operational environments.

References

[1] Adnan, N. A., Izadi, I., and Chen, T. (2011). On expected detection delays for alarm systems with deadbands and delay-timers. *Journal of Process Control*, 21(9):1318–1331.

[2] Afzal, M. S., Chen, T., Bandehkhoda, A., and Izadi, I. (2018). Analysis and design of time-deadbands for univariate alarm systems. *Control Engineering Practice*, 71:96–107.

[3] Alinezhad, H. S., Shang, J., and Chen, T. (2022). Open set online classification of industrial alarm floods with alarm ranking. *IEEE Transactions on Instrumentation and Measurement*, 72:1–11.

[4] Angluin, D. (1988). Queries and concept learning. *Machine Learning*, 2(4):319–342.

[5] Baldridge, J. and Palmer, A. (2009). How well does active learning *actually* work? Time-based evaluation of cost-reduction strategies for language documentation. In Koehn, P. and Mihalcea, R., editors, *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 296–305, Singapore. Association for Computational Linguistics.

[6] Blum, A., Chalasani, P., Goldman, S. A., and Slonim, D. K. (1998). Learning with unreliable boundary queries. *Journal of Computer and System Sciences*, 56(2):209–222.

[7] Dasgupta, S. (2004). Analysis of a greedy active learning strategy. In Saul, L., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press.

[8] Freund, Y., Seung, H., Shamir, E., and Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168.

[9] Guo, Y. and Schuurmans, D. (2007). Discriminative batch mode active learning. *Advances in neural information processing systems*, 20.

[10] Hu, W., Al-Dabbagh, A. W., Chen, T., and Shah, S. L. (2018). Design of visualization plots of industrial alarm and event data for enhanced alarm management. *Control Engineering Practice*, 79:50–64.

[11] ISA (2009). *Management of Alarm Systems for the Process Industries*. International Society of Automation.

[12] Izadi, I., Shah, S. L., Shook, D. S., and Chen, T. (2009). An introduction to alarm analysis and design. *IFAC-PapersOnLine*, 42(8):645–650.

[13] Khaleghy, H. and Izadi, I. (2021). Detection of correlated alarms using graph embedding. In *2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–7. IEEE.

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

[15] Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In Croft, B. W. and van Rijsbergen, C. J., editors, *SIGIR '94*, pages 3–12, London. Springer London.

[16] Niyazmand, T. and Izadi, I. (2019). Pattern mining in alarm flood sequences using a modified prefixspan algorithm. *ISA Transactions*, 90:287–293.

[17] Schein, A. I. and Ungar, L. H. (2007). Active learning for logistic regression: an evaluation. *Machine Learning*, 68:235–265.

[18] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

[19] Settles, B. (2009). Active learning literature survey.

[20] Settles, B. and Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In Lapata, M. and Ng, H. T., editors, *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, Honolulu, Hawaii. Association for Computational Linguistics.

[21] Tomanek, K. and Olsson, F. (2009). A web survey on the use of active learning to support annotation of text data. In Ringger, E., Haertel, R., and Tomanek, K., editors, *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 45–48, Boulder, Colorado. Association for Computational Linguistics.

[22] Zha, D., Lai, K.-H., Wan, M., and Hu, X. (2020). Meta-AAD: Active anomaly detection with deep reinforcement learning. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 771–780.