Clustering of Similar Historical Alarm Subsequences Using Alarm Series and Characteristic Coactivations

Gianluca Manca[1], **Marcel Dix**[2] and **Alexander Fay**[1]
[1]Institute of Automation Technology, Helmut-Schmidt-University, Hamburg, Germany
email: gianluca.manca89@gmail.com, alexander.fay@hsu-hh.de
[2]Industrial Data Analytics, ABB Corporate Research Center, Ladenburg, Germany
email: marcel.dix@de.abb.com

Abstract

Alarm flood similarity analysis (AFSA) methods are frequently used as a primary step for root-cause analysis, alarm flood pattern mining, and, eventually, online operator support. AFSA methods have been promoted in several research activities in recent years. However, addressing an often-observed ambiguity of the order of alarms and the annunciation of irrelevant alarms in otherwise similar alarm subsequences remains a challenging task. To address and solve these limitations, this paper presents a novel AFSA method that uses alarm series as input to two extended term frequency-inverse document frequency (TF-IDF)-based clustering approaches and a novel outlier validation. The method proposed here utilizes both characteristic alarm variables and their coactivations, thus, emphasizing the dynamic properties of alarms to a greater extent. Our method is compared to three relevant methods from the literature. The effectiveness and performance of the examined methods are illustrated by means of an openly accessible dataset based on the "Tennessee-Eastman-Process". It is shown that the integration of alarm series data improves the overall performance and robustness of the AFSA. Furthermore, the clustering results are less influenced by the ambiguity of the order of alarms and irrelevant alarms, thus overcoming a persistent challenge in alarm management research.

1 Introduction

Driven by the advances in automation technologies, industrial process plants have become very data intense. A typical process plant, like a chemical plant or oil refinery, involves a vast amount of data. Typical data types are time series readings from sensors, alarm and event logs, electronic operator shift books, laboratory results (e.g., from samples taken during production), maintenance and repair reports, and more. The amount of data being processed and stored in plants today can easily sum up to several hundreds of gigabytes every year [14]. All this data provides a great potential for artificial intelligence (AI) and machine learning, to better understand plant behavior and thereby take better operator decisions.

A valuable type of data for industrial analytics and machine learning is the alarm data, because of the high importance of alarms, which is to warn operators about critical process deviations. Here, additional insights that machine learning could provide about the alarm data could potentially help operators in better understanding various alarm situations, and thereby operating process plants more effectively and safely.

In process control systems, alarms are raised when a predefined critical threshold value at a field sensor (e.g., a pressure, flow, level, or temperature sensor) is exceeded. As process plants can have hundreds of sensors [14], there can be hundreds of different types of alarms that can occur. Ideally, the number of alarms raised at a time should be as low as possible, so that the number of alarms is still manageable by a human. The industry norm [2] recommends that this number should be less than 10 alarms in 10 minutes. However, in more anomalous situations there can be a much larger number of alarms, that becomes more difficult to handle, which is a known challenge in the industry and literature, typically referred to as alarm showers or floods [2].

In situations of alarm floods a simple sequential handling of alarms may not be the most practical approach, due to the limited time available to resolve the overall (critical) plant situation, but also due to the fact that the various alarms cannot be handled in isolation but have dependencies. In many cases, alarms of an alarm flood were triggered by the same root-cause [11]. Operators are aware of this and try to understand the "bigger picture" of the overall alarm situation, to be able to respond to this situation more effectively and quickly. This, however, requires a lot of experience from operators and is a demanding task at high time pressure and responsibility. Here, an interesting use case for data analytics and machine learning in the industrial domain is to help plant operators in extracting the implicit knowledge about different alarm situations more automatically. Such an AI-based operator support function could potentially save the operator a lot of time in the decision-making process, where a manual assessment of complex alarm situations, such as alarm floods, can be time consuming.

Driven by the demographic change of society, an additional challenge seems to be emerging, that a lot of experience may get lost from operator rooms. Following generations of young and inexperienced operators will need to relearn this knowledge again, how to correctly assess and respond to different plant situations [7]. The use of AI is one

possible approach that may help to capture the implicit knowledge about historic alarm situations, to prevent it from getting lost. Moreover, there is opportunity that such an automatic learning could save time for a human having to learn some of the non-obvious rules and patterns from experience over years. Based on a learned model, AI-based operator support functions could be imagined as part of the process control system, that could explain to the (inexperienced) operator a recurrent alarm flood.

There is already a good body of research, that seeks to provide data-driven solutions for analyzing similar and recurrent industrial alarm floods, which will be presented in detail in the following "Related Work" section. In this article a novel solution approach is presented for the analysis and clustering of similar alarm floods, that makes use of insights about the dynamic properties of alarms and their coactivations. The proposed approach was compared to existing methods using an openly accessible alarm management dataset based on the "Tennessee-Eastman-Process" (TEP) [3][5] that includes 300 abnormal alarm situations [16], and the results from this evaluation are presented in this article. It turns out that the proposed approach leads to more accurate and meaningful clusters than if having left the intrinsic knowledge about the dynamic structure of the alarm sequences in the data unconsidered.

This study seeks to contribute to AI researchers from academia and industrial practitioners, by proposing a feasible solution for the extraction of implicit knowledge about recurrent alarm flood situations and their dynamics through alarm data clustering, which could be the basis for the development of novel AI-based operator support functions in process control systems.

The rest of the paper is organized as follows: Section 2 describes and analyzes the state-of-the-art methods for the analysis of similar alarm floods regarding existing requirements and limitations. Section 3 describes the development of a novel approach based on the findings from the related work. In Section 4 an in-depth evaluation and comparison of the methods in Sections 2 and 3 is conducted. Finally, this paper concludes with a comprehensive discussion of the evaluation results and an outlook on potential future work in Section 5.

2 Related Work

A comprehensive overview of the existing alarm data analysis approaches is given in [15]. One major branch is alarm flood similarity analysis (AFSA) methods, which detect and group recurrent historical alarm flood situations or, more generally, alarm subsequences (ASs) [15]. Here, ASs are smaller partitions of an original alarm sequence [1][22]. The task of grouping similar historical ASs allows for the collection of different variants of otherwise similar abnormal situations, which can improve further analysis steps [6]. Most commonly, AFSA methods are used for alarm rationalization or to generate the input for advanced alarm analysis methods [15]. For example, in [6], [9], and [18], clusters of similar ASs are subject to a causal analysis to detect common root-cause disturbances. This information can then be used online to support the operator with suggestions regarding the most likely root-cause disturbance of a recurring AS [9]. Reference [4] defined two requirements (R1 and R2) regarding the similarity analysis of ASs:

- R1: A suitable method should tolerate irrelevant alarms annunciated in some ASs.
- R2: A suitable method should tolerate a swapped order of alarm activations (ACTs) in otherwise similar ASs.

One category of AFSA approaches applies "frequent pattern mining" (FPM) methods to sequences of ACTs. For example, [7] and [22] use FPM to detect the most relevant combinations of alarms in historical alarm data. However, these methods are restricted to alarm clusters that have minimum support in the data, i.e., either the absolute or relative frequency, and thus, they show limitations when an abnormal situation is uncommon.

Another category that is promoted in several research activities is the pairwise alignment of ASs. For this purpose, [1] proposed a global sequence alignment method using the dynamic time warping (DTW) algorithm to detect common alarm patterns. Prior to that, a prefiltering step groups potentially similar ASs according to the Jaccard-distance of AS pairs (s. (7)). However, DTW does not tolerate any ambiguity of order in otherwise similar ASs. This challenging task was to some extent solved by [4], in which a local sequence alignment was used that allows for a certain ambiguity of order if the alarms are close in time. It introduced the modified Smith-Waterman (MSW) algorithm, which is considered a prevailing benchmark in the AFSA literature [15]. The MSW algorithm generates a similarity matrix, which is used as the input for an agglomerative hierarchical clustering approach with a single-linkage (AHC-SL) to cluster similar ASs [4]. One limitation arises from the penalization of alarms in one AS that could not possibly be aligned with a matching counterpart in another AS. A disagreement on the number of ACTs in two ASs therefore negatively affects their similarity, thus making the MSW approach less robust to irrelevant alarms. Reference [18] proposed an improved version of the MSW algorithm by applying a filtering step based on the Jaccard-distance, as described in [1]. Henceforth, this method is referred to as *MSW-J*. Further alignment approaches were presented that aimed at reducing the computational effort required to carry out the MSW approach [10] and that applied alarm priority information as a primary similarity indicator [12].

A third category of AFSA methods is string metrics, which are based on distance or similarity measures [15]. For example, in addition to its utilization in the pre- or postprocessing of AS pairs, the Jaccard-distance was also used in [6] and [8] as a primary measure for the clustering of similar ASs. It considers only the binary activity of alarm variables, which are the unique identifiers of configured alarms, and not the number or order of ACTs and is therefore robust to any ambiguity in both. However, the Jaccard-distance overrates the similarity between two ASs that share common alarms but have considerable disagreement in their respective dynamics. Another string metric is the Levenshtein-distance, which uses the number of edits, i.e., insertion, deletion, and substitution of ACTs, that are needed for the transformation of one AS into another AS [8]. It shares some properties with the DTW in [1] and therefore has limitations if ACTs are annunciated in a swapped order. Another promising AFSA string metric, proposed in [8], uses the term frequency-inverse document frequency (TF-IDF) for the pairwise comparison of ASs. The TF-IDF is a frequently utilized measure in natural language processing that applies a

bag-of-words model, i.e., a simplified representation of the alarms in an AS that does not consider their order but rather their quantity. Moreover, a unique feature of the TF-IDF is its weighting of the relevance of alarm variables according to their probability of occurrence with regard to all ASs. Eventually, similar ASs are clustered using the "density-based spatial clustering of applications with noise" (DBSCAN) [19]. Reference [8] demonstrated that this method generates robust and meaningful results compared to other methods, especially when Jaccard-distance-based postprocessing is applied. Henceforth, this method is referred to as *T-A-J*. It was also used in [9] as a primary step for the causal analysis of ASs. However, it is less robust to irrelevant ACTs of alarm variables with a high weight.

In conclusion, the data-driven AFSA approaches described here show some deficits in fulfilling both requirements R1 and R2. Moreover, most of these approaches use fixed alarm rates and time windows to detect ASs in historical data, e.g., in [1], [4], [8], [9], and [18], which could result in important alarms or ASs being missed [16]. This deficiency justifies the proposal of a novel method that is robust against both order ambiguity and some irrelevant ACTs while still considering relevant aspects of the AS's dynamic structure.

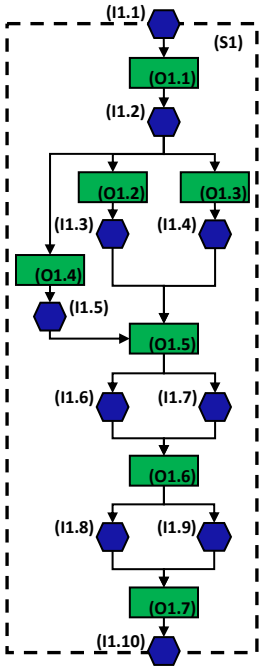
It was further shown in [15] that all of the existing AFSA methods share the common property of using an alarm sequence representation as input, i.e., a sequence of alarm instances ordered by their ACT times. However, [15] also examined two research areas that are similar to the idea of AFSA, namely, alarm similarity analysis and online alarm flood classification. The former examines the correlation between alarm variables, and the latter identifies known AS patterns in incoming alarm floods [15]. In both areas, several approaches have demonstrated that using alarm series, i.e., alarm data represented as time series, can be beneficial and produce more meaningful results, e.g., in [15] and [23], than when using only alarm activations. Moreover, [15] illustrated the advantages of using alarm coactivations for alarm analysis, i.e., two or more alarm variables that are active at the same time.

3 Proposed Approach

3.1 Overview of the Proposed Approach

Based on the findings in [15] and the promising *T-A-J* approach in [8], this paper proposes an improvement to the *T-A-J* approach that aims at meeting the requirements R1 and R2. The improvement is achieved by using two novel TF-IDF-based AS clustering methods that utilize alarm series data for the analysis of individual alarm variables (*T-S-J*) and their coactivations (*T-C-J*). Here, each configured alarm, e.g., a high- or low-alarm, is denoted by an individual alarm variable. Eventually, the postprocessed clustering results from *T-S-J* and *T-C-J* are merged by a novel validation step that focuses on the detected AS outliers.

Figure 1 shows the general structure of the proposed "alarm series similarity analysis method" (ASSAM) using the "formalized process description" given in [21]. The process operators (green rectangles) and generated and processed information (blue hexagons) are described in detail below. *T-S-J* is specified by process operators O1.1, O1.2, O1.4, O1.5, and O1.6 and results in I1.8, whereas *T-C-J* is defined by O1.1, O1.3, O1.4, O1.5, and O1.6 and generates the output I1.9.



ID	System Limit
S1	Alarm Series Similarity Analysis Method (ASSAM)

ID	Process Operator
O1.1	Preprocessing of historical alarm subsequences
O1.2	Computation of the alarm subsequence distance matrix using alarm variables
O1.3	Computation of the alarm subsequence distance matrix using alarm coactivations
O1.4	Computation of the Jaccard-distance matrix
O1.5	Postprocessing of the alarm subsequence distance matrices
O1.6	Clustering of similar alarm subsequences
O1.7	Validation of the alarm subsequence clusters

ID	Information
I1.1	Detected historical alarm subsequences
I1.2	Preprocessed alarm subsequences
I1.3	T-S-J: Alarm subsequence distance matrix
I1.4	T-C-J: Alarm subsequence distance matrix
I1.5	Jaccard-distance matrix
I1.6	T-S-J: Postprocessed distance matrix
I1.7	T-C-J: Postprocessed distance matrix
I1.8	T-S-J: Clusters of similar alarm subsequences
I1.9	T-C-J: Clusters of similar alarm subsequences
I1.10	Validated alarm subsequence clusters

Figure 1: *"Formalized process description" of the proposed "alarm series similarity analysis method" (ASSAM).*

3.2 Details of the Proposed Approach

The ASSAM starts with O1.1. The input of this first step is a set of detected historical ASs (I1.1), which were obtained using the "alarm coactivations and events detection method" (ACEDM) proposed in [16]. The ACEDM uses a "median absolute deviation"-based outlier detection in time distances between ACTs and alarm deactivations to find potential ASs. Subsequently, an alarm coactivation constraint is used to validate the detected historical ASs. It was shown that the ACEDM is more precise and robust in detecting coherent abnormal situations than are methods that use arbitrary alarm rate-thresholds in fixed or sliding time windows [16]. The ASSAM uses a time series representation of alarm data, i.e., a binary alarm series. Here, each alarm variable α_i is represented by a time series [15]:

$$S_i(t) = \begin{cases} 1, \text{if } t \in \mathcal{T}_i \\ 0, \text{else} \end{cases} \quad (1)$$

where $\mathcal{T}_i$ is the set of times t in which α_i is active. Trivial ASs with only one active alarm variable are eliminated. Moreover, to reduce the computational effort in the following steps, only those alarm variables that are active at least once in any of the subsequences are selected (I1.2).

Similar to *T-A-J*, the TF-IDF is used to weight alarm variables and their pairwise coactivations in O1.2 and O1.3, respectively. The time series for the coactivation of two alarm variables α_i and α_j can be represented as follows (following [15]):

$$S_{ij}(t) = \begin{cases} 1, \text{if } S_i(t) = S_j(t) = 1 \\ 0, \text{else.} \end{cases} \quad (2)$$

To calculate S_{ij} for all possible α_i and α_j in an ASs, alarm variables must have an identical sampling rate and an identical number of samples. Here, only those alarm variable pairs that are coactive at least once in any of the analyzed ASs are selected. The TF-IDF is then computed for each alarm variable (*T-S-J* in O1.2) or pair of alarm variables (*T-*

C-J in O1.3) and each alarm subsequence AS (following [8]):

$$\text{tf-idf}(a, AS) = \text{tf}(a, AS) * \text{idf}(a) \quad (3)$$

with the "term frequency":

$$\text{tf}(a, AS) = |S_a|_{AS} \,/\, \max(\{|S_a|_{AS} \mid a \in AS\}) \quad (4)$$

and the "inverse document frequency":

$$\text{idf}(a) = \log_e(|\boldsymbol{AS}| \,/\, |\{AS \in \boldsymbol{AS} | a \in AS\}|) \quad (5)$$

where a can be either an alarm variable or a pair of alarm variables, $|S_a|_{AS}$ is the number of samples in which a is active in AS, and $\boldsymbol{AS}$ is the set of all ASs. Subsequently, the Euclidean distance is used to calculate the pairwise distances between any two alarm subsequences AS_i and AS_j [8]:

$$d_{ij} = \sqrt{\sum_{k=1}^{m} (\text{tf-idf}(a_k, AS_i) - \text{tf-idf}(a_k, AS_j))^2} \quad (6)$$

where m is either the total number of alarm variables (O1.2) or the total number of alarm variable pairs (O1.3). Eventually, both resulting distance matrices in O1.2 (I1.3) and O1.3 (I1.4) are normalized to the range 0 to 1.

Identical to [8], the AS distance matrices are postprocessed here. This step aims to reduce spurious low distances between ASs that share only a small number of active alarm variables [1]. In O1.4, the Jaccard distances for all AS pairs are calculated using the following formula (following [8]):

$$d_{ij}^{Jac} = n_{ij}^{\text{xor}} \,/\, n_{ij}^{\text{or}} \quad (7)$$

where n_{ij}^{xor} is the number of distinct alarm variables that are exclusively active in either AS_i or AS_j and n_{ij}^{or} is the number of distinct alarm variables that are active in any of the two ASs. The resulting Jaccard distance matrix (I1.5) is then used in O1.5 for the postprocessing of I1.3 and I1.4. Each distance value in the postprocessed distance matrices I1.6 and I1.7 can be calculated as follows [8]:

$$\hat{d}_{ij} = \begin{cases} d_{ij}, \text{if } d_{ij}^{Jac} < \tau^{Jac} \\ 1 \quad , \text{else} \end{cases} \quad (8)$$

where τ^{Jac} is the Jaccard distance threshold that determines whether an AS pair is considered potentially similar.

In O1.6, both I1.6 and I1.7 are used to generate two partitions of $\boldsymbol{AS}$ using DBSCAN [19]. Reference [8] demonstrated the feasibility of utilizing DBSCAN when used for the clustering of ASs with different distance measures. DBSCAN identifies regions of high density, i.e., ASs that are close to each other in terms of the distance. Clusters are identified by core points, where an AS is considered as such if at least $(minPts - 1)$ other ASs are within a distance less than or equal to a threshold ε. ASs with no neighboring subsequences in proximity are considered outliers. Two advantages of DBSCAN are its distinct outlier label and the absence of a manual specification of the number of clusters [19]. The resulting clustering solution can be represented as $C = \{c_{-1}, c_0, c_1, \ldots, c_n\}$, where c_i depicts the ith cluster and c_{-1} groups all detected outliers. Here, *T-S-J* and *T-C-J* generate C^S (I1.8) and C^C (I1.9), respectively.

It is reasonable to assume that C^S and C^C possibly differ to some extent. In fact, preliminary tests have suggested that for some abnormal situations, one of the two chosen criteria, i.e., alarm variables and their coactivations, can have advantages over the other and result in more meaningful clusters. To benefit from both criteria, we propose a novel step (O1.7) that aims at validating the outliers in *T-S-J* (I1.8) by using *T-C-J* (I1.9). The former is used as the basis here since preliminary performance results have indicated that it is more robust to different settings of ε. The concept of the proposed approach is the following: for each outlier in c_{-1}^S, the corresponding label in C^C is analyzed. If *T-C-J* considers this subsequence as an outlier as well, it is labeled as such in the validated clustering solution $\hat{C}^{SC}$ (I1.10). If, however, the AS is part of c_i^C with $i \geq 0$, the outlier label in *T-S-J* is considered potentially erroneous. Next, we try to find the best match for c_i^C in C^S. One way to achieve this is to compare c_i^C to each regular cluster in C^S using a suitable similarity measure. Here, we propose using the Braun-Blanquet formula for the calculation of the similarity s_{ij}^{BB} between two clusters c_i and c_j. It can be calculated as follows [17]:

$$s_{ij}^{BB} = n_{ij} \,/\, \max(|c_i|, |c_j|) \quad (9)$$

where n_{ij} denotes the number of shared ASs in both clusters and $|c_i|$ and $|c_j|$ represent the number of ASs in c_i and c_j, respectively. Of all clusters in C^S with a similarity greater than or equal to a validation threshold τ^{Val}, the one with the highest similarity to c_i^C is considered the best match, i.e., c_j^S. Eventually, the former outlier is clustered in $\hat{c}_j^{SC}$. Otherwise, it remains an outlier and is grouped in $\hat{c}_{-1}^{SC}$. Moreover, all nonoutlier cluster labels in $\hat{C}^{SC}$ are assigned according to the cluster labels in C^S.

3.3 Discussion of the Limitations and Advantages of the Proposed Approach

One limitation of the ASSAM arises from the computational effort necessary for the calculation of *T-C-J*. In fact, one characteristic of *T-C-J* is that its TF-IDF vectors can yield a length of $\binom{m}{2}$ at most, i.e., the maximum number of unordered alarm variable pairs, where m is the total number of alarm variables. Subsequently, the coactivation of each alarm variable pair needs to be determined for each sample. Furthermore, as *T-C-J* considers only alarm variable pairs, a single alarm variable can have an excessive impact on the TF-IDF representation of an AS; i.e., it is considered in numerous elements of the TF-IDF vector. Future research could therefore apply suitable feature selection to determine the most relevant alarm variable combinations for the analysis of alarm coactivations, e.g., using an IDF threshold to select the most characteristic alarm variable pairs.

Nevertheless, the ASSAM shows relevant advantages compared to state-of-the-art methods. Swapped alarm orders and a varying number of ACTs in similar abnormal situations can be characteristic of real-world industrial processes [4][15]. The proposed utilization of time series data in AFSA expands the view to the dynamic properties of activated alarm variables and the dynamic structure of the underlying ASs instead of focusing on a point-to-point examination of sequenced ACTs. In fact, the calculation of the TF in (4) is not affected by the order or number of ACTs.

Moreover, randomly activated short alarms that are irrelevant for the situation have only a small impact due to the consideration of the number of active samples in (4). Hence, the proposed ASSAM and its components *T-S-J* and *T-C-J* fully satisfy the requirements R1 and R2 given in [4].

4 Evaluation

This section evaluates and compares the performances and characteristics of three relevant AFSA methods described in Section 2 and the method proposed in Section 3. Subsection 4.1 gives a brief overview of the evaluation dataset used. Subsection 4.2 deals with choosing a suitable evaluation measure. The obtained evaluation results are presented in Subsection 4.3.

4.1 Evaluation Dataset

The examined clustering methods are applied to the openly accessible simulation dataset[1] introduced in [16]. It is based on a simulation model of the TEP, a frequently used benchmark in process automation [3][5]. It can be separated into five modules: a two-phase chemical reactor, a condenser, a vapor-liquid separator, a stripper, and a reboiler. Furthermore, the TEP includes 11 automatic pneumatic control valves, two pumps, and one compressor [5]. The alarm system of the TEP defines 81 low-alarm and 81 high-alarm thresholds as well as five high-high-alarm and three low-low-alarm thresholds [16].

The dataset includes 100 simulation runs with 300 specified abnormal situations. These situations were designed using eight different root-cause disturbances with variations in their respective durations, disturbance scaling, and combinations. These variations as well as random influences affect the number of activated alarm variables, the order of alarm instances, and their dynamic behavior. The alarm system generates a total of 7343 alarm instances over all 300 situations [16].

The application of the ACEDM on the TEP dataset results in 358 detected ASs (I1.1 in Figure 1), of which 310 subsequences show more than one alarm instance (I1.2 in Figure 1). The latter are used as the preprocessed input for all clustering methods examined here, thus being able to specifically compare the performances of the selected AFSA methods. One advantage of the TEP simulation dataset is that all induced abnormal situations are explicitly known [16], thus making it possible to use an external validity index, which compares the computed clusters to a given ground-truth partition [17]. The 310 preprocessed ASs are therefore manually assigned to 22 ground-truth clusters according to the details described in [16]. Each cluster includes 4 to 30 similar ASs. Furthermore, 14 of the 310 subsequences are labeled outliers, as they contain only random parts of the respective underlying abnormal situation and show no similarities to any other ASs.

4.2 External Validity Index

For evaluation, a suitable external validity index needs to be chosen, which facilitates an appropriate performance comparison of different clustering methods in terms of which clusters best fit with the given ground-truth [17]. A frequently used pair-counting index, which evaluates the agreement of a ground-truth partition C_0 and a computed trial partition C_1 on the pairs of objects in the dataset [17], is the adjusted Rand-index (ARI) [13]. Reference [20] suggested using the ARI as a benchmark in cluster evaluations. The ARI can be calculated using the following formula [13][20]:

$$ARI = \frac{2(ad-bc)}{(a+b)(b+d)+(a+c)(c+d)} \quad (10)$$

where a is the number of AS pairs that are in the same cluster in both partitions, b is the number of AS pairs that are in the same cluster in C_0 but in different clusters in C_1, c is the number of AS pairs that are in the same cluster in C_1 but in different clusters in C_0, and d is the number of AS pairs that are in different clusters in both partitions. If C_0 and C_1 are identical, the ARI yields a value of 1. A value of 0 arises in the case where C_0 and C_1 are statistically independent [13][20]. A detailed analysis of the properties and characteristics of the ARI can be found in [20].

4.3 Evaluation Results

An overview of the methods examined here is given in Table 1. Two methods, *J* and *MSW-J*, are used as benchmarks for the evaluation of the TF-IDF-based methods, namely, *T-A-J*, the proposed ASSAM, and its components *T-S-J* and *T-C-J*. In addition, some of these methods are compared to versions of them that do not use the postprocessing step in process operator O1.7 (s. Figure 1), namely, *T-A, T-S,* and *T-C*. This evaluation approach allows for a systematic and in-depth examination of the effectiveness of the ASSAM and its components, i.e., the proposed alarm variable and coactivation input, as well as the postprocessing of the obtained distance matrices and the validation of potential outliers. For *MSW-J,* the algorithm parameters were set according to [4], i.e., $\delta = -0.4$, $\mu = -0.6$, and $\sigma^2 = 4$. The Jaccard-distance threshold for *MSW-J, T-A-J, T-S-J,* and *T-C-J* was set to 0.4. The validation threshold τ^{Val} of the ASSAM was set to 0.5. Based on preliminary tests, the *minPts* parameter of DBSCAN was set to 3 for all methods. The distance threshold of the AHC-SL and ε of DBSCAN values between 0.1 and 1.0 with a step size of 0.001 were examined. For the evaluation of the ASSAM, both components *T-S-J* and *T-C-J* used the same ε due to the assumption that the individual tuning of two parameter settings would be cumbersome in an industrial application of the ASSAM.

For each method, the highest ARI value, which was obtained by applying all considered parameter settings, is shown in Figure 2. J, T-A, and T-A-J, which do not consider alarm activation durations or their order, show the lowest ARI values of all examined methods. Indeed, in some cases, these three methods detected similarities between ASs that are in different ground-truth clusters and arose from different root- cause disturbances, thus resulting in fewer, though larger, computed clusters, i.e., 15 clusters for J. By using an optimal ε of 0.081, both T-A and T-A-J labeled 32 outliers, which represents the highest number for all examined methods. The corresponding ASs were characterized by random variations in the number of ACTs of those alarm variables with a high value for the IDF-vector.

[1] https://dx.doi.org/10.21227/326k-qr90

Abbreviation	Method	Post-proc.	Clustering Method
J	Jaccard-distance [6][8]	No	DBSCAN
MSW-J	Modified Smith-Waterman [4][18]	Yes	AHC-SL
T-A	TF-IDF using alarm sequences [8]	No	DBSCAN
T-A-J	TF-IDF using alarm sequences [8]	Yes	DBSCAN
T-S	TF-IDF using alarm series: alarm variables	No	DBSCAN
T-S-J	TF-IDF using alarm series: alarm variables	Yes	DBSCAN
T-C	TF-IDF using alarm series: alarm coactivations	No	DBSCAN
T-C-J	TF-IDF using alarm series: alarm coactivations	Yes	DBSCAN
ASSAM	TF-IDF using alarm series: alarm variables and coactivations	Yes	DBSCAN

Table 1: *Overview of the examined and compared methods.*

In contrast, the consideration of the order of ACTs with ambiguity to short-term variations in *MSW-J* resulted in a higher ARI value. *MSW-J* detected 20 clusters and 24 outliers using an optimal distance threshold of 0.276. An in-depth inspection of the obtained results revealed that *MSW-J* was not always able to distinguish between significant variations for the same root-cause disturbance. Moreover, the detected outliers differed considerably from those in the given ground-truth; i.e., *MSW-J* was not always able to find similarities between two ASs with identical ground-truth cluster labels in cases where both disagreed on the number of ACTs.

The proposed methods *T-S-J* and *T-C-J*, as well as the alternative versions *T-S* and *T-C,* showed an improved performance compared to that of the existing AFSA methods. Of the 170 alarm variables of the TEP, only 76 were active at least once in the dataset, with 1851 alarm variable pairs showing coactivation. As a result, the TF-IDF vectors of *T-C-J* contain more than 24 times as many elements as the TF-IDF vectors of *T-S-J*. Both *T-S-J* and *T-C-J* were able to detect 23 clusters and 12 outliers with as few as 12 mislabeled ASs. The optimal ε values for *T-S-J* and *T-C-J* were 0.095 and 0.080, respectively. An in-depth inspection of the cluster labels resulting from *T-S-J* and *T-C-J* revealed that they are essentially identical except for four ASs, which stem from two different abnormal situations. Interestingly, in both cases, one of the methods classified two of the subsequences as outliers, whereas the other method classified them correctly according to the ground-truth cluster labels. The application of both *T-S-J* and *T-C-J* and the subsequent validation of outliers in the ASSAM were shown to result in more meaningful clusters; i.e., only 10 ASs were mislabeled, which resembles the ground-truth best. This finding was also supported by the ASSAM yielding an ARI value superior to that of all other examined methods. Another significant phenomenon revealed in Figure 2 is that the postprocessing of the TF-IDF-based methods was beneficial regarding the optimal ARI value. This phenomenon is further analyzed in Figures 3 and 4.

Figure 3 illustrates the heatmaps of the distance matrices for the TF-IDF-based methods. The ASs in the columns and rows are ordered by the ground-truth cluster labels. This allows for the visual evaluation of the distance measures used. A trial partition identical to the ground-truth is characterized

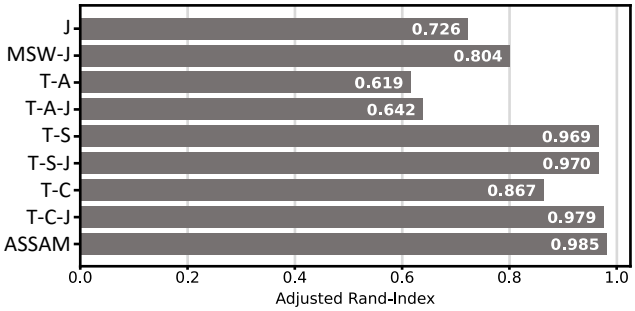


Figure 2: *Performance of the examined AS clustering methods under the optimal parameter settings.*

by the dark colored blocks along the diagonal of the distance matrix. In contrast, undesired similarities between different ground-truth clusters appear as dark colored off-diagonal blocks. Figures 3 (a), (b), and (c) show the computed distances without the application of the postprocessing step. The distance matrix of *T-A* in Figure 3 (a) contains some erroneously high distances between ASs that have the same ground-truth cluster label and numerous spuriously high similarities in terms of the off-diagonal blocks. Only one cluster presents a desirable high visual contrast. The corresponding simple ASs are characterized by only two alarm variables that are both active throughout the respective abnormal situations. The distance matrices of *T-S* and *T-C* in Figures 3 (b) and (c) show a substantially higher visual contrast between blocks along the diagonal and in the off-diagonal areas than shown in Figure 3 (a). In fact, the highest contrast can be found in Figure 3 (b), which is reflected by *T-S* having the highest ARI value of all TF-IDF-based approaches without postprocessing. The lower performance and lower visual contrast of *T-C* in Figure 3 (c) can possibly be explained by its relatively high-dimensional TF-IDF vectors; i.e., ASs with only a few coactive alarm variables tend to show a shorter distance than that of subsequences with a high number of nonzero elements in the TF-IDF vectors. Figures 3 (d), (e), and (f) show the computed distance matrices after the application of the postprocessing step. By assigning the highest distance value to most of the erroneous AS pairs, the resulting visual contrast shows high agreement with the cluster structure of the ground-truth. However, Figure 3 (d) demonstrates that *T-A-J* yields low distance values for most of the remaining subsequence pairs, thus impeding the detection of the correct ground-truth clusters. In contrast, Figures 3 (e) and (f) depict overall higher distances in the remaining off-diagonal pairs for *T-S-J* and *T-C-J*. This advantageous characteristic resulted in higher ARI values for both proposed components of the ASSAM.

The performance and the number of resulting clusters for the TF-IDF-based methods over all considered settings of the DBSCAN parameter ε are illustrated in Figure 4. The corresponding diagram for the ASSAM is similar to that of *T-S-J* and is therefore not depicted here. The comparison of Figures 4 (a), (b), and (c) and Figures 4 (d), (e), and (f) reveals the benefits of the postprocessing step: *T-A, T-S,* and *T-C* show a steep and sudden performance decline for ε greater than 0.07 (for *T-C*) and 0.15 (for *T-A* and *T-S*). On the other hand, *T-A-J, T-S-J,* and *T-C-J* present an improved performance baseline for ε greater than 0.25 with an ARI value of approximately 0.47 and 11 to 14 detected clusters. In that case, the clustering results are mainly determined by

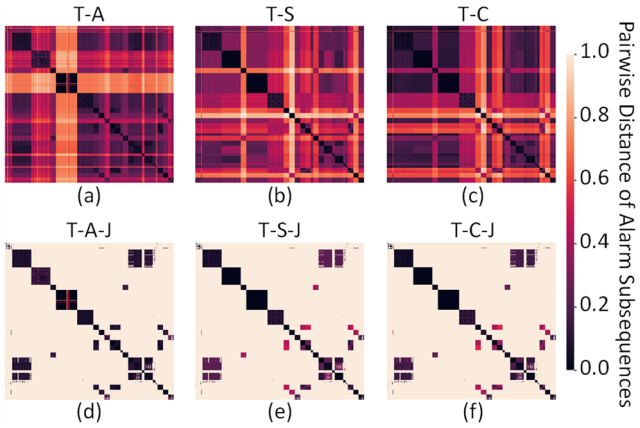


Figure 3: *Matrices of the pairwise distances of alarm subsequences for the TF-IDF-based methods. Each pixel represents the distance between two alarm subsequences. The alarm subsequences in the columns and rows are ordered by the ground-truth cluster labels. (a) T-A. (b) T-S. (c) T-C. (d) T-A-J. (e) T-S-J. (f) T-C-J.*

the postprocessing step. Moreover, the close inspection of Figure 4 indicates that the range of suitable values for ε, which results in ARI values close to the maximum, is approximately twice as long for *T-S-J* and *T-C-J* compared to *T-S* and *T-C*. In conclusion, the postprocessing step makes the proposed methods more robust to changes in ε and the clustering results more reliable in cases where an optimal ε cannot be determined using a ground-truth partition.

5 Discussion and Conclusions

The evaluation in Subsection 4.3 showed that the existing AFSA methods are not able to meet the requirements defined in [4] (s. Section 2) to the fullest extent. In fact, the in-depth examination revealed that the methods *J, T-A, T-A-J*, and *MSW-J* can handle a certain ambiguity of the order of alarms in two compared ASs (*R1*), whereas none of them could suitably tolerate irrelevant alarms occurring in one or both ASs (*R2*). These methods are therefore not able to correctly detect all underlying AS similarities. Despite this distinct limitation, the clustering results obtained by *MSW-J* showed a relatively high agreement with the given ground-truth of the TEP dataset used here. However, the *MSW* necessitates the cumbersome tuning of four interrelated parameters, i.e., δ, μ, σ^2, and the distance threshold of the AHC-SL. It was further demonstrated that the proposed TF-IDF-based method ASSAM as well as its components *T-S-J* and *T-C-J* are able to fulfill all given requirements. Moreover, the ASSAM achieves the best performance among all considered AS clustering methods. This result confirms the assumption that the clustering results can be improved when using alarm series data and alarm coactivations as input. Overall, the evaluation showed that clustering methods that consider the dynamic properties of activated alarm variables and the dynamic structure of the ASs consistently demonstrate a higher performance than that of methods that utilize a less extensive data input.

One limitation of the ASSAM results from its need for a relatively high computational effort using *T-C-J*; i.e., each sample in a subsequence needs to be analyzed on occurring pairwise alarm coactivations. In contrast, *T-S-J* maintains a

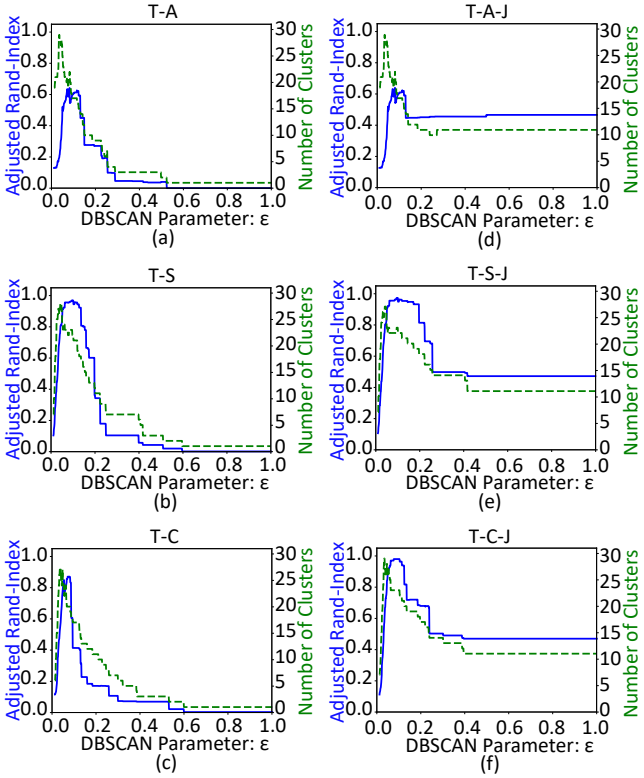


Figure 4: *Performance (blue solid lines) and number of clusters (green dashed lines) for the TF-IDF-based methods over all considered settings of the DBSCAN parameter ε. (a) T-A. (b) T-S. (c) T-C. (d) T-A-J. (e) T-S-J. (f) T-C-J.*

relatively low computational burden. Another limitation results from the necessity of tuning the DBSCAN parameter ε. In this context, it was proven that the postprocessing step of *T-S-J* and *T-C-J* makes them and the ASSAM more robust to changes in the parameter settings than without postprocessing and compared to *T-A-J*. It is noteworthy that this beneficial characteristic of the ASSAM makes it more suitable for an industrial application where a priori knowledge for parameter tuning can be limited. Moreover, this finding substantiates the viability of the postprocessing step, as hypothesized in [8].

Furthermore, the evaluation indicated a high agreement between the clustering results of *T-S-J* and *T-C-J*. However, the data also showed that the proposed combined approach ASSAM has advantages over the individual methods. For industrial practitioners, we recommend using *T-S-J* in cases where a low computational burden is of relevance. In other cases, we propose using the ASSAM as intended. It is reasonable to assume that in processes similar to the TEP used here, this approach can produce more meaningful clustering results. Future studies should apply the proposed ASSAM and its components *T-S-J* and *T-C-J* to further industrial and experimental datasets. Furthermore, it should be investigated whether suitable feature selection for *T-C-J* can be found to reduce the relatively high dimensionality of its TF-IDF vectors. Moreover, further research should evaluate whether modern machine learning methods, e.g., representation learning, can improve the analysis of similar historical ASs.

Additional recommendations for future research opportunities include causal investigations of the detected AS

clusters and the mining of meaningful alarm patterns therein. Subsequently, different clusters that share common root-cause disturbances can be further examined regarding intermittent similarities and differences in their patterns. AI may help to capture the different versions of an abnormal situation, which can arise due to varying operator interventions. Such versions could be characterized by a normalization or an escalation, e.g., an emergency shutdown, of the process. This approach could help to explore suitable operator interventions and characteristic indicators that allow for a timely identification of the specific version of an abnormal situation that is most likely to occur.

References

[1] K. Ahmed, I. Izadi, T. Chen, D. Joe and T. Burton, "Similarity analysis of industrial alarm flood data," *IEEE Trans. Autom. Sci. Eng.*, 10(2):452–457, Apr. 2013, doi: 10.1109/TASE.2012.2230627.

[2] *Management of Alarm Systems for the Process Industries*, ANSI/ISA Standard 18.2-2016, 2016.

[3] A. Bathelt, N. Lawrence Ricker and M. Jelali, "Revision of the Tennessee Eastman process model,". *IFAC-PapersOnLine*, 48(8):309–314, Jun. 2015, doi: 10.1016/j.ifacol.2015.08.199.

[4] Y. Cheng, I. Izadi and T. Chen, "Pattern matching of alarm flood sequences by a modified Smith-Waterman algorithm", *Chem. Eng. Res. Des.,* 91, 1085–1094, 2013, http://dx.doi.org/10.1016/ j.cherd.2012.11.001.

[5] J.J. Downs and E.F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, 17(3):245–255, Mar. 1993, doi: 10.1016/0098-1354(93)80018-I.

[6] M. Fahimipirehgalin, I. Weiss and B. Vogel-Heuser, "Causal inference in industrial alarm data by timely clustered alarms and transfer entropy," *2020 European Control Conf.*, 2056–2061, May 2020, doi: 10.23919/ECC51009.2020.9143823.

[7] J. Folmer and B. Vogel-Heuser, "Computing dependent industrial alarms for alarm flood reduction," *IEEE 9th Int. Multi-Conf. Systems, Signals and Devices*, 1-6, Mar. 2012, doi: 10.1109/SSD.2012.6198008.

[8] M. Fullen, P. Schüller and O. Niggemann, "Validation of similarity measures for industrial alarm flood analysis", in O. Niggemann and P. Schüller (Eds.), *"IMPROVE - Innovative Modelling Approaches for Production Systems to Raise Validatable Efficiency*," 8:93-110, Berlin, Heidelberg, Germany: Springer Viehweg, Aug. 2018, doi: 10.1007/978-3-662-57805-6_6.

[9] M. Fullen, P. Schüller, and O. Niggemann, "Semi-supervised case-based reasoning approach to alarm flood analysis," in J. Beyerer, A. Maier, and O. Niggemann (Eds.), *"Machine Learning for Cyber Physical Systems,"* 11:53-61, Berlin, Heidelberg, Germany: Springer Viehweg, Apr. 2019, doi: 10.1007/978-3-662-59084-3_7.

[10] C. Guo, W. Hu, S. Lai, F. Yang and T. Chen, "An accelerated alignment method for analyzing time sequences of industrial alarm floods", *J. Process Contr.,* 57:102–115, 2017, http://dx.doi.org/10.1016/j.jprocont.2017.06.019.

[11] M. Hollender and C. Beuthel, "Intelligent alarming: effective alarm management improves safety, fault diagnosis and quality control," *ABB Review Magazine* 1/2007:20–23.

[12] W. Hu, J. Wang and T. Chen, "A local alignment approach to similarity analysis of industrial alarm flood sequences", *Control Eng. Pract.,* 55:13–25, 2016, http://dx.doi.org/10.1016/j.conengprac.2016.05.021.

[13] L. Hubert and P. Arabie, "Comparing partitions," *J. Classif.*, 2(1):193–218, Dec. 1985, doi: 10.1007/BF01908075.

[14] B. Klöpper, M. Dix, D. Siddapura, and L.T. Taverne, "Integrated search for heterogeneous data in process industry applications—A proof of concept," *IEEE 14th Int. Conf. Ind. Inf.,* 1306-1311, Jul. 2016, doi: 10.1109/INDIN.2016.7819369.

[15] M. Lucke, M. Chioua, C. Grimholt, M. Hollender and N.F. Thornhill, "Advances in alarm data analysis with a practical application to online alarm flood classification," *J. Process Contr.*, 79:56–71, Jul. 2019, doi: 10.1016/j.jprocont.2019.04.010.

[16] G. Manca and A. Fay, "Detection of historical alarm subsequences using alarm events and a coactivation constraint," *IEEE Access*, 9:46851-46873, Mar. 2021, doi: 10.1109/ACCESS.2021.3067837.

[17] M. Rezaei and P. Franti, "Set matching measures for external cluster validity," *IEEE Trans. Knowl. Data Eng*, 28(8):2173–2186, Aug. 2016, doi: 10.1109/TKDE.2016.2551240.

[18] V. Rodrigo, M. Chioua, T. Hagglund and M. Hollender, "Causal analysis for alarm flood reduction," *IFAC-PapersOnLine*, 49(7):723–728, Jun. 2016, doi: 10.1016/j.ifacol.2016.07.269.

[19] E. Schubert, J. Sander, M. Ester, H.P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.,* 42(3), Article 19, doi: 10.1145/3068335.

[20] D. Steinley, "Properties of the Hubert-Arabie adjusted Rand index," *Psychological Methods,* 9(3):386-396, Sep. 2004, doi: 10.1037/1082-989X.9.3.386.

[21] *Formalised process descriptions - Concept and graphic representation*, VDI/VDE Publication 3682 Blatt 1:2015-05, 2015.

[22] B. Vogel-Heuser, D. Schuetz and J. Folmer, "Criteria-based alarm flood pattern recognition using historical data from automated production systems (aPS)", *Mechatronics,* 31:89–100, 2015, http://dx.doi.org/10.1016/j.mechatronics.2015.02.004.

[23] Z. Yang, J. Wang, and T. Chen, "Detection of correlated alarms based on similarity coefficients of binary data," *IEEE Trans. Autom. Sci. Eng.*, 10(4):1014-1025, Oct. 2013, doi: 10.1109/TASE.2013.2248000.