

RESEARCH ARTICLE  

Convolutional kernel-based classification of industrial alarm floods

Gianluca Manca^{1,2}  and Alexander Fay¹

¹Institute of Automation Technology, Helmut-Schmidt-University Hamburg, Hamburg, Germany

²Industrial AI, ABB Corporate Research Center, Ladenburg, Germany

Corresponding author: Gianluca Manca; Email: gianluca.manca@de.abb.com

Received: 13 November 2023; **Revised:** 03 June 2024; **Accepted:** 11 July 2024



Keywords: abnormal situations; industrial alarm floods; industrial process diagnosis; open-set classification; time series classification

Abstract

Alarm flood classification (AFC) methods are crucial in assisting human operators to identify and mitigate the overwhelming occurrences of alarm floods in industrial process plants, a challenge exacerbated by the complexity and data-intensive nature of modern process control systems. These alarm floods can significantly impair situational awareness and hinder decision-making. Existing AFC methods face difficulties in dealing with the inherent ambiguity in alarm sequences and the task of identifying novel, previously unobserved alarm floods. As a result, they often fail to accurately classify alarm floods. Addressing these significant limitations, this paper introduces a novel three-tier AFC method that uses alarm time series as input. In the transformation stage, alarm floods are subjected to an ensemble of convolutional kernel-based transformations (MultiRocket) to extract their characteristic dynamic properties, which are then fed into the classification stage, where a linear ridge regression classifier ensemble is used to identify recurring alarm floods. In the final novelty detection stage, the local outlier probability (LoOP) is used to determine a confidence measure of whether the classified alarm flood truly belongs to a known or previously unobserved class. Our method has been thoroughly validated using a publicly available dataset based on the Tennessee-Eastman process. The results show that our method outperforms two naive baselines and four existing AFC methods from the literature in terms of overall classification performance as well as the ability to optimize the balance between accurately identifying alarm floods from known classes and detecting alarm flood classes that have not been observed before.

Impact Statement

We introduce the convolutional kernel-based alarm subsequence identification method (CASIM), which improves industrial alarm flood classification. CASIM extracts a wide range of alarm dynamics, unlike previous approaches that use a limited set of alarm characteristics. This helps CASIM identify more relevant features, improving its ability to classify complex alarm floods. Moreover, expanding windows in CASIM's online application, inspired by early time series classification, allows alarm flood classification over time. Our evaluation shows that this can provide faster and more accurate insights than existing methods. We believe that our proposed method CASIM can improve operational decision-making and reduce operator effort. By making the implementation of our method publicly available, we aim to encourage wider adoption and research in the field.

  This research article was awarded Open Data and Open Materials badges for transparent practices. See the Data Availability Statement for details.

© The Author(s), 2024. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.



1. Introduction

Due to advancements in automation technology, modern industrial process plants have become more data intensive. The amount of data collected and stored annually, such as time series readings from sensors and alarm logs, may reach hundreds of gigabytes (Klopper et al., 2016). These data can potentially be used in machine learning (ML) to gain insight into a process's complex interdependencies and behavior. Consequently, ML can offer a human operator valuable decision support (Manca et al., 2021).

Process control systems implement alarms that are triggered when a predetermined threshold in a process variable, such as the level in a column, is exceeded. Alarms inform operators of critical process deviations requiring manual intervention (Takai et al., 2012). Ideally, the number of simultaneous alarms should be kept to a minimum (EEMUA, 2013). In more complex abnormal situations, however, many alarms can be activated in a short period. This is referred to as an alarm flood (Takai et al., 2012; EEMUA, 2013), and it has the potential to impair the operator's situational awareness, making prompt and precise decisions difficult (ASM Joint R&D Consortium, 2009; Mustafa et al., 2023).

Alarm floods are typically caused, among other things, by propagating disturbances that result in deviations and, consequently, the activation of alarms in different but interconnected plant sections (Wang et al., 2016; Mustafa et al., 2023). In such a scenario, the operator may not be able to rely on addressing activated alarms in chronological order, as arbitrary alarm thresholds may prevent critical alarms from being activated first (Rodrigo et al., 2016). Instead, the operator must initiate a decision-making process, during which the evaluation of the underlying abnormal situation can be a time-consuming and challenging manual task (ASM Joint R&D Consortium, 2009; Takai et al., 2012; EEMUA, 2013). Here, an ML-based analysis may capture implicit patterns and knowledge from historical alarm floods and provide the operator with valuable insights in the event that a similar situation occurs. Such advanced operator support could facilitate the safe and effective restoration of a desired process (ASM Joint R&D Consortium, 2009; Takai et al., 2012; EEMUA, 2013; Wang et al., 2016; Lucke et al., 2019; Manca & Fay, 2022; Mustafa et al., 2023).

One type of alarm data analysis technique is alarm flood classification (AFC), which classifies recurrent alarm flood situations based on similar historical alarm floods (Lucke et al., 2019) or, more generally, alarm subsequences (ASs). In this context, ASs are smaller subsets of a potentially infinite alarm data stream (Ahmed et al., 2013; Manca & Fay, 2021b; Vogel-Heuser et al., 2015). Specifically, AFC seeks to determine whether an AS belongs to an existing class of historical ASs or if it represents a novel, previously unobserved AS class (Lucke et al., 2019). The operator could then be provided with readily available and useful information regarding the most likely AS class, such as the underlying root cause or recommendations regarding appropriate actions (Lucke et al., 2019; Parvez et al., 2022). As a result, AFC methods may reduce task saturation among human experts and relieve them of time-consuming and error-prone manual assessment.

In recent years, a substantial body of research has emerged to address the challenges presented by AFC methods. While these methods have brought about advancements, they also reveal persistent limitations, as detailed in Section 2. Specifically, existing methods often fail to represent the intricate and diverse dynamic characteristics inherent in alarm data. By focusing predominantly on a single property, for example, the alarm activation order, these methods fail to capture the full range of characteristics exhibited by the alarm data. This observation underpins the motivation for our research. In response, this paper introduces a novel AFC method designed to more comprehensively harness the dynamic properties of alarms. Our contributions are twofold. First, we adopt a state-of-the-art time series transformation and classification technique and tailor it to industrial alarm data. Second, we adapt a detection method for open-set classification, enabling the distinction between familiar AS classes and those previously unobserved. With these improvements, our method promises a more nuanced and detailed representation of alarm dynamics, addressing the gaps observed in AFC.

Building on our previous work in Manca and Fay (2022),¹ which was presented and discussed at the "IEEE 20th International Conference on Industrial Informatics" in Perth, Australia, in July 2022, this

¹ <https://doi.org/10.1109/INDIN51773.2022.9976139>.

paper introduces significant enhancements, including an oversampling technique to tackle class imbalance—a common challenge in industrial datasets. By generating synthetic samples, our method aims to offer a more robust solution suitable for both practitioners and researchers. Furthermore, our improved AFC method incorporates the use of an expanding windows strategy, which is applied during both the off-line training phase and the online inference phase. This approach, inspired by developments in the field of early time series classification (ETSC), enables our method to adapt to local alarm dynamics as an AS unfolds, ensuring more prompt and accurate classifications. This feature represents a notable improvement over our previous approach in Manca and Fay (2022), which was limited to the off-line classification of alarm floods. Further improvements encompass an expanded coverage of related work, a more comprehensive exposition of our novel method, and a more thorough evaluation.

Our contribution to the field extends beyond theoretical advancements. In a commitment to fostering growth and transparency in industrial artificial intelligence (AI) applications, we have made our code and its usage rights openly accessible.² This decision underscores our belief in the transformative potential of open-source collaboration to address complex challenges in alarm management and process safety.

The remainder of the paper is structured as follows: Section 2 outlines the AFC requirements and examines the related work. In Section 3, a novel AFC method is developed. In Section 4, a publicly available dataset is utilized to evaluate and compare our proposed method with two naive baselines and four relevant methods from the literature. Finally, this paper concludes in Section 5 with the most significant findings and suggestions for future research.

2. Related work

2.1. Overview and requirements

Both Parvez et al. (2022) and Alinezhad et al. (2023) provided detailed reviews of existing AFC methods. All methods described in these reviews presume that similar ASs result from similar abnormal situations (Rodrigo et al., 2016). The definitions of similarity, however, vary to some extent (Lucke et al., 2019). In fact, there is no consensus on a standard set of AFC requirements. Instead, research on the clustering of similar ASs provides relevant criteria, which we assert are also applicable in AFC. In Cheng et al. (2013), two requirements (*R1* and *R2*) are defined:

R1: AFC methods should be tolerant of irrelevant alarm activations from ASs that stem from similar situations.

R2: AFC methods should permit variations in the alarm activation order from ASs caused by similar situations.

In this context, irrelevant alarms are those that are activated for a short period or for a small number of similar ASs and are thus not representative of the underlying situation (Manca et al., 2021). For instance, Charbonnier et al. (2015) defined irrelevant alarms as those that occur in less than half of all alarm floods within a given class. Typically, the alarm management system within a process control system is responsible for handling short-term alarms, for example, using delay timers or deadbands (Takai et al., 2012; EEMUA, 2013). However, not all irrelevant alarms may be effectively managed, inadvertently blurring certain pertinent alarm dynamics (Wang et al., 2016; Lucke et al., 2019). Therefore, AFC methods should be capable of handling irrelevant alarms independently.

Figure 1 shows two exemplary ASs, depicted as alarm sequences, that illustrate the phenomena underlying *R1* and *R2*. Alarm sequences are lists of chronologically ordered and time-stamped alarm activations in alarm variables, which are the unique identifiers for the alarms defined in a process control system (Manca et al., 2021). Both ASs in Figure 1 are included in the dataset used for evaluation in Section 4 and are derived from similar root causes. Nevertheless, Figure 1 reveals that both ASs differ in

²<https://doi.org/10.24433/CO.4874993.v1>.

Alarm Sequence "A"		Alarm Sequence "B"	
Alarm Variable	Activation Time	Alarm Variable	Activation Time
XMEAS1-L	00:00:00	XMEAS10-L	00:00:00
XMV3-H	00:35:20	XMV3-H	00:02:40
XMEAS38-H	01:26:00	XMEAS38-H	00:57:30
XMEAS31-H	01:44:20	XMEAS49-H	01:18:20
XMEAS49-H	01:46:40	XMEAS23-L	01:48:10
XMEAS23-L	01:54:20	XMEAS33-H	01:55:00
XMEAS27-H	01:58:10	XMEAS29-L	01:57:00
XMEAS25-H	02:15:40	XMEAS31-H	01:57:20
XMEAS29-L	02:20:30	XMEAS27-H	01:57:40
XMEAS33-H	02:35:00	XMEAS36-L	02:13:50
XMEAS36-L	02:37:30	XMEAS25-H	02:15:50

Figure 1. Two alarm sequences "A" and "B". The alarm variable column shows the name of the process variable (XMEAS) or manipulated variable (XMV) and the activated alarm type (L: low, H: high). The black lines between the two sequences connect pairs of identical alarm variables.

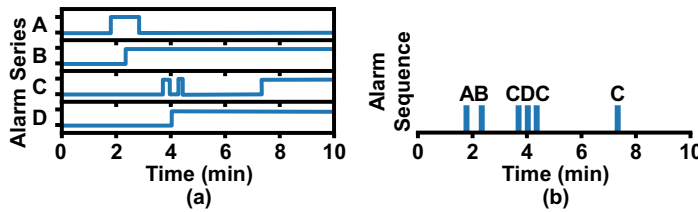


Figure 2. Two types of alarm data representations for alarm variables A to D. (a) Alarm series. The solid blue lines represent alarm variable time trends. A higher level represents an active alarm. (b) Alarm sequence. The solid blue lines indicate alarm activations.

terms of the activated alarms (*R1*) and the sequence in which the alarms are activated (*R2*), as denoted by the black lines connecting corresponding alarms in both sequences.

In the context of real-world industrial scenarios, where a set of observed disturbances and abnormal situations may grow throughout the lifespan of a plant, Alinezhad et al. (2023) introduced an additional open-set requirement (*R3*) for AFC:

R3: AFC methods should detect ASs that belong to previously unobserved classes.

Most AFC methods described in the literature use a multiclass classification approach with supervised learning. These methods rely on historical alarm data and annotated class labels (Alinezhad et al., 2023) and typically follow a three-tier structure. That is, during the transformation stage, the most important features or characteristics are extracted from the alarm data of a new AS. Next, in the classification stage, a classifier matches the extracted features to the most fitting class of historical ASs. Finally, in the novelty detection stage, a threshold is used to determine whether the AS to be classified belongs to the most likely known class or is derived from a novel previously unobserved class (Lucke et al., 2019). Despite this common approach, AFC methods can be categorized based on their alarm data input (Lucke et al., 2019), which can be a set, sequence, or series representation. An alarm set comprises the unique alarm variables that are activated at least once in an AS. An alarm series consists of multiple binary time series that show the dynamic activity of individual alarm variables (Lucke et al., 2019). Figure 2 illustrates alarm series and sequence representations of a typical AS. The corresponding alarm set consists of the alarm variables A, B, C, and D.

2.2. Alarm set-based methods

The majority of alarm set-based AFC methods utilize string metrics, which estimate the distance between any two ASs and do not emphasize the number or order of alarm activations (Lucke et al., 2019). A weighted dissimilarity index-based method that converts alarm data into binary alarm set vectors was proposed by Charbonnier et al. (2015). For each historical AS class, a single template alarm set vector is generated, containing only those alarm variables active in at least half of the ASs in the class. The resulting templates are additionally weighted considering how characteristic a certain alarm variable is of a class compared to all the other classes. In the case of a new AS, the dissimilarities between it and each of the weighted historical templates are calculated. This is followed by a classification stage using a nonparametric first-nearest-neighbor (1NN) classifier and a novelty detection stage with a to-be-set threshold. Henceforth, this AFC method is referred to as *WDI-INN*.

Reference Charbonnier et al. (2015) demonstrated *WDI-INN*'s improved AFC accuracy compared to that of other methods. In fact, by removing the order and number of alarm activations, as well as extracting only the most relevant alarms, *WDI-INN* becomes more resistant to the ambiguities described in *R1* and *R2*. Despite these advantages, string metrics tend to exaggerate the similarity of two ASs that share alarms but have substantial differences in their respective dynamics, which may hinder the ability to detect previously unobserved classes (*R3*) (Charbonnier et al., 2015; Manca et al., 2021). The Jaccard distance, which quantifies the unweighted disagreement on alarm sets in two ASs, is a less extensive string metric that can also be used for AFC (Fullen et al., 2018; Lucke et al., 2019).

2.3. Alarm sequence-based methods

Most alarm sequence-based AFC methods align the alarm activations in two ASs (Lucke et al., 2019). For example, in Cheng et al. (2013), a local sequence alignment approach, the Smith–Waterman algorithm, was modified to allow for a swapped order of alarm activations if the affected alarms are close in time (*R2*). However, penalizing a difference in the number of alarm activations between ASs makes this method less resistant to irrelevant alarms (*R1*) (Manca et al., 2021). Nonetheless, the method proposed by Cheng et al. (2013) remains an often-used benchmark in AFC (Lai et al., 2017; Lucke et al., 2019; Parvez et al., 2022). This method is used to calculate the pairwise distances between an AS to be classified and a set of historical ASs. The new AS is then classified using a 1NN classifier in the classification stage and a distance threshold in the novelty detection stage.

The high computational cost of the pairwise AS distance calculation is one limitation of the modified Smith–Waterman algorithm (Lucke et al., 2019). Additionally, if the historical AS closest to the AS to be classified was previously mislabeled, the class predicted by the 1NN classifier would be incorrect as well. Furthermore, using a simple detection threshold on AS distances may not explain that different classes may exhibit different intraclass densities and, consequently, different distance distributions, making it more challenging to tune this threshold and detect previously unobserved AS classes (*R3*) (Alinezhad et al., 2022b). To solve some of these limitations, Lai and Chen (2017) and Lai et al. (2019) extended the method proposed by Lucke et al. (2019) using a pattern extraction technique that generates a single AS pattern for each class.

Other AFC methods apply alternative sequence alignment approaches, such as the basic local alignment search tool (BLAST) (Hu et al., 2016), the match-based accelerated alignment (MAA) (Guo et al., 2017), or the Needleman–Wunsch algorithm (Charbonnier et al., 2016; Parvez et al., 2020).

With an exponentially attenuated component (EAC) vector representation, Shang and Chen (2019) proposed an alarm sequence-based AFC method that emphasizes earlier activated alarms to a greater extent while maintaining their chronological order. To define the EAC feature vector of any AS, the time distance between the first activated alarm of each alarm variable and the start of the AS is calculated. To weight earlier alarm activations more, the relative activation time information is incorporated using an attenuation coefficient, which is a parameter specific to the process's dynamic characteristics. After calculating the EAC feature vectors for all historical ASs, a k-d tree is constructed to facilitate an efficient nearest neighbor search.

For classifying a new AS, the method proposed by Shang and Chen (2019) uses a 1NN classifier based on the distances incorporated in the constructed k-d tree, e.g., utilizing the L1 or L2 norm. Alternately, in

Alinezhad et al. (2022a), Gaussian mixture models (GMMs) were utilized to estimate the posterior probability for each AS class. For the final novelty detection stage, Alinezhad et al. (2022a) implemented a detection threshold applicable to the posterior class probabilities of the GMM as well as the distances computed in Shang and Chen (2019) (R3). Henceforth, the AFC method proposed by Shang and Chen (2019) with the detection threshold used by Alinezhad et al. (2022a) is referred to as *EAC-INN*.

One benefit of *EAC-INN* is that using the Euclidean distance on the EAC-weighted relative time distances between alarm activations could help smooth out variations in the alarm order if the activations are close together (R2). However, the emphasis on a single alarm activation per alarm variable as well as *EAC-INN* weighting may be viewed as limitations in cases where the new AS deviates from historical patterns over time. A further limitation of EAC's feature vectors is that *EAC-INN* may not be completely tolerant of irrelevant alarms if they occur early and therefore have high weights (R1).

Most recently, Alinezhad et al. (2023) presented a novel alarm sequence-based AFC method, building upon a previous method outlined in Alinezhad et al. (2022b). This method uses a modified bag-of-words (MBW) approach, a vectorization concept from natural language processing that employs the term frequency-inverse document frequency (TF-IDF), and a weighting strategy to incorporate key characteristics from alarm sequences into a feature vector. The latter is a vector with n dimensions, where n is the number of alarm variables implemented. Each feature includes three weighting terms: the term frequency, the inverse document frequency, and the time weight. Similar to the *EAC-INN* time weight in Shang and Chen (2019), the time weight in Alinezhad et al. (2023) preserves information about temporal characteristics and alarm order by assigning greater weights to earlier activations. The set of historical MBW feature vectors is calculated based on the historical ASs and then used to train multiple binary logistic regression classifiers, one for each existing AS class.

When a new AS emerges, its MBW feature vector is calculated and provided as input to the logistic regression classifiers. The latter returns an estimate for the respective AS class probabilities. Instead of using a single threshold for all classes in the novelty detection stage, the authors of Alinezhad et al. (2023) used an individual threshold for each class to account for different classes with distinct probability distributions. Consequently, a threshold estimation technique is proposed that models each class's probability distribution as one-half of a Gaussian distribution using the historical ASs' probability estimates. Each threshold is then calibrated based on the 95% confidence interval of the respective Gaussian distribution. The resulting set of thresholds is then used to determine whether the new AS belongs to a known class or a novel class. Henceforth, this AFC method is referred to as *MBW-LR*.

Because the transformation stages of *MBW-LR* and *EAC-INN* are similar, some of *EAC-INN*'s advantages and limitations apply, such as its tolerance for alarm order variations (R2). Furthermore, for novelty detection, *MBW-LR* has the distinct advantage of not requiring manual threshold tuning. However, if historical AS classes have a high intraclass variance of probability estimates, using the standard deviation to calculate the corresponding thresholds can result in high values, which may impede the detection of any novel classes (R3). Furthermore, *MBW-LR* may be sensitive to irrelevant alarm activations (R1) and assign them a high IDF weight when the corresponding alarm variable is rarely activated (Manca et al., 2021).

In addition, AFC methods have been proposed that make use of different classification techniques. For example, in Zhou et al. (2022), a modified closed fast sequence mining algorithm was used to detect frequent alarm itemsets. In Dorgo et al. (2018), long short-term memory unit-based recurrent neural networks were used to classify ASs according to their chronologically ordered alarm activations and deactivations. All sequence alignment methods discussed here rely on the activated alarm order to classify an AS. In industrial processes, however, the dynamic behavior of process variables is not always deterministic. Thus, alarm activations and their order can be arbitrary and volatile (R1 and R2) (Charbonnier et al., 2015; Lucke et al., 2019; Manca et al., 2021; Manca & Fay, 2021b; Rodrigo et al., 2016).

2.4. Alarm series-based methods

Alarm series were suggested for AFC methods because these methods consider the dynamics of an AS to a greater extent, whereas the chronological order of alarm activations is less important (R2) (Lucke et al.,

2019; Manca et al., 2021). The AFC method proposed by Lucke et al. (2019) interprets alarm series as multivariate time series and implements a time series classification approach. First, each historical AS is transformed into an alarm coactivation matrix (ACM), where alarm variable pairs are assigned a similarity measure according to the amount of time they are simultaneously active. The resulting matrices are used to train a set of support vector machines (SVMs). If a new AS occurs, both the corresponding ACM and the trained SVM-based classifier are used in the classification stage to determine the most likely AS class. Next, the posterior class probability of the latter is compared to a threshold in the novelty detection stage to decide whether the AS to be classified belongs to the most likely known class or to a novel class (Lucke et al., 2019). Henceforth, this AFC method is referred to as *ACM-SVM*.

One limitation of *ACM-SVM* is derived from the representation of the alarm series using alarm coactivations. That is, the pairwise similarities in an ACM conceal relevant information about the dynamics of an AS, such as the approximate time windows where alarms are active, as well as the situation's overall duration. Disregarding these dynamics in the AFC may lead to a more ambiguous AS representation. Moreover, the posterior class probabilities used in *ACM-SVM* can cause some deficits if two or more classes in the training data are close to each other and a new AS belongs to either of them. In that case, the probabilities of the most likely classes might be low since the sum of all probabilities always equals one. This limits the dynamic range of the detection threshold and could result in erroneous results when attempting to differentiate between known and previously unobserved classes (*R3*).

2.5. Findings summary

Upon careful examination of the existing AFC methods, a number of significant limitations become apparent in the literature. Firstly, existing methods do not sufficiently account for the inherent ambiguity in alarm sets and sequences, resulting in the misclassification of ASs when irrelevant alarms or variations in the order of alarm activations obscure the true class (*R1* and *R2*). Secondly, the detection of novel, previously unobserved ASs classes remains a challenge, as existing AFC methods often lack the flexibility and sensitivity required to identify newly emerging patterns that differ significantly from known classes (*R3*). These limitations justify the proposal of a novel AFC method that makes use of the most relevant characteristics in an alarm series and enables a more advanced distinction between the known and novel AS classes.

3. Proposed approach

3.1. Overview of the proposed approach

In response to the limitations discussed in Section 2 and recent advances in time series classification, we propose a novel three-tier AFC method. First, in the transformation stage, alarm series are fed into an ensemble of convolutional kernel-based multivariate time series transformations that can handle irrelevant alarm activations (*R1*) and alarms in varying order (*R2*) to extract a wide range of alarm dynamics from historical ASs. This approach significantly enhances our ability to discern the nuanced dynamics within ASs, in contrast to existing AFC methods that only focus on a limited number of dynamic characteristics. Next, this information is fed into the classification stage, where an ensemble of linear ridge regression classifiers is used to learn the typical characteristics of the AS classes. In the final novelty detection stage, a local outlier-based novelty detection method is used to determine whether a new AS belongs to a known class or to a novel class (*R3*). This differs from existing AFC methods, where the output of the classification stage is typically utilized directly in the novelty detection stage. Here, our method enables a more distinct assessment of the novelty of an AS to be classified. These innovations collectively address the identified gaps in Section 2 by offering a more flexible and accurate method for AFC.

Figure 3 presents a detailed formalized process description (VDI/VDE, 2015) of our proposed convolutional kernel-based alarm subsequence identification method (*CASIM*). Central to *CASIM* is its bifurcation into two distinct operational phases: the off-line training phase and the online inference

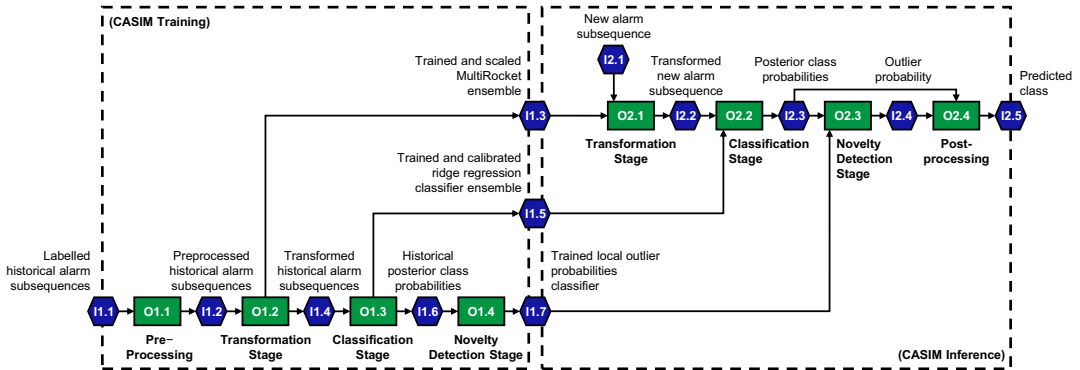


Figure 3. Formalized process description of the proposed convolutional kernel-based alarm subsequence identification method (CASIM).

phase. The training phase leverages historical ASs to generate and train the AFC model. Subsequently, the inference phase applies the trained model to new, unseen ASs, facilitating online analysis and classification. This methodology is consistent with conventional ML-based classification approaches (Lucke et al., 2019; Ruiz et al., 2021). In the following subsection, the specific components involved in each phase, including the process operators (green rectangles) and the processed information (blue hexagons), are described in detail.

3.2. Details of the proposed approach

CASIM begins with a preprocessing step (O1.1) using multivariate alarm series derived from labeled historical ASs (I1.1). These ASs are obtained through the alarm coactivation and event detection method (ACEDM) (Manca & Fay, 2021b), which identifies ASs in historical alarm data by detecting outliers in the time distances between alarms. This approach has proven to be superior to simple alarm activation rate thresholds (Manca & Fay, 2021b), for example, those described in Lucke et al. (2019) and Cheng et al. (2013). Subsequently, clustering algorithms, such as the alarm series similarity analysis method (ASSAM) (Manca et al., 2021) or convolutional kernel-based alarm subsequence transformation and clustering ensemble (CASTLE) (Manca et al., 2022b), are applied. These approaches can be used to group the detected ASs into clusters of similar ASs. For example, the ASSAM employs a TF-IDF-based AS similarity analysis.

CASIM utilizes ASs in the form of multivariate binary alarm series. Here, each alarm variable is associated with its own series, where an active alarm is denoted by 1, and an inactive alarm is denoted by 0, as shown in Figure 2a. If the alarm system does not directly provide multivariate binary alarm series, the alarm data representation can be generated from alarm activation and deactivation information (Lucke et al., 2019).

Next, we ensure that all ASs in I1.1 are of the same length. To achieve this, we zero pad shorter ASs, appending zero values to the end of the alarm series. AS length standardization is an integral part of the CASIM preprocessing and is designed to facilitate subsequent transformations in O1.2. Moreover, an essential assumption here is that conventional preprocessing steps, as outlined in Ahmed et al. (2013), Alinezhad et al. (2022a, 2022b, 2023), Charbonnier et al. (2015, 2016), Cheng et al. (2013), Dorgo et al. (2018), Fullen et al. (2018), Guo et al. (2017), Hu et al. (2016), Lai and Chen (2017), Lai et al. (2017, 2019), Lucke et al. (2019), Parvez et al. (2020, 2022), Shang and Chen (2019), and Zhou et al. (2022), which are designed to eliminate irrelevant alarms, are unnecessary. This is because our method has the capability to learn the most relevant alarm dynamics for recurrent abnormal situations from historical data, considering a variety of dynamic properties. Consequently, CASIM is less affected by irrelevant alarms. The specific details of this aspect are described in subsequent steps.

In the transformation stage, O1.2, the alarm series of the preprocessed historical alarm clusters (I1.2) are transformed into features that capture relevant alarm dynamics. There is a large body of research on multivariate time series transformation and classification. An overview of this topic is provided by Ruiz et al. (2021). Relevant methods include the hierarchical vote collective of transformation-based ensembles version 2.0 (HIVE-COTE 2.0) (Middlehurst et al., 2021), time series combination of heterogeneous and integrated embedding forest (TS-CHIEF) (Shifaz et al., 2020), InceptionTime (Fawaz et al., 2021), and Rocket (Dempster et al., 2020). Recently, a version of Rocket, minimally random convolutional kernel transform with multiple pooling operators and transformations (MultiRocket), was proposed by Tan et al. (2022). MultiRocket was found to achieve state-of-the-art classification accuracies while being considerably faster than other methods. In Tan et al. (2022), MultiRocket was trained approximately 350 times faster than InceptionTime, which uses deep convolutional neural networks. Due to these benefits, MultiRocket is well suited for AFC.

MultiRocket’s application to alarm data is shown in Figure 4. For each alarm variable in each AS in I1.2, MultiRocket employs two alarm series representations, namely, the original binary series, with $X^q = \{x_0^q, x_1^q, \dots, x_{l-1}^q\}$, where x_t^q is the alarm state of alarm variable q at time t , l is the length of the series, and a first-order difference representation of X^q given as (Tan et al., 2022):

$$\dot{X}^q = \{x_t^q - x_{t-1}^q : \forall t \in \{1, \dots, l-1\}\}. \tag{1}$$

Then, in a convolution operation, both X and \dot{X} of each alarm variable in an AS are transformed into a single feature vector \hat{X} of length $n_{\text{feat}} = n_{\text{kernel}} \times 8$, where the parameter n_{feat} directly determines the number of convolutional kernels n_{kernel} used in the transformation (Tan et al., 2022). Each kernel W_d has nine weights, $W = \{w_0, w_1, \dots, w_8\}$, with six weights of -1 and three weights of 2 , a dilation factor d in the range $\{[2^0], \dots, [2^{\log_2(n/8)}]\}$, where n is either l or $l-1$, a bias b , a padding option, where $n_{\text{kernel}}/2$ kernels utilize zero padding and the other $n_{\text{kernel}}/2$ kernels do not, and the assignment of a random set of selected alarm variables, K , with a size between 1 and 9 (Ruiz et al., 2021; Tan et al., 2022). Each W_d is

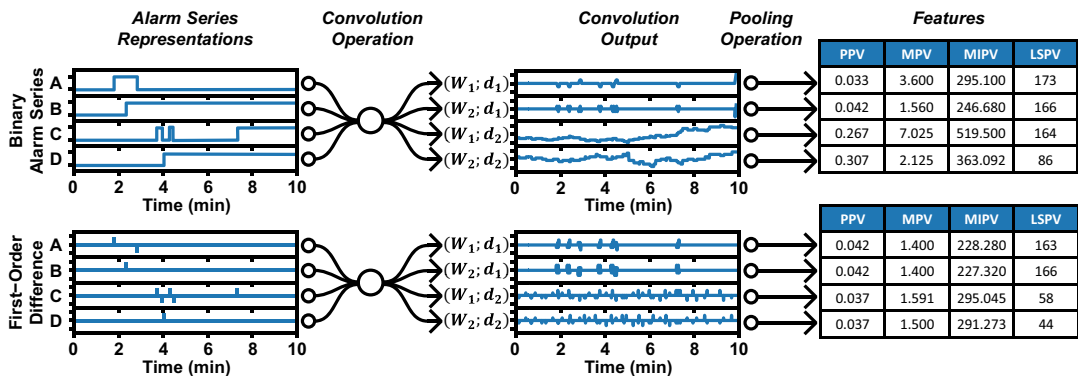


Figure 4. Application of MultiRocket to a binary alarm series and its first-order difference representation, both of which consist of alarm variables A to D and have a sampling rate of 1/s. The convolution operation employs two kernels $W_1 = [2, 2, -1, -1, 2, -1, -1, -1, -1]$ and $W_2 = [-1, -1, 2, 2, -1, -1, -1, -1, 2]$ in addition to two dilation factors $d_1 = [2^0]$ and $d_2 = [2^{\log_2(n/8)}]$ and a set of selected alarm variables $K = \{A, B, C, D\}$. The combination of kernels and dilation factors yields four convolution outputs per alarm series representation. All used kernels utilize zero padding. MultiRocket computes four features per convolution output using the bias $b = 0$ and four pooling operators: proportion of positive values (PPV), mean of positive values (MPV), mean of indices of positive values (MIPV), and longest stretch of positive values (LSPV).

then used to calculate two convolution outputs, Z , one for X and one for \hat{X} . For X , each $z_i \in Z$ is given by the following (Dempster et al., 2020):

$$z_i = \sum_{q \in K} x_i^q * W_d + b = \sum_{q \in K} \left(\sum_{j=0}^8 x_{i+(j \times d)}^q \times w_j \right) + b, \tag{2}$$

where $\forall i \in \{0, 1, \dots, n - 1\}$, $*$ symbolizes convolution and b is derived by randomly selecting a single AS from II.2 and calculating the quantiles of the respective $X * W_d$. A detailed description of the convolution operation can be found in Dempster et al. (2020). Figure 5 shows an illustration of the convolution operation applied to a generic univariate series without zero padding, Figure 5a, and with zero padding, Figure 5b. Three distinct dilation factors d are used to demonstrate how this parameter affects the spread of the kernel over the time series. After this operation, the initial convolution output z_0 with a bias value b of 0 is obtained, as depicted in Figure 5. Figure 4 illustrates how these outputs appear when computed for a multivariate alarm series, which includes time series data from more than one alarm variable.

Next, four different pooling operators are used to generate distinct features $\hat{x}_i \in \hat{X}$ from each Z . The proportion of positive values (PPV) (Tan et al., 2022) is given as:

$$PPV(Z) = (1/n) \sum_{i=0}^{n-1} [z_i > 0]. \tag{3}$$

The mean of positive values (MPV) (Tan et al., 2022) is as follows:

$$MPV(Z) = (1/m) \sum_{i=0}^{m-1} z_i^+, \tag{4}$$

where Z^+ is the vector of positive values in Z with length m . The mean of indices of positive values (MIPV) (Tan et al., 2022) is given as:

$$MIPV(Z) = \begin{cases} (1/m) \sum_{j=0}^{m-1} i_j^+ & \text{if } m > 0 \\ -1 & \text{otherwise,} \end{cases} \tag{5}$$

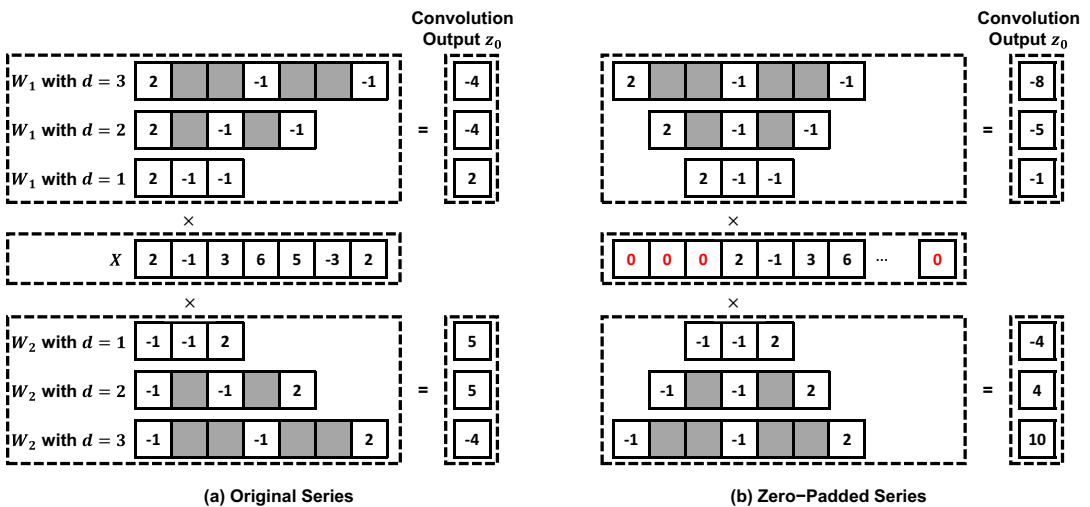


Figure 5. Convolution operation applied to a univariate series X of seven integer values. The convolution operation employs two kernels of length three, W_1 and W_2 , and three dilation factors d . The combination of kernels and dilation factors yields six convolution outputs z_0 , which are calculated by multiplying the respective kernel weights by the aligned series values and then summing the results. (a) Original series without zero padding. (b) Modified series with zero padding. Red zeros represent padded values.

where I^+ is the vector of indices of positive values in Z . The longest stretch of positive values (LSPV) is as follows (Tan et al., 2022):

$$\text{LSPV}(Z) = \max [j - i | \forall_{i \leq k \leq j} z_k > 0], \quad (6)$$

providing the maximum length of any subsequence that consists solely of positive values in the convolution output.

Figure 4 depicts an example of a pooling operation involving the four operators. The resulting features exhibit varying statistical properties; for instance, the relative PPV ranges between 0 and 1, whereas the MIPV and LSPV are absolute representations of the respective alarm variables' activation periods. To avoid a detrimental effect on the classification performance, we scale all features to unit variance, that is, we divide each value by the feature's standard deviation. We do not adjust for the mean of the features to preserve the sparsity structure of the transformed alarm data, that is, due to inactive alarm variables (Pedregosa et al., 2011).

MultiRocket is not fully deterministic (Middlehurst et al., 2021). Thus, we propose an ensemble approach to improve the classification performance by repeating the transformation stage O1.2 n_{clf} times. The trained MultiRocket instances and standard deviation of each feature $\hat{x}_i \in \hat{X}$ are then stored in I1.3 for future use in transforming new alarm data.

During the classification stage training in O1.3, the transformed historical AS clusters (I1.4), that is, feature vectors, are used to train an ensemble of linear ridge regression classifiers (I1.5), as recommended for MultiRocket by Tan et al. (2022). For each classifier, a set of n_{class} estimators is trained in a one-versus-rest approach, where n_{class} describes the number of classes $y_i \in Y$, that is, clusters, in I1.4. The regularization strength parameter, α , is tuned using leave-one-out cross-validation (Dempster et al., 2020). A description and examples of the ridge regression classifier can be found in the scikit-learn documentation (Pedregosa et al., 2011). The ridge regression classifier, however, does not provide any confidence level for the resulting classification (Middlehurst et al., 2021), which we require for CASIM's novelty detection stage in O1.4 and O2.3. Hence, we propose applying the multiclass case of Platt's probabilistic output, which is also used for ACM-SVM (Lucke et al., 2019), to the ridge regression classifier, as described in Lin et al. (2007) and Wu et al. (2004). Platt's probabilistic output calculates a calibrated estimate of the posterior class probabilities $p(Y|\hat{X})$ for each classified AS feature vector \hat{X} . The parameters of Platt's probabilistic output, γ and δ , are tuned when fitting its regression model in I1.4 (Lin et al., 2007).

To produce a single set of posterior class probabilities $p_i(Y|\hat{X}_i)$ for the i -th AS, where \hat{X}_i represents the n_{clf} feature vectors of the i -th AS, we merge the classifiers' outputs across all ensemble estimators as follows:

$$p_i(Y|\hat{X}_i) = (1/n_{\text{clf}}) \sum_{j=0}^{n_{\text{clf}}-1} p_i^j(Y|\hat{X}_i^j), \quad (7)$$

where $p_i^j(Y|\hat{X}_i^j)$ are the posterior class probabilities of all classes in Y for the j -th ensemble classifier. The label c_i^j of the most likely class for the i -th AS can then be determined as follows:

$$c_i^j = \arg \max_{j \in [1, \dots, n_{\text{class}}]} \{p_i(y_j|\hat{X}_i^j)\}. \quad (8)$$

In O1.4, CASIM's novelty detection stage is trained. To overcome the limitations described in Section 2, we propose using the novelty detection method local outlier probability (LoOP) presented in Kriegel et al. (2009). The choice of LoOP for this component of CASIM is rooted in its unique advantages, particularly its capability to provide intuitive and easily interpretable probabilistic assessments of whether an AS represents a novel class. This feature is especially beneficial for operators, including those with limited data science expertise, facilitating immediate and informed decision-making based on the likelihood of an AS being an outlier within a 0 to 1 probability range (Kriegel et al., 2009). Alternative outlier detection

methods, such as the local outlier factors (LOF) (Breunig et al., 2000), do not have a fixed range of 0 to 1 and their interpretation is not straightforward for the operator. This is because the range is influenced by the local density (Kriegel et al., 2009)—a common scenario in AFC where ASs can significantly differ in density depending on the complexity of the disturbance they represent. In the evaluation dataset in Section 4, we found that classes with short ASs are denser than clusters with complex disturbance propagations. An advantage of the LoOP in this context is its tolerance in handling varying densities within the data (Kriegel et al., 2009). This adaptability ensures that the LoOP can effectively detect novel AS classes, even in datasets with diverse alarm characteristics. By utilizing the LoOP, CASIM can more accurately differentiate between known and previously unobserved AS classes, thus improving overall AFC effectiveness.

Based on a set of clustered training samples, the LoOP computes the local density of the clusters by averaging the distances of the samples to their k -nearest neighbors. To allow for some variations in previously unobserved samples, an outlier is defined as deviating from the surrounding cluster’s density by more than λ times the standard deviation, assuming a Gaussian distribution. If a new sample lies outside this boundary, the outlier probability increases with the sample’s distance to its k -nearest neighbors (Kriegel et al., 2009). A detailed mathematical description of LoOP is given in Kriegel et al. (2009).

Rather than using only the posterior class probability of the most likely class, as in ACM-SVM (Lucke et al., 2019), we propose using a more comprehensive input to the LoOP, namely, a set of features containing the posterior class probabilities for all the known classes $p_i(Y|\hat{X}_i)$ and the difference $\Delta d_i^{1,2}$ between the probabilities of the most and second most likely classes, which can be computed as follows:

$$\Delta d_i^{1,2} = p_i(c_i^1|\hat{X}_i) - p_i(c_i^2|\hat{X}_i), \tag{9}$$

where c_i^2 is the label of the second most likely class:

$$c_i^2 = \arg \max_{j \in [1, \dots, n_{\text{class}}], y_j \neq c_i^1} \left\{ p_i(y_j|\hat{X}_i) \right\}. \tag{10}$$

A similar input was used for SVMs in another context in Schäfer and Leser (2020) and Gupta et al. (2020). Prior to generating the LoOP input, we first compute the posterior class probabilities $p(Y|\hat{X})$ of all historical ASs (II.6) using the historical AS feature vectors \hat{X} (II.4) and the ensemble of trained and calibrated ridge regression classifiers (II.5). The correctly classified historical ASs are then used to train a single LoOP classifier (II.7) with a parameter k_{LoOP} . The initial tests indicate the need to oversample classes that have fewer than $k_{\text{LoOP}} + 1$ correctly classified ASs to ensure that the classifier has a robust representation of the entire AS spectrum and does not suffer from data scarcity for certain classes, which could potentially lead to inaccurately high outlier probabilities for known classes (Manca & Fay, 2022).

To this end, we employ the synthetic minority oversampling technique (SMOTE) (Chawla et al., 2002) as our oversampling strategy, focusing directly on the generated posterior class probabilities (II.6) of correctly classified historical ASs. By using the SMOTE to integrate synthetic samples into the original distribution, we retain the intrinsic characteristics of known AS classes, ensuring that the synthetic data points do not distort the underlying probability structure (Chawla et al., 2002).

The SMOTE operates by selecting samples that are close in the feature space, drawing a line between the selected samples, and generating new synthetic points along that line. Specifically, for each minority class sample, the SMOTE traditionally selects k of its nearest neighbors within the same class (Chawla et al., 2002). However, in our adapted approach, for classes with a sample count less than $k_{\text{LoOP}}/2$, we use all available samples within the class for synthetic sample generation. For larger classes, we introduce a constraint, using $k_{\text{LoOP}}/2$ as an upper bound for the number of neighbors considered. After determining the set of neighbors, a random neighbor is then selected, and a synthetic sample is created at a random point between the two instances in the feature space. This process is repeated until each class consists of at least $k_{\text{LoOP}} + 1$ samples.

Given the alterations that the SMOTE can introduce to the class probability distributions, a renormalization step becomes necessary to adjust every synthetic sample's class probabilities so that their sum equals 1, in accordance with probability theory. Next, using (9) to append the difference $\Delta d_i^{1,2}$ to the resultant vectors, the LoOP classifier (I1.7) is trained, completing the *CASIM* training.

Online inference is made using *CASIM*'s transformation (O2.1), classification (O2.2), and novelty detection (O2.3) stages after a new AS is recorded (I2.1). I2.1 is zero-padded first if it is shorter than the ASs in I1.2. In O2.1, the new AS is transformed into a set of scaled feature vectors (I2.2) using the trained MultiRocket instances and feature standard deviations (I1.3), keeping the randomly assigned alarm variables K the same as those in O1.2. The vectors in I2.2 are then classified using the trained and calibrated ridge regression classifier ensemble (I1.5) in O2.2. The new AS posterior class probabilities (I2.3) are subsequently transformed into the proposed LoOP input vector, and in O2.3, the outlier probability p_{out} (I2.4) is determined using the trained LoOP classifier (I1.7).

Finally, *CASIM* concludes with a postprocessing step in O2.4, where p_{out} is compared to a threshold τ . If $p_{\text{out}} < \tau$, the label of the predicted class for the new AS (I2.5) is set to c^1 , that is, the one with the highest probability in the merged posterior class probabilities (I2.3); otherwise, I2.5 indicates a novel AS class, and $c^1 = -1$.

3.3. Classification of evolving alarm subsequences

In this section, we introduce the application of our proposed method *CASIM* to the classification of ASs as they evolve over time. While the fundamentals remain the same as described in Section 3.2, we will specifically focus on the necessary adaptations required to ensure that the method can effectively address dynamic and time-sensitive industrial scenarios.

We adapt concepts from the research field of early time series classification (ETSC), which focuses on predicting class labels for evolving time series data as quickly and accurately as possible (Gupta et al., 2020). Specifically, we employ the two-tier early and accurate series classifier (TEASER) method, presented by Schäfer and Leser (2020). We choose TEASER because it closely aligns with the framework used in our method, incorporating both a classification and a novelty detection stage. In TEASER, the novelty detection stage is utilized to determine if sufficient dynamics of a time series have been observed to confidently output the class label predicted by the classification stage. Furthermore, TEASER has been shown to outperform other state-of-the-art early time series classification approaches in terms of both accuracy and earliness (Bilski & Jastrzębska, 2023; Gupta et al., 2020; Schäfer & Leser, 2020; Kladis et al., 2021), making it a suitable choice for our adaptation of *CASIM*.

To classify evolving ASs, we employ the concept of segmenting historical ASs using an expanding window strategy and training multiple sets of respective stage instances for different window lengths as used in Schäfer and Leser (2020). This concept is illustrated in Figure 6, where Figure 6a demonstrates the expanding windows used for segmenting historical ASs, with $T(s_i)$ describing the window beginning at time 0 and ending at the i -th step $s_i = i \times w$, where w is the selected interval length. In addition, Figure 6b showcases how a set of stage instances is trained on a specific window $T(s_2)$, that is, the transformation (ts_2), classification (cs_2), and novelty detection stages (ds_2). This approach differs from the one employed in *ACM-SVM* (Lucke et al., 2019), where the classifier is trained on the entire window length and, as a result, might be unable to classify based on early dynamics because other characteristics that might emerge later could be still missing. We argue, that utilizing the proposed sequence of stages can be advantageous as it enables the learning of relevant and characteristic dynamics for smaller portions of the ASs, allowing for early identification.

Upon concluding the off-line training using historical ASs, the next step is to apply the trained stages for on-line identification of evolving ASs. To achieve this, we incrementally process incoming alarm data and detect ASs as they occur. While the scope of this paper does not encompass a detailed exploration of specific methodologies for online detection and segmentation of ASs, there are several established techniques in the literature that can be adapted for this purpose. In Lucke et al. (2019), Parvez et al. (2022), and Alinezhad et al. (2023), for instance, the alarm activation rate is calculated incrementally over

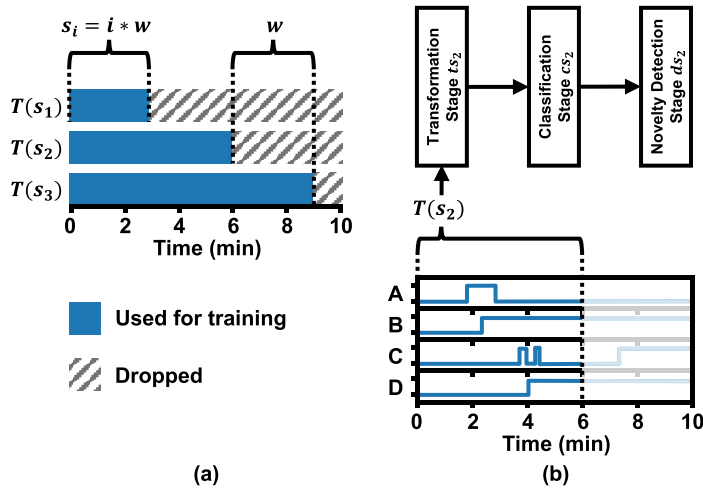


Figure 6. Concept of expanding windows for training the proposed CASIM for online classification of evolving alarm subsequences applied to an exemplary alarm series. With window T , step s , and interval length w . (a) Application of expanding windows to segment a historical alarm subsequence for off-line training. (b) Training a set of stage instances using the window $T(s_2)$ as input.

sliding windows to detect the beginning of an emerging AS. More broadly, in Manca and Fay (2021b), a comprehensive overview of various methods for detecting and segmenting ASs is provided.

Following the successful online detection of an emerging AS, the next step is its incremental identification using the trained stage instances of our proposed method as new data points are progressively added to the AS. The transformation (ts_i), classification (cs_i), and novelty detection stages (ds_i) are executed for each expanding window $T(s_i)$, leveraging the multiple sets of trained stage instances that correspond to different window lengths. This incremental approach enables our method to provide the human operator with a swift identification and classification of evolving ASs while also accurately detecting any novel patterns that may emerge in an industrial setting.

3.4. Limitations and advantages of the proposed approach

One limitation of the proposed CASIM arises from the calibration of the ridge regression classifier in O1.3. In fact, Platt’s probabilistic output assumes that the posterior class probabilities follow a sigmoid function (Lin et al., 2007). Nevertheless, this assumption has only been evaluated for SVMs (Lin et al., 2007). Due to the similarities between SVMs and ridge regression (Baesens et al., 2000), we argue that the calibration step in O1.3 applies to both approaches. A further limitation stems from the transformation stage utilized by CASIM, where MultiRocket’s convolution operation does not permit a straightforward interpretation of the resulting classifications, making it difficult for a human operator to comprehend the model’s prediction-making process (Kotriwala et al., 2021; Westin et al., 2016). Another limitation stems from the proposed ensemble approach in O1.2 and O1.3, which substantially increases the computational cost of CASIM depending on n_{clf} . However, (Middlehurst et al., 2021) demonstrated that using an ensemble approach enabled smaller values of n_{feat} , reduced the computational complexity of the transformation, and yielded a classification performance comparable to that of a single classifier instance with a considerably higher n_{feat} .

Nonetheless, the proposed CASIM shows advantages compared to relevant methods from the literature. With two different alarm series representations, diverse kernels, and four pooling operators, CASIM allows for the examination of not only a single characteristic, but a diverse range of dynamics arising from both alarm activation periods and alarm activation and deactivation events, thus capturing an AS’s alarm dynamics to a greater extent. Owing to the resulting diversity of the computed features, CASIM is less

affected by irrelevant alarm activations (*R1*) and a swapped order of alarms (*R2*). Moreover, the multivariate analysis of alarm variables is not limited to pairwise considerations, as in *ACM-SVM*, as the dynamic behavior of up to nine alarm variables is considered for a single feature. This allows for the identification of relationships between different alarm variables and extraction of meaningful patterns that would otherwise go unnoticed.

Furthermore, using the local density-based novelty detection method, the LoOP, in the novelty detection stage of *CASIM* instead of a simple nearest-neighbor distance or class probability threshold allows for a more sophisticated differentiation between the known and novel classes (*R3*). This is because our novelty detection stage more accurately captures the characteristics of alarm data, where different classes may exhibit varying densities. Furthermore, the utilization of techniques from ETSC in *CASIM*'s online AFC enhances its ability to focus on the specific dynamics and patterns that are relevant to the currently observed segment of the evolving AS to be classified. This is in contrast to other ML-based AFC methods that prioritize globally significant characteristics for distinguishing classes, which may not be relevant at the present moment.

4. Evaluation and discussion

In this section, we compare and analyze the performance and characteristics of two naive baselines, four AFC methods from Section 2, and *CASIM* from Section 3. In Section 4.1, the evaluation dataset is summarized, and in Section 4.2, the selection of appropriate evaluation measures is addressed. Section 4.3 describes the experimental setup. The results of the AFC evaluation are presented in Section 4.4.

4.1. Tennessee–Eastman process evaluation dataset

Hevner et al. (2004) discuss various systematic evaluation techniques used in information systems research. These include observational techniques, such as case studies, and experimental techniques, namely controlled experiments and simulations. Both case studies and controlled experiments utilize real plant data. Chioua et al. (2019), however, state that the task of obtaining appropriate alarm and process data from industrial plants is still a major obstacle in the development and assessment of alarm management methods. This is because industrial companies may have reservations, and alarm systems may perform inadequately, which hinders the use of advanced alarm analysis methods. Moreover, intentionally inducing faults and abnormal situations in industrial plants in order to generate authentic data could result in substantial harm to machinery, resources, individuals, and the surrounding environment. Hence, this study uses a dataset comprising artificial process and alarm data derived from a realistic simulation model of an industrial process. This model allows the replication of disturbances without posing any risk to equipment, products, or personnel.

For evaluation, we use the open access alarm management benchmark (Melo et al., 2022) dataset presented in Manca and Fay (2021b) and available in Manca (2020).³ The dataset is based on the MATLAB Simulink implementation of the Tennessee–Eastman process (TEP), which replicates the operations of a real plant owned by the Eastman Chemical Company in Tennessee, USA (Downs & Vogel, 1993). Since its initial publication in 1993, the TEP has gained recognition as a benchmark simulation model in the process automation of chemical plants (Arroyo, 2017; Bathelt et al., 2015; Melo et al., 2022). Its academic significance is maintained through various publications in process automation (Ricker, 1996), diagnosis (Arroyo, 2017), and alarm management (Shang & Chen, 2019; Alinezhad et al., 2022a, 2022b, 2023; Tamascelli et al., 2023).

Figure 7 shows the piping and instrumentation diagram (P&ID) of the TEP. The TEP features five process modules: a chemical reactor, a condenser, a vapor–liquid separator, a stripper, and a reboiler (Bathelt et al., 2015; Downs & Vogel, 1993). Furthermore, the process model includes 73 field sensors to measure and record process variables (Bathelt et al., 2015). Some of these measurements are used in the

³ <https://doi.org/10.21227/326k-qr90>.

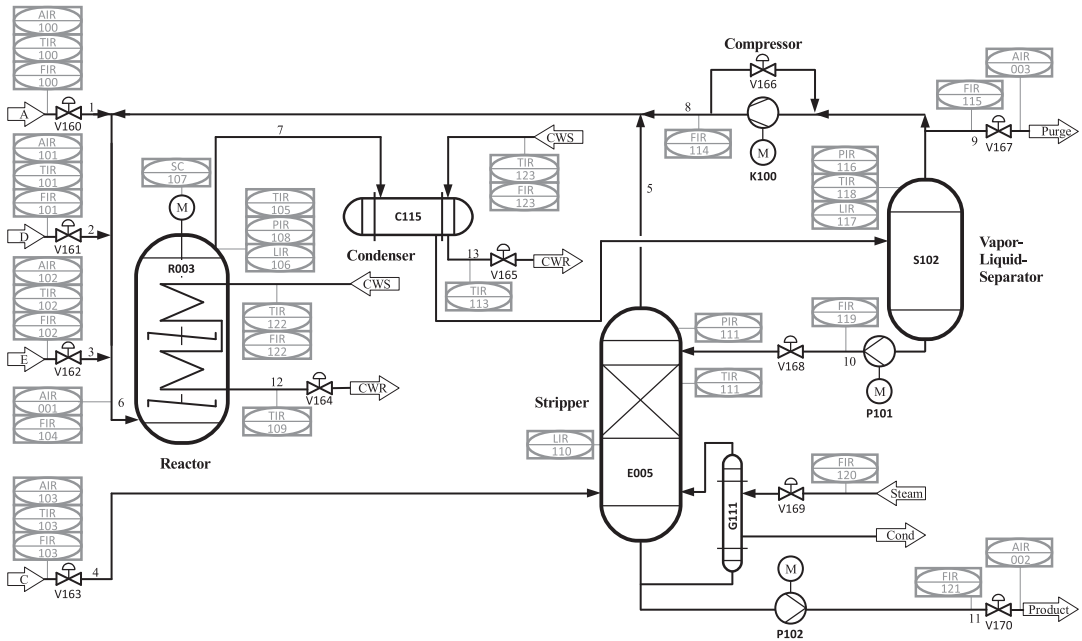


Figure 7. Piping and instrumentation diagram (P&ID) of the Tennessee-Eastman process (TEP) (Arroyo, 2017; Bathelt et al., 2015; Downs & Vogel, 1993; Manca & Fay, 2021b).

TEP’s 17 control loops that regulate 11 automated pneumatic control valves (Ricker, 1996). Detailed descriptions and visualizations of the process can be found in (Melo et al., 2022; Ricker, 1996), and (Bathelt et al., 2015).

Moreover, the dataset provides an alarm system for the TEP that defines 81 low-alarm and 81 high-alarm thresholds, as well as five high-high-alarm and three low-low-alarm thresholds (Manca & Fay, 2021b). These 170 alarm thresholds follow design recommendations given in industrial standards, that is, those in Takai et al. (2012) and EEMUA (2013). In addition, two alarm management techniques, an exponential weighted moving average filter and alarm deadbands, are implemented and parameterized, as described in Takai et al. (2012) and EEMUA (2013). As a result, there are no chattering alarms, that is, alarms that frequently toggle between alarm states (ASM Joint R&D Consortium, 2009; Takai et al., 2012; EEMUA, 2013) or alarms that are only briefly activated, also known as fleeting alarms (ASM Joint R&D Consortium, 2009; Takai et al., 2012; EEMUA, 2013).

The TEP dataset contains 100 simulation runs with 300 distinct abnormal situations and a total of 29 variations that differ in their respective disturbance duration, impact scaling, or disturbance combination (Manca et al., 2022b; Manca & Fay, 2021b). The eight different root cause disturbances that make up these abnormal situations include, for example, a step change that causes a loss in the material feed flow to the reactor or an increase in the chilled water supply temperature of the reactor. From the 300 situations, the TEP alarm system generates a total of 7343 alarm activations (Manca & Fay, 2021b). The number of alarm variables that are activated, the order in which alarm instances appear, and their dynamic behavior are all impacted by the considered variations as well as by random factors. Figure 1 provides an example of how, for two ASs originating from the same disturbance, the alarm order can differ significantly. Additional information can be found in the dataset’s technical report.

Figure 8 depicts a typical simulation run with three consecutive abnormal situations and an example subset of alarm series for 18 alarm variables (Manca et al., 2021). The first and second abnormal situations are caused by a step change in the amount of the process catalyst and reactants in one of the inlet feeds, respectively. The third abnormal situation, which rapidly escalates into an emergency shutdown, is caused by a complete blockage of the control valve for another inlet feed (Manca & Fay, 2021b).

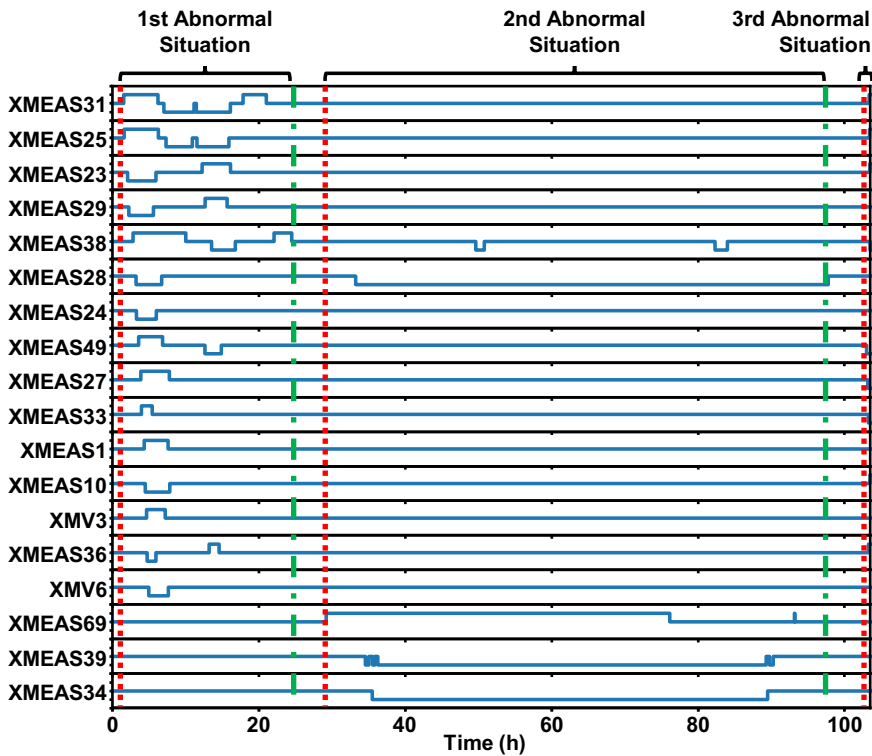


Figure 8. Three examples of consecutive abnormal situations. The solid blue lines represent the time trends of the alarm variables. The lower level for each alarm variable represents a low alarm, and the higher level represents a high alarm. The red dotted lines represent the initiation of a root cause disturbance. The green dashed-dotted lines represent the return to a normal operation (following Manca et al. (2021) and Manca and Fay (2021b)).

Because all disturbances are explicitly known and documented (Manca et al., 2021, 2022b), it is possible to compare the computed AFC results to a given ground-truth partition. Here, we apply the ACEDM⁴ (Manca & Fay, 2021a) and CASTLE⁵ (Manca et al., 2022a) to the TEP dataset according to Manca and Fay (2021b) and Manca et al. (2022b), respectively. As a result, 310 historical ASs are detected that are clustered in 14 distinct classes, and each class contains 3 to 48 similar ASs. Additionally, a cluster of 8 outliers ASs that contain only random components of the respective underlying abnormal situation and share few similarities with other ASs is found. The Appendix of Manca et al. (2022b) includes the labels for the resulting classes. To reduce the computational complexity of the subsequent evaluation steps, we resample the alarm data at a sampling rate of 1/min and select only those 76 alarm variables that are active at least once in the dataset.

Then, using stratified 5-fold cross-validation, we divide the 310 ASs into random sets, each containing a training and test split in which the relative frequency of the AS classes is maintained between the two splits. In addition, to evaluate the detection performance of novel AS classes, we repeat the cross-validation 14 times, each time excluding an entire class from the training splits and only utilizing it in the test splits. Consequently, 70 unique training-test sets are created.

Figure 9 uses t-distributed stochastic neighbor embedding (t-SNE) diagrams to illustrate the distribution and similarity of all 310 detected and clustered ASs. Two distance measures are applied: one measure

⁴<https://doi.org/10.24433/CO.9728090.v1>.

⁵<https://doi.org/10.24433/CO.9085464.v1>.

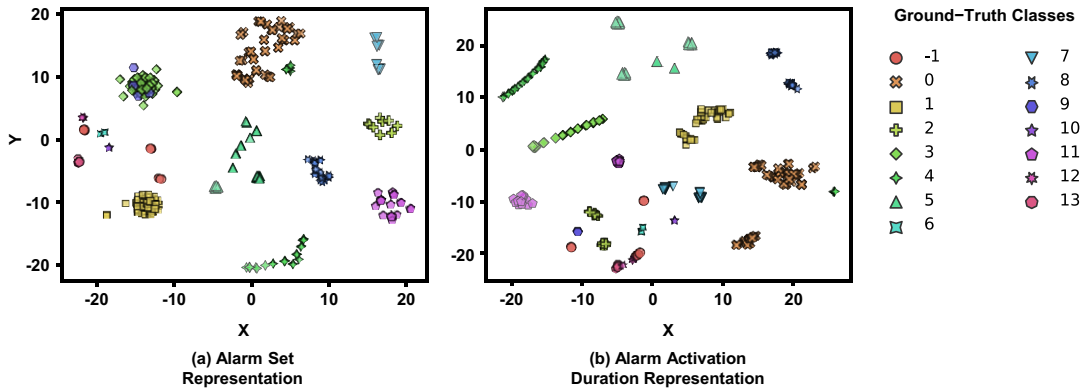


Figure 9. *t*-distributed stochastic neighbor embedding (*t*-SNE) representations of the 310 alarm subsequences in the Tennessee–Eastman process alarm management dataset presented in Manca and Fay (2021b). Each symbol represents a unique alarm subsequence, with the color coding and shape indicating the alarm subsequence’s class. The *t*-SNE representation employs two distinct alarm subsequence distance measures. (a) Alarm set-based Euclidean distances. (b) Alarm activation duration-based Euclidean distances.

focuses on the set of activated alarm variables (Fullen et al., 2018), as shown in Figure 9a, and the other focuses on the relative alarm variable activation duration (Manca et al., 2021), as illustrated in Figure 9b. The diagrams highlight classes such as nos. 3 and 9, which activate similar alarms but differ in their dynamics. Moreover, classes such as no. 0 exhibit substantial intraclass variances, while classes such as no. 1 demonstrate lower levels of variance. Fig. 9 further reveals that despite some classes being easily distinguishable, identifying new ASs as known or novel remains a challenge, especially when relying on a single alarm characteristic and facing closely related classes.

4.2. Classification performance metrics

We consider the actual condition of ASs belonging to the set of known classes as positive (P), whereas ASs from an unknown class are considered negative (N). A true positive (TP) is an AS from a known class whose predicted class c^1 matches the given ground-truth classification c^{true} . A true negative (TN) is an AS belonging to a previously unobserved class that has been classified as such, that is, $c^{true} = -1$. The classification performance is assessed based on three metrics, including the true positive rate (TPR), which is weighted equally across all classes in Y , the true negative rate (TNR), and the balanced accuracy (bACC). The TPR is given as follows:

$$TPR = (1/n_{class}) \sum_{i \in Y} (|TP_i|/|P_i|), \tag{11}$$

where $|x|$ describes the number of ASs in the respective test split with the condition x , P_i are ASs from known classes with $c^{true} = i$, and TP_i are ASs from known classes with $c^{true} = i$ and $c^1 = i$. The TNR is given as follows:

$$TNR = |TN|/|N|. \tag{12}$$

The bACC measures the trade-off between identifying known classes and detecting novel classes and is given as follows:

$$bACC = (TPR + TNR)/2. \tag{13}$$

4.3. Experimental setup

Two naive benchmarks are used: guess random class (*GRC*) and guess most common class (*GMC*). *GRC* randomly guesses the class of an AS to be classified based on the set of known classes. *GMC* guesses

based on the most common class in the respective training split. Both *GRC* and *GMC* lack a novelty detection stage; therefore, they are unable to distinguish between the known and novel classes. Furthermore, we select from the literature four existing AFC methods that cover the three different categories described in Section 2, that is, *WDI-INN*, *EAC-INN*, *MBW-LR*, and *ACM-SVM*. We compare the AFC performance of these four methods and the two naive benchmarks to those of our proposed *CASIM*.

Furthermore, *CASIM* is compared to a version that replaces the LoOP with the same novelty detection stage as that in *ACM-SVM*, namely, a novelty detection threshold on the posterior class probabilities (*CASIM-V1*). A second alternative version, *CASIM-V2*, implements *CASIM* as proposed in Manca and Fay (2022), that is, without the oversampling step proposed here. Additionally, a third version, *CASIM-V3*, is employed to assess the suggested expanding window strategy and sequence of stages for the online classification of evolving ASs as described in Section 3.3. *CASIM-V3* employs a single set of trained stages that utilize the entire historical ASs, identical to the implementation of *ACM-SVM* described by Lucke et al. (2019). This enables us to assess whether any significant local patterns can be acquired when training on various window lengths. This evaluation approach permits a comprehensive evaluation of *CASIM*'s effectiveness, as well as its components.

For *WDI-INN*, we implement the method according to the detailed description given in Charbonnier et al. (2016). According to Shang and Chen (2019), for the TEP, *EAC-INN*'s process-specific attenuation coefficient λ is set to 0.0667/min. Additionally, the normalized Euclidean distance is applied to *EAC-INN* to normalize the distances to the range 0 to 1, allowing for a more accurate comparison with other methods. For *MBW-LR*, the individual class thresholds are computed for each test based on the 95% confidence interval of the respective Gaussian distribution of training probability estimates (Alinezhad et al., 2023). The parameter θ for the binary logistic regression classifiers is obtained according to Jurafsky and Martin (2020) during training using a one-versus-rest strategy. For *ACM-SVM*, the parameters of Platt's probabilistic output to the SVMs, γ and δ , are tuned according to Lin et al. (2007). The parameters that undergo optimization through cross-validation are adjusted for each individual training-test set, rendering them dynamic and not explicitly defined in this context.

For the ridge regression classifiers of the proposed *CASIM*, a set of α values to test needs to be specified. Here, we use the default values provided by the Python implementation of MultiRocket (Tan et al., 2022), that is, an array of 10 values between 10^{-3} and 10^3 that are spaced evenly on a \log_{10} scale. Regarding n_{feat} , Tan et al. (2022) recommend using 50,000 features. The proposed ensemble approach, however, allows for fewer features. Thus, we set n_{feat} to the smallest possible number of 672 features (Tan et al., 2022), that is, $n_{\text{kernel}} = 84$. We also use the recommended 50,000 features ($n_{\text{kernel}} = 6,250$) and a reduced number of 10,000 features ($n_{\text{kernel}} = 1,250$) for comparison. Moreover, preliminary tests of the proposed ensemble approach suggest that 10 is a suitable setting for n_{clf} . However, we also test *CASIM* with n_{clf} values of 1, 5, and 25. In Middlehurst et al. (2021), $n_{\text{clf}} = 25$ was recommended for the related Rocket time series transformation. For LoOP, we set λ to 3 and k_{LoOP} to 10, according to the default setting described in Kriegel et al. (2009). To account for MultiRocket's remaining nondeterministic behavior, we repeat each test for *CASIM* 10 times and calculate the mean performance.

Moreover, except *MBW-LR*, the novelty detection stages of the examined AFC methods use different types of threshold parameters that share an admissible range of 0 to 1. These parameters, henceforth simply referred to as novelty detection thresholds, have a substantial influence on the method's detection trade-off between the known and novel classes (Charbonnier et al., 2016; Manca et al., 2021). In this paper, we consider novelty detection threshold settings for each method ranging from 0.001 to 1.000 with a step size of 0.001. This step size setting enables us to determine a suitable performance resolution, as preliminary tests have demonstrated that smaller step sizes do not significantly alter the performance dynamics of the methods examined here.

We further investigate the online classification capabilities of the analyzed AFC methods when confronted with evolving ASs. In our proposed method *CASIM* and its variant *CASIM-V3*, we employ an interval length ω of 10 minutes to ensure detailed performance resolution, with the initial window, denoted as $T(s_1)$, set at 10 minutes. This choice aligns with the common practice of assessing the presence of an alarm flood by evaluating the frequency of alarm activations within a 10-minute timeframe

(EEMUA, 2013; Manca & Fay, 2021b; Takai et al., 2012). Moreover, we extend our evaluation to AS lengths of 420 min, based on preliminary tests indicating that this duration results in a performance plateau for all examined AFC methods, akin to results obtained with full ASs.

For *ACM-SVM*, initially designed to be trained solely on the entire length of historical ASs (Lucke et al., 2019), we utilize both the original version and an alternative version, *ACM-SVM-V1*, which employs our proposed expanding window strategy. This comparison demonstrates the potential benefits of the expanding window strategy for other AFC methods. Additionally, we apply the expanding window strategy to *WDI-INN*, as it lacks specific guidance on handling evolving ASs (Charbonnier et al., 2015). Although references for *EAC-INN* and *MBW-LR* suggest incremental classification after each new alarm activation, our evaluation dataset features relatively low alarm rates (Manca & Fay, 2021b), making updates infrequent and accurate comparisons more difficult. Therefore, we adopt the same expanding window strategy for both *EAC-INN* and *MBW-LR* to ensure consistency in our evaluation methodology.

The computational experiments are conducted on a 64-bit Windows PC with an Intel(R) Core(TM) i7-7700HQ 2.80 GHz CPU and 16.0 GB memory. All AFC methods examined here are implemented in Python (3.9.7) using NumPy (Harris et al., 2020) (1.22.4), Pandas (McKinney, 2010) (1.5.1), sktime (0.13.4) (Löning et al., 2019), imbalanced-learn (0.11.0) (Lemaître et al., 2017), and Scikit-learn (Pedregosa et al., 2011) (1.1.3) as additional libraries. The implementation of our proposed *CASIM* and all other examined AFC methods is publicly available⁶ (Manca & Fay, 2023a).

4.4. Evaluation results

For each method, the TPRs, derived from applying their respective transformation and classification stages to all training-test sets in the TEP dataset, are visualized in Figure 10. As expected, naive guessing-based *GRC* and *GMC* achieve the lowest median values and overall classification performance.

The alarm set-based method *WDI-INN* achieves a median TPR of 0.923 and a TPR range of 0.154, which is narrower than that of other AFC methods. This result aligns with the theory presented by Charbonnier et al. (2015), which suggests that classifying ASs using sets of activated alarm variables might be more effective than using sequence alignment techniques. Nonetheless, *WDI-INN* has discernible limitations. For instance, this approach struggles to differentiate between ASs that share a common root cause but have significantly different disturbance propagation speeds. Such AS classes exhibit distinct alarm dynamics while sharing a common set of activated alarm variables.

Both alarm sequence-based methods, *EAC-INN* and *MBW-LR*, exhibit distinct performance differences. *EAC-INN* achieves a median TPR of 0.871, outperforming *MBW-LR*, which scores 0.769. This discrepancy may be due to *MBW-LR*'s TF-IDF representation, which considers alarm activation counts to differentiate between classes. However, in the TEP dataset, this count can vary significantly even within the same class. However, relying on the initial alarm activations in *EAC-INN* presents limitations as well, especially when discerning different escalation paths of an abnormal situation, as subsequent dynamics may reveal distinct characteristics.

ACM-SVM, which uses alarm series as its input, achieves a median TPR of 0.796. Upon careful examination of the evaluation results, it becomes apparent that there are certain challenges encountered in the classification of relatively short ASs that contain more than 19 active alarm variables. These challenges are particularly pronounced in cases where the ASs ultimately result in an emergency shutdown of the TEP. The presence of a significant quantity of coactive alarms within these ASs frequently leads to a high similarity, which consequently hinders *ACM-SVM*'s distinguishing capabilities.

Our proposed *CASIM* demonstrates superior performance in comparison to that of the existing methods that were evaluated, achieving the highest median TPR of 1.000 and a relatively narrow range of 0.083. Despite our method's promising results, there are instances where misclassification occurs. This happens when *CASIM* is unable to differentiate between various combinations of similar root cause disturbances.

⁶<https://doi.org/10.24433/CO.4874993.v1>.

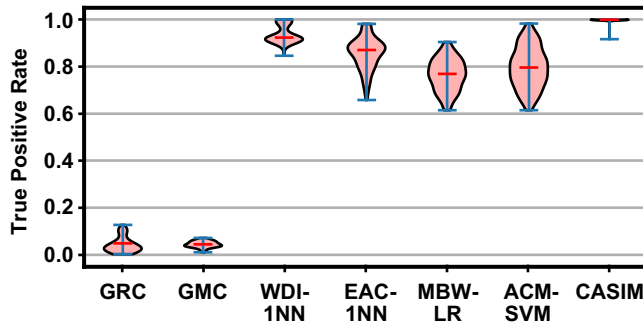


Figure 10. Violin plots that depict the true positive rate using the examined alarm flood classification methods across all tests. The median is represented by a red line. The range is depicted by two blue horizontal lines. The probability distribution is shown by the black boundary lines.

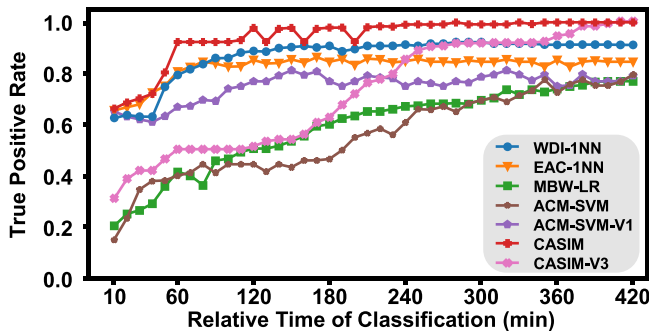


Figure 11. Median true positive rate of the examined alarm flood classification methods across all tests. The performance measurements were collected at relative time intervals post initiation of the corresponding alarm subsequences in the test.

Figure 11 provides insight into the online classification capabilities of the examined AFC methods when confronted with ASs that evolve over time. The essence of this evaluation is to comprehend these methods' ability to identify known AS classes when presented with truncated segments of the corresponding ASs, as opposed to their complete subsequences shown in Figure 10. Such an analysis considers scenarios requiring operator intervention during the evolution of an abnormal situation in which online support in the form of timely and accurate AFC might become critical (Alinezhad et al., 2023). As a result, Figure 11 depicts the median TPRs of the examined AFC methods across all tests and using different AS lengths.

In Figure 11, *WDI-1NN*, *EAC-1NN*, *ACM-SVM-V1*, and our *CASIM*, which all utilize an expanding window strategy, start strongly, with median TPR rates between 0.626 and 0.663 in the initial 10 minutes. This highlights their potential in capturing intrinsic alarm data characteristics with limited data available. A closer examination of the ASs and their associated ground-truth labels reveals certain classes having a temporal identity. Specifically, unique escalation paths that emerge later in the underlying abnormal situations can make early class differentiation challenging.

ACM-AVM and *CASIM-V3*, on the other hand, which are both trained on the full length of the historical training ASs, initially have lower TPR rates. However, they gradually improve and reach similar performance levels as their expanding window versions, *ACM-SVM-V1* and *CASIM*, respectively, between 350 and 380 min. These findings demonstrate the advantages of our suggested expanding window strategy in classifying evolving ASs by focusing on local dynamics.

WDI-1NN, *EAC-1NN*, *ACM-SVM-V1*, and *CASIM* reach peak performance between 60 and 120 min, with *CASIM* either outperforming or matching others. Together with *EAC-1NN*, *CASIM* swiftly achieves

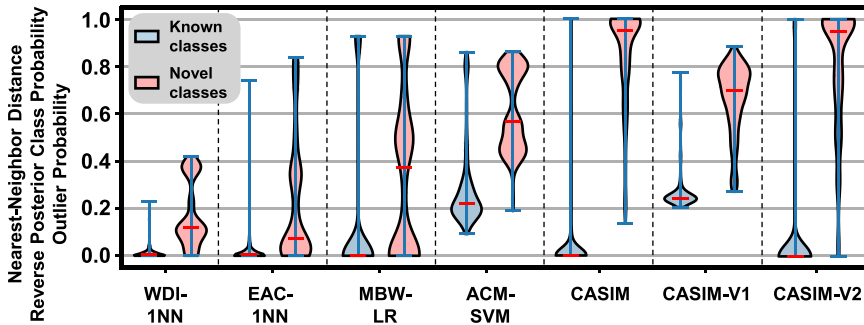


Figure 12. Violin plots that depict the nearest-neighbor distances (*WDI-1NN* and *EAC-1NN*), the reverse posterior class probabilities (*MBW-LR* and *ACM-SVM*), and the outlier probabilities (*CASIM*) over all tests. The distributions of alarm subsequences belonging to the known and novel classes are represented by blue and red shape fills, respectively. The median is represented by a red line. The range is depicted by two blue horizontal lines.

its peak within the first hour. In contrast, *MBW-LR*, which prioritizes alarm activation frequency through its TF-IDF approach, starts with a modest 0.204 TPR at 10 min. Its gradual climb to peak performance extends to 420 min. This performance can be attributed to the TEP dataset's challenges, as outlined in Manca et al. (2021), wherein the frequencies of alarm activations exhibit notable variations within AS classes.

Figure 12 depicts the distributions of nearest-neighbor distances and probabilities for all tests conducted on both the known and novel classes, utilizing the complete ASs. These values are employed by the novelty detection stages of the analyzed methods. For a more straightforward comparison, Figure 12 illustrates the reverse posterior class probabilities for *ACM-SVM*, *MBW-LR*, and *CASIM-V1*, that is, 1 minus the class probability of the most likely class. Having low values for ASs from known classes and high values for ASs from novel classes is advantageous for distinguishing between known and novel classes.

Upon examining the AFC methods illustrated in Figure 12, it becomes apparent that none of them are able to achieve perfect differentiation between known and novel classes. The presence of overlapping value ranges underscores the inherent trade-off associated with determining the threshold for novelty detection, leading to subsequent misclassifications. Notably, *WDI-1NN*, *EAC-1NN*, *MBW-LR*, and our *CASIM* demonstrate median values close to zero for known classes, suggesting that the corresponding ASs are accurately identified even when employing low threshold values. In contrast, *ACM-SVM* exhibits significantly higher values for known classes, with a median of 0.220 and a minimum of 0.093. This phenomenon can be attributed to the utilization of SVM's posterior class probabilities, which frequently produce lower values, particularly for similar AS classes.

Regarding novel classes, *WDI-1NN*'s low median distance of 0.119 emphasizes the challenges of using alarm sets alone for AFC. This implies that there is a tendency for alarm set-based interclass similarities to be relatively high. *EAC-1NN* and *MBW-LR*, employing alarm sequence representation, register medians of 0.075 and 0.374, respectively. While *ACM-SVM* records a median reverse posterior class probability of 0.566 for novel classes, our *CASIM* stands out with a median outlier probability of 0.952, indicating superior trade-offs among all methods. However, *CASIM* does present long tails for both class types, occasionally resulting in misclassifications during the novelty detection stage.

Furthermore, the probabilities of the alternative versions, *CASIM-V1* and *CASIM-V2*, of our proposed method *CASIM*, are also depicted in Figure 12. *CASIM-V1*, employing a novelty detection stage akin to *ACM-SVM*, that is, utilizing only the posterior class probabilities, demonstrates a narrower range of values for both known classes, with a median of 0.252, and novel classes, with a median of 0.706, when compared to *CASIM*. *CASIM-V2*, which excludes the suggested oversampling in the novelty detection stage of *CASIM*, exhibits an overall similar value distribution for both known and novel classes as *CASIM*.

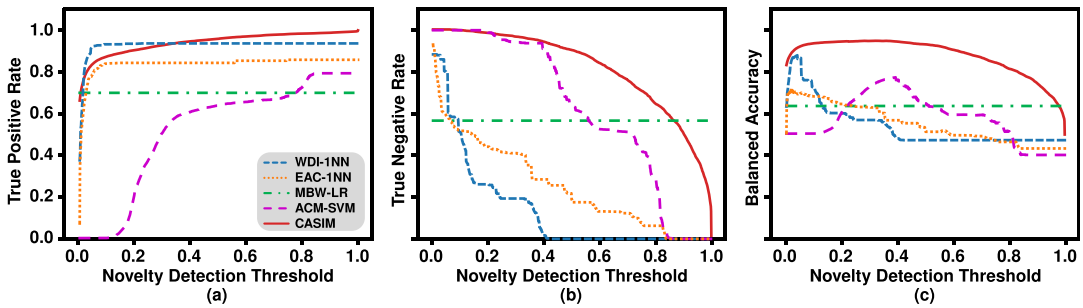


Figure 13. Average performance using the classification and detection stages of the four existing alarm flood classification methods and the proposed CASIM over all detection threshold parameter settings and tests. (a) True positive rate. (b) True negative rate. (c) Balanced accuracy.

However, there is a noticeable disparity in the mean values. The mean value for known classes is 0.046 for CASIM and 0.088 for CASIM-V2. The mean value of CASIM for novel classes is 0.850, while the mean value of CASIM-V2 is 0.798. This illustrates that the utilization of the suggested oversampling technique has the potential to enhance the distinction between known and novel classes.

Figure 13 presents the average classification performance for all examined AFC methods, derived from both their classification and novelty detection stages. Specifically, Figure 13a displays the classification and detection results of known classes via the TPR, Figure 13b illustrates the TNR associated with novel classes, and Figure 13c shows their trade-off through the bACC.

In the context of Figure 13, WDI-1NN performs well when dealing with known AS classes and initially demonstrates a notable ability to detect novel classes. Nonetheless, when the novelty detection threshold exceeds 0.052, the TNR decreases significantly. As shown in Figure 13c, WDI-1NN has a narrow dynamic range within which it excels in both classification tasks, peaking with a bACC of 0.875 for a novelty detection threshold of 0.039. Thus, inadequately calibrating WDI-1NN may prevent it from detecting new classes, limiting its applicability in industrial settings such as the TEP used here.

The characteristics of EAC-1NN result in TPRs with comparable dynamics to those of WDI-1NN, as shown in Figure 13a. However, due to the wider range of distance values for novel classes in Figure 12, EAC-1NN outperforms WDI-1NN in terms of TNRs for thresholds greater than 0.091, as shown in Figure 13b. The noticeable TNR plateaus are a direct result of EAC-1NN's distance accumulations, as shown in Figure 12. The trade-off between the TPR and TNR produces a bACC with a relatively high maximum value of 0.711 for a novelty detection threshold of 0.028, followed by a gradual decrease, with a mean average of 0.546, similar to that of WDI-1NN.

For MBW-LR, which does not require the user to define a novelty detection threshold, we observe the lowest maximum performance with respect to the TPRs and TNRs in Figure 13a and b, respectively. In fact, for the challenging TEP dataset used here, the class boundaries are sometimes set at high levels, such as 0.905 for class no. 12, thus making it difficult to detect novel classes if they share similarities with at least one previously known class. Nonetheless, MBW-LR's bACC performance is higher than that of all other examined AFC methods from the literature, making it a viable solution when parameter optimization on the novelty detection threshold is not possible.

Although ACM-SVM demonstrates commendable classification performance, as shown in Figure 10, the probability distribution for known classes in Figure 12 results in the most gradual rise in respective TPRs for ACM-SVM in Figure 13a. In contrast, ACM-SVM achieves relatively high TNR values for the majority of the possible novelty detection threshold values. As a result, in the classification trade-off depicted in Figure 13c, ACM-SVM outperforms WDI-1NN most of the time. Nonetheless, for a novelty detection threshold of 0.392, ACM-SVM achieves a lower peak bACC of 0.770.

For CASIM, the effects of the long tails present in Figure 12's probability distributions can be seen in Figure 13a and b, with the TPR increasing and then plateauing and the TNR decreasing gradually. For

novelty detection thresholds greater than 0.344 and 0.217, *CASIM* exhibits the highest TPRs and TNRs of all examined methods. For lower thresholds, *CASIM* provides at least the second-best TNR and TPR results among all examined methods. This finding is also supported by *CASIM* yielding a maximum bACC superior to that of all other examined methods, that is, a value of 0.947 for a novelty detection threshold of 0.324. Another notable phenomenon revealed in Figure 13c is that *CASIM* demonstrates an unmatched trade-off, with a mean average bACC of 0.879 over all considered novelty detection threshold values.

As shown in Figure 14a, the impact of MultiRocket’s nondeterministic properties on the performance of the proposed *CASIM* is investigated using 10 randomly instantiated instances of the latter. To provide a comprehensive understanding of this effect, we examine the bACC using the mean of all *CASIM* instances as well as two envelope curves: the interquartile range, which describes a range of performance values that the middle 50% of the *CASIM* instances fall within, and the range that encompasses all the *CASIM* instances. Intriguingly, Figure 14a reveals a narrow envelope for both ranges, especially for lower novelty detection threshold values, which slightly expands for higher thresholds. An in-depth analysis reveals that this phenomenon is primarily attributable to variations in the outlier probability values of ASs from novel classes, which in turn influence the TNR and, by extension, the bACC. Overall, MultiRocket’s non-deterministic components have a limited impact on *CASIM*.

To enable a systematic and in-depth examination of the efficacy of our proposed *CASIM*’s default parameter settings, we compare them with other recommended settings given in Middlehurst et al. (2021) and Tan et al. (2022). In Figure 14b, we compare various settings for the number of estimators in the ensemble, namely, n_{clf} values of 1, 5, 10, and 25, while maintaining the number of features n_{feat} at 672. It is demonstrated that different n_{clf} values have less of an effect, and the observed bACC exhibits a high level of consistency across the range of novelty detection thresholds examined.

In Figure 14c, we compare various settings for the number of features used in MultiRocket, specifically n_{feat} values of 672, 10,000, and 50,000, while keeping the ensemble’s number of estimators n_{clf} at 10. It is shown that the performance achieved when using the two higher settings is comparable. Interestingly, *CASIM*’s observed bACC is slightly better with a higher number of features. The reason for this, once again, lies in the detection of novel classes, where a higher n_{feat} may account for finding similarities that are either desirable or undesirable, depending on the ground truth against which the methods are evaluated.

In Figure 14d, we examine and compare the performance of the proposed *CASIM* with that of *CASIM-V1*, a *CASIM* variant that uses reverse posterior class probabilities rather than LoOP’s outlier probabilities in the novelty detection stage, and *CASIM-V2*, a variant that does not use the oversampling step in the novelty detection stage. It is demonstrated that incorporating the LoOP-based novelty detection stage and minority class oversampling in the otherwise imbalanced TEP dataset improves *CASIM*’s overall performance. *CASIM*’s bACC is greater than that of *CASIM-V1* and *CASIM-V2* for most of the detection thresholds considered.

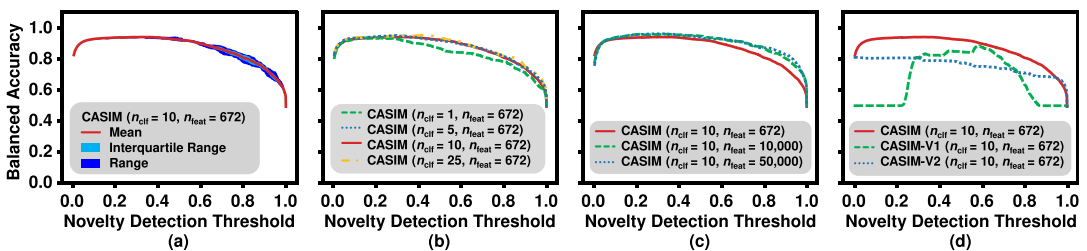


Figure 14. Balanced accuracy over all detection threshold parameter settings and tests using different parameter settings and versions of the proposed *CASIM*. (a) Ten randomly instantiated *CASIM* instances. (b) Different number of estimators n_{clf} . (c) Different numbers of features n_{feat} . (d) *CASIM*, *CASIM-V1*, and *CASIM-V2*.

5. Conclusions and outlook

The evaluation in Section 4 offers a comprehensive insight into the AFC methods examined in this study. Notably, *WDI-INN*, *EAC-INN*, *MBW-LR*, and *ACM-SVM* consistently outperform the naive baseline classifiers *GMC* and *GRC*. This underscores their ability to discern vital alarm characteristics and recognize recurring abnormal situations. However, these methods face some challenges with alarm order ambiguity (*R1*), irrelevant alarm activations (*R2*), and differentiation between known and novel AS classes (*R3*). Our evaluation dataset is particularly challenging, marked by significant variations in alarm order, extended intervals between alarm activations, and instances of both high intraclass variance and interclass similarity. In the context of this complex dataset, the examined methods were not able to capture the full spectrum of alarm dynamics inherent in historical ASs.

Section 4 demonstrates that our proposed *CASIM* effectively handles most of the dataset's intricacies, allowing this method to meet the specified requirements *R1* to *R3*. Notably, *CASIM* stands out by delivering superior classification performance and demonstrating a more favorable balance in identifying both known and novel AS classes relative to other AFC methods examined. These findings emphasize the merits of utilizing alarm series for multivariate time series transformation, classification, and novelty detection to enhance AFC outcomes. Furthermore, it is shown that the employment of an oversampling strategy to generate synthetic samples during the novelty detection phase significantly improves the detection efficacy. Furthermore, the implementation of an expanding window strategy on our proposed *CASIM* demonstrates a substantial enhancement in the online classification of ASs that evolve over time, allowing the AFC model to focus on the most relevant local dynamics during the training process. Therefore, we argue that *CASIM* holds distinct advantages for industrial AFC applications, where the dynamic nature of industrial processes necessitates a detailed and nuanced representation of diverse alarm dynamics and clear differentiation between novel and known classes is essential.

While our research has provided valuable insights into AFC methods and the potential of *CASIM*, it is imperative to recognize the constraints given the limited data available for this study. Alarm flood open access datasets are notoriously sparse (Chioua et al., 2019; Manca & Fay, 2021b), underscoring the need for additional benchmarks.

In this paper, parameter optimization mainly focuses on the novelty detection threshold, which serves as the primary parameter for balancing the differentiation between known and novel AS classes. Nevertheless, it is worth noting that, in this context, Section 4 demonstrates that the set of default settings for the remaining parameters in our proposed *CASIM* yields sufficient classification results. Furthermore, the carefully chosen parameter settings for *WDI-INN*, *EAC-INN*, *MBW-LR*, and *ACM-SVM* allow an examination of these methods' features in terms of their capacity to extract and learn meaningful and diverse alarm dynamics. Future research, however, can further evaluate the application of additional parameter optimization methods. Such efforts can possibly help to improve the performance of *CASIM* and make it applicable to a wide range of industrial processes. However, our findings illustrate that for our evaluation dataset's selected 76 alarm variables, a feature count of 672 is feasible for *CASIM*. Given that the recommended number of features is 50,000 (Tan et al., 2022), it is conceivable that our proposed method can represent approximately 5655 alarm variables with similar effectiveness.

While we have addressed the multifaceted dynamics of alarm floods with *CASIM*, the challenge of model interpretability remains due to the convolution operation employed by MultiRocket (Tan et al., 2022). To bridge this gap and enhance understanding, we have recently explored the domain of explainable artificial intelligence. Specifically, in Manca and Fay (2023b), we introduced a model-agnostic method that leverages counterfactual alarm floods. This approach holds the potential to offer clearer and more actionable insights into *CASIM*'s classification decisions, thus complementing our efforts to overcome interpretability limitations.

Abbreviations.

INN	first-nearest-neighbor
ACEDM	alarm coactivation and event detection method

ACM	alarm coactivation matrix
AFC	alarm flood classification
AI	artificial intelligence
AS	alarm subsequence
ASSAM	alarm series similarity analysis method
bACC	balanced accuracy
BLAST	basic local alignment search tool
CASIM	convolutional kernel-based alarm subsequence identification method
CASTLE	convolutional kernel-based alarm subsequence transformation and clustering ensemble
EAC	exponentially attenuated component
ETSC	early time series classification
GMM	Gaussian mixture model
HIVE-COTE 2.0	hierarchical vote collective of transformation-based ensembles version 2.0
LoOP	local outlier probability
LSPV	longest stretch of positive values
MAA	match-based accelerated alignment
MBW	modified bag-of-words
MIPV	mean of indices of positive values
ML	machine learning
MPV	mean of positive values
MultiRocket	minimally random convolutional kernel transform with multiple pooling operators and transformations
PPV	proportion of positive values
SMOTE	synthetic minority oversampling technique
SVM	support vector machine
TEP	Tennessee-Eastman process
TF-IDF	term frequency-inverse document frequency
TNR	true negative rate
TPR	true positive rate
TS-CHIEF	time series combination of heterogeneous and integrated embedding forest
t-SNE	t-distributed stochastic neighbor embedding
WDI	weighted dissimilarity index

Data availability statement. All data that supports the findings of this study are openly available and archived. The TEP dataset used in our evaluation, which includes a detailed technical report explaining the implemented alarm system, simulated abnormal situations, and the rationale behind them, is publicly archived and can be accessed via the following DOI hyperlink: <https://doi.org/10.21227/326k-qr90>. The Python implementation, along with the virtual environment setup and reproducible results for both ACEDM and CASTLE, is also made available to the public via the following DOI hyperlinks: <https://doi.org/10.24433/CO.9728090.v1> (ACEDM) and <https://doi.org/10.24433/CO.9085464.v1> (CASTLE). Furthermore, our proposed CASIM and all other examined AFC methods have been archived publicly. Included in the archive are the Python implementation, virtual environment setup, alarm flood data, ground-truth labels, and reproducible results for a 5-fold stratified cross-validation run, excluding open set classification. All these can be accessed via the following DOI hyper-link: <https://doi.org/10.24433/CO.4874993.v1>.

Acknowledgements. The project on which this article is based was funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS22030. The responsibility for the content of this publication lies with the authors.

Author contribution. Conceptualization: G.M.; A.F. Data curation: G.M. Investigation: G.M. Methodology: G.M. Resources: G.M. Software: G.M. Supervision: A.F. Validation: G.M. Visualization: G.M. Writing – original draft: G.M. Writing – review & editing: A.F.; G.M. All authors approved the final submitted draft.

Funding statement. The project on which this article is based was funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS22030. The responsibility for the content of this publication lies with the authors. The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interest. The authors declare none.

Ethical standard. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

- Ahmed K, Izadi I, Chen T, Joe D and Burton T (2013) Similarity analysis of industrial alarm flood data. *IEEE Transactions on Automation Science and Engineering* 10(2), 452–457. <https://doi.org/10.1109/tase.2012.2230627>.
- Alinezhad HS, Shang J and Chen T (2022a) Early classification of industrial alarm floods based on semisupervised learning. *IEEE Transactions on Industrial Informatics* 18(3), 1845–1853. <https://doi.org/10.1109/tii.2021.3081417>.
- Alinezhad HS, Shang J and Chen T (2022b) A modified bag-of-words representation for industrial alarm floods. In 2022 *IEEE international symposium on advanced control of industrial processes (AdCONIP)*. IEEE, pp. 331–336.
- Alinezhad HS, Shang J and Chen T (2023) Open set online classification of industrial alarm floods with alarm ranking. *IEEE Transactions on Instrumentation and Measurement* 72, 1–11. <https://doi.org/10.1109/tim.2022.3232617>.
- Arroyo E (2017) *Capturing and exploiting plant topology and process information as a basis to support engineering and operational activities in process plants* [Ph.D. dissertation] Helmut-Schmidt-University. <https://doi.org/10.24405/4189>.
- ASM Joint R&D Consortium (2009) *Effective alarm management practices*. ASM Consortium.
- Baesens B, Viaene S, Van Gestel T, Suykens J A K, Dedene G, De Moor B and Vanthienen J (2000). An empirical assessment of kernel type performance for least squares support vector machine classifiers. In *KES'2000. Fourth international conference on knowledge-based intelligent engineering systems and allied technologies. Proceedings (Cat. No.00TH8516)*. IEEE, pp. 313–316.
- Bathelt A, Ricker NL and Jelali M (2015) Revision of the Tennessee Eastman process model. *IFAC-PapersOnLine* 48(8), 309–314. <https://doi.org/10.1016/j.ifacol.2015.08.199>.
- Breunig MM, Kriegel HP, Ng RT and Sander J (2000) LOF: identifying density-based local outliers. In *SIGMOD'2000. 2000 ACM SIGMOD international conference on Management of data. Proceedings*. ACM, pp. 93–104. <https://doi.org/10.1145/335191.335388>.
- Bilski JM and Jastrzębska A (2023) CALIMERA: A new early time series classification method. *Information Processing & Management* 60(5), 103465. <https://doi.org/10.1016/j.ipm.2023.103465>.
- Charbonnier S, Bouchair N and Gayet P (2015) A weighted dissimilarity index to isolate faults during alarm floods. *Control Engineering Practice* 45, 110–122. <https://doi.org/10.1016/j.conengprac.2015.09.004>.
- Charbonnier S, Bouchair N and Gayet P (2016) Fault template extraction to assist operators during industrial alarm floods. *Engineering Applications of Artificial Intelligence* 50, 32–44. <https://doi.org/10.1016/j.engappai.2015.12.007>.
- Chawla NV, Bowyer KW, Hall LO and Kegelmeyer WP (2002) SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 321–357. <https://doi.org/10.1613/jair.953>.
- Cheng Y, Izadi I and Chen T (2013) Pattern matching of alarm flood sequences by a modified Smith–Waterman algorithm. *Chemical Engineering Research and Design* 91(6), 1085–1094. <https://doi.org/10.1016/j.cherd.2012.11.001>.
- Chioua M, Hollender M and Rodrigo V (2019) *Systematic alarm flood reduction*. Available at <https://doi.org/10.5281/zenodo.3530676> (accessed September 29, 2023).
- Dempster A, Petitjean F and Webb GI (2020) ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery* 34(5), 1454–1495. <https://doi.org/10.1007/s10618-020-00701-z>.
- Dorgo G, Pigler P and Abonyi J (2018) Understanding the importance of process alarms based on the analysis of deep recurrent neural networks trained for fault isolation. *Journal of Chemometrics* 32(4), e3006. <https://doi.org/10.1002/cem.3006>.
- Downs JJ and Vogel EF (1993) A plant-wide industrial process control problem. *Computers & Chemical Engineering* 17(3), 245–255. [https://doi.org/10.1016/0098-1354\(93\)80018-i](https://doi.org/10.1016/0098-1354(93)80018-i).
- EEMUA (2013) *Alarm systems, publication 191*. EEMUA.
- Fawaz HI, Lucas B, Forestier G, Pelletier C, Schmidt DF, Weber J, Webb GI, Idoumghar L, Muller P-A and Petitjean F (2021) InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery* 34(6), 1936–1962. <https://doi.org/10.1007/s10618-020-00710-y>.
- Fullen M, Schüller P and Niggemann O (2018) Validation of similarity measures for industrial alarm flood analysis. In Niggemann O and Schüller P (eds), *IMPROVE - Innovative modelling approaches for production systems to raise validatable efficiency*. Springer, pp. 93–109. https://doi.org/10.1007/978-3-662-57805-6_6.
- Guo C, Hu W, Lai S, Yang F and Chen T (2017) An accelerated alignment method for analyzing time sequences of industrial alarm floods. *Journal of Process Control* 57, 102–115. <https://doi.org/10.1016/j.jprocont.2017.06.019>.
- Gupta A, Gupta HP, Biswas B and Dutta T (2020) Approaches and applications of early classification of time series: A review. *IEEE Transactions on Artificial Intelligence* 1(1), 47–61. <https://doi.org/10.1109/taai.2020.3027279>.
- Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen D, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, Del Río JF, Wiebe M, Peterson P, ... Oliphant TE (2020) Array programming with NumPy. *Nature* 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>.
- Hevner AR, March ST, Park J and Ram S (2004) Design science in information systems research. *MIS Quart.* 28(1), 75–105. <https://doi.org/10.5555/2017212.2017217>.
- Hu W, Wang J and Chen T (2016) A local alignment approach to similarity analysis of industrial alarm flood sequences. *Control Engineering Practice* 55, 13–25. <https://doi.org/10.1016/j.conengprac.2016.05.021>.

- Jurafsky D and Martin JH** (2020) Logistic regression. In Jurafsky D and Martin JH (eds), *Speech and language processing*. Stanford University, pp. 75–95.
- Kladis E, Akasiadis C, Michelioudakis E, Alevizos E, & Artikis A** (2021). An empirical evaluation of early time-series classification algorithms. In *24th International Conference on Database Theory*. Available at https://ceur-ws.org/Vol-2841/SIMPLIFY_6.pdf (accessed April 13, 2024).
- Klopper B, Dix M, Siddapura D and Taverne LT** (2016). Integrated search for heterogeneous data in process industry applications — A proof of concept. In *2016 IEEE 14th international conference on industrial informatics (INDIN)*. IEEE, pp. 1306–1311. <https://doi.org/10.1109/INDIN.2016.7819369>.
- Kotriwala AM, Klöpper B, Dix M, Gopalakrishnan G, Ziobro D and Potschka A** (2021). XAI for operations in the process industry. In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle and F.V. Harmelen (Eds.), *Proceedings of the AAAI 2021 spring symposium on combining machine learning and knowledge engineering (AAAI-MAKE 2021)*. Available at <https://ceur-ws.org/Vol-2846/paper26.pdf> (accessed November 12, 2023).
- Kriegel H-P, Kröger P, Schubert E and Zimek A** (2009). LoOP. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, pp. 1649–1652. <https://doi.org/10.1145/1645953.1646195>.
- Lai S and Chen T** (2017) A method for pattern mining in multiple alarm flood sequences. *Chemical Engineering Research and Design* 117, 831–839. <https://doi.org/10.1016/j.cherd.2015.06.019>.
- Lai S, Yang F and Chen T** (2017) Online pattern matching and prediction of incoming alarm floods. *Journal of Process Control* 56, 69–78. <https://doi.org/10.1016/j.jprocont.2017.01.003>.
- Lai S, Yang F, Chen T and Cao L** (2019) Accelerated multiple alarm flood sequence alignment for abnormality pattern mining. *Journal of Process Control* 82, 44–57. <https://doi.org/10.1016/j.jprocont.2019.06.004>.
- Lemaître G, Nogueira F and Aridas CK** (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17), 1–5. Available at <https://jmlr.org/papers/volume18/16-365/16-365.pdf> (accessed November 12, 2023).
- Lin H-T, Lin C-J and Weng RC** (2007) A note on Platt’s probabilistic outputs for support vector machines. *Machine Learning* 68(3), 267–276. <https://doi.org/10.1007/s10994-007-5018-6>.
- Löning M, Bagnall A, Ganesh S, Kazakov V, Lines J and Király F** (2019). Sktime: A unified interface for machine learning with time series. In *Workshop on Systems for ML at the 33rd conference on neural information processing systems (NeurIPS 2019)*, pp. 1–10. Available at http://learningsys.org/neurips19/assets/papers/sktime_ml_systems_neurips2019.pdf (accessed November 12, 2023).
- Lucke M, Chioua M, Grimholt C, Hollender M and Thornhill NF** (2019) Advances in alarm data analysis with a practical application to online alarm flood classification. *Journal of Process Control* 79, 56–71. <https://doi.org/10.1016/j.jprocont.2019.04.010>.
- Manca G** (2020) “Tennessee-Eastman-Process” alarm management dataset. IEEE Dataport. <https://doi.org/10.21227/3226k-qr90>.
- Manca G, Dix M and Fay A** (2021) Clustering of similar historical alarm subsequences in industrial control systems using alarm series and characteristic coactivations. *IEEE Access* 9, 154965–154974. <https://doi.org/10.1109/access.2021.3128695>.
- Manca G, Dix M and Fay A** (2022a). Convolutional kernel-based alarm subsequence transformation and clustering ensemble (CASTLE). *Code Ocean*. <https://doi.org/10.24433/CO.9085464.v1>.
- Manca G, Dix M and Fay A** (2022b). Convolutional kernel-based transformation and clustering of similar industrial alarm floods. In *2022 IEEE eighth international conference on big data computing service and applications (BigDataService)*. IEEE, pp. 161–166. <https://doi.org/10.1109/BigDataService55688.2022.00033>.
- Manca G and Fay A** (2021a). Alarm coactivation and event detection method (ACEDM). *Code Ocean*. <https://doi.org/10.24433/CO.9728090.v1>.
- Manca G and Fay A** (2021b) Detection of historical alarm subsequences using alarm events and a coactivation constraint. *IEEE Access* 9, 46851–46873. <https://doi.org/10.1109/access.2021.3067837>.
- Manca G and Fay A** (2022). Identification of industrial alarm floods using time series classification and novelty detection. In *2022 IEEE 20th international conference on industrial informatics (INDIN)*. IEEE, pp. 698–705. <https://doi.org/10.1109/INDIN51773.2022.9976139>
- Manca G and Fay A** (2023a). Convolutional kernel-based alarm subsequence identification method (CASIM). *Code Ocean*. <https://doi.org/10.24433/CO.4874993.v1>
- Manca G and Fay A** (2023b). Explainable AI for industrial alarm flood classification using counterfactuals. In *49th annual conference IEEE IES*. IEEE.
- McKinney W** (2010). Data structures for statistical computing in python. In *Proceedings of the python in science conference*. SciPy, pp. 51–56. <https://doi.org/10.25080/Majora-92bf1922-00a>.
- Melo A, Câmara MM, Clavijo N and Pinto JC** (2022) Open benchmarks for assessment of process monitoring and fault diagnosis techniques: A review and critical analysis. *Computers & Chemical Engineering* 165, 107964. <https://doi.org/10.1016/j.compchemeng.2022.107964>.
- Middlehurst M, Large J, Flynn M, Lines J, Bostrom A and Bagnall A** (2021) HIVE-COTE 2.0: A new meta ensemble for time series classification. *Machine Learning* 110(11–12), 3211–3243. <https://doi.org/10.1007/s10994-021-06057-9>.
- Mustafa FE, Ahmed I, Basit A, Alvi U-EH, Malik SH, Mahmood A and Ali PR** (2023) A review on effective alarm management systems for industrial process control: Barriers and opportunities. *International Journal of Critical Infrastructure Protection* 41, 100599. <https://doi.org/10.1016/j.ijcip.2023.100599>.

- Parvez MR, Hu W and Chen T** (2020). Comparison of the Smith-Waterman and Needleman-Wunsch algorithms for online similarity analysis of industrial alarm floods. In *2020 IEEE electric power and energy conference (EPEC)*. IEEE, pp. 1–6. <https://doi.org/10.1109/EPEC48502.2020.9320080>
- Parvez MR, Hu W and Chen T** (2022) Real-time pattern matching and ranking for early prediction of industrial alarm floods. *Control Engineering Practice* 120, 105004. <https://doi.org/10.1016/j.conengprac.2021.105004>.
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M and Duchesnay É** (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830. Available at <https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf> (accessed November 12, 2023).
- Ricker NL** (1996) Decentralized control of the Tennessee Eastman challenge process. *Journal of Process Control* 6(4), 205–221. [https://doi.org/10.1016/0959-1524\(96\)00031-5](https://doi.org/10.1016/0959-1524(96)00031-5).
- Rodrigo V, Chioua M, Hagglund T and Hollender M** (2016) Causal analysis for alarm flood reduction. *IFAC-PapersOnLine* 49 (7), 723–728. <https://doi.org/10.1016/j.ifacol.2016.07.269>.
- Ruiz AP, Flynn M, Large J, Middlehurst M and Bagnall A** (2021) The great multivariate time series classification bake off: A review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* 35(2), 401–449. <https://doi.org/10.1007/s10618-020-00727-3>.
- Schäfer P and Leser U** (2020) TEASER: Early and accurate time series classification. *Data Mining and Knowledge Discovery* 34 (5), 1336–1362. <https://doi.org/10.1007/s10618-020-00690-z>.
- Shang J and Chen T** (2019) Early classification of alarm floods via exponentially attenuated component analysis. *IEEE Transactions on Industrial Electronics* 67(10), 8702–8712. <https://doi.org/10.1109/tie.2019.2949542>.
- Shifaz A, Pelletier C, Petitjean F and Webb GI** (2020) TS-CHIEF: A scalable and accurate forest algorithm for time series classification. *Data Mining and Knowledge Discovery* 34(3), 742–775. <https://doi.org/10.1007/s10618-020-00679-8>.
- Tamascelli N, Rao HRM, Cozzani V, Paltrinieri N and Chen T** (2023). Online classification of alarm floods using a Word2vec algorithm. In *49th Annual Conf. IEEE Ind. Electron. Society (IECON)*. IEEE, pp. 1–6. <https://doi.org/10.1109/IECON51785.2023.10312435>.
- Takai T, Kutsuma Y and Ishihara H** (2012). Management of alarm systems for the process industries. In *2012 proceedings of SICE annual conference (SICE)*. IEEE, pp. 688–692.
- Tan CW, Dempster A, Bergmeir C and Webb GI** (2022) MultiRocket: Multiple pooling operators and transformations for fast and effective time series classification. *Data Mining and Knowledge Discovery* 36(5), 1623–1646. <https://doi.org/10.1007/s10618-022-00844-1>.
- VDI/VDE (2015) *Formalised process descriptions, publication 3682*. VDI/VDE.
- Vogel-Heuser B, Schütz D and Folmer J** (2015) Criteria-based alarm flood pattern recognition using historical data from automated production systems (aPS). *Mechatronics* 31, 89–100. <https://doi.org/10.1016/j.mechatronics.2015.02.004>.
- Wang J, Yang F, Chen T and Shah SL** (2016) 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. <https://doi.org/10.1109/tase.2015.2464234>.
- Westin C, Borst C and Hilburn B** (2016) Strategic conformance: Overcoming acceptance issues of decision aiding automation? *IEEE Transactions on Human-Machine Systems* 46(1), 41–52. <https://doi.org/10.1109/thms.2015.2482480>.
- Wu TF, Lin CJ and Weng RC** (2004) Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research* 5, 975–1005. <https://doi.org/10.1007/s10994-016-5600-x>.
- Zhou B, Hu W and Chen T** (2022) Pattern extraction from industrial alarm flood sequences by a modified CloFAST algorithm. *IEEE Transactions on Industrial Informatics* 18(1), 288–296. <https://doi.org/10.1109/tii.2021.3071361>.