Validation of similarity measures for industrial alarm flood analysis*

Marta Fullen[1] and Peter Schüller[2] and Oliver Niggemann[1,3]

[1] Fraunhofer IOSB-INA Institutsteil für industrielle Automation, Lemgo, Germany
[2] Technische Universität Wien, Institut für Logic and Computation, Vienna, Austria
[3] Institute Industrial IT, Lemgo, Germany

Abstract. The aim of industrial alarm flood analysis is to assist plant operators who face large amounts of alarms, referred to as alarm floods, in their daily work. Many methods used to this end involve some sort of a similarity measure to detect similar alarm sequences. However, multiple similarity measures exist and it is not clear which one is best suited for alarm analysis. In this paper, we perform an analysis of the behaviour of the similarity measures and attempt to validate the results in a semi-formalised way. To do that, we employ synthetically generated floods, based on assumption that synthetic floods that are generated as 'similar' to the original floods should receive similarity scores close to the original floods. Consequently, synthetic floods generated as 'not-similar' to the original floods are expected to receive different similarity scores. Validation of similarity measures is performed by comparing the result of clustering the original and synthetic alarm floods. This comparison is performed with standard clustering validation measures and application-specific measures.

1 Introduction

The phenomenon of alarm flooding is a recurring problem in industrial plant operation [19]. It occurs when the frequency of alarm annunciations is so high that it exceeds the operators capability of understanding the situation. This creates a dangerous situation where the operator might overlook critical alarms that could lead to significant downtime, irreversible damage or even loss of life [10].

The main reason for alarm flooding is imperfect alarm system design. Figure 1 presents a typical alarm generation system consisting of several modules. The alarms are triggered based on sensor values, thresholds or more complex rules. Basic signal and alarm filtering can be used to remove alarms that are known to be noise before they are displayed to the operator. Operators themselves have the opportunity to shelve alarms that they consider irrelevant or redundant. However, the real potential lies in the "contextual preclassification" block, where expert knowledge can be combined with intelligent computational methods to assist the operator.

* This paper is an extended version of [7] and involves a more detailed analysis of the behaviour of similarity measures.

O. Niggemann and P. Schüller (Eds.), *IMPROVE – Innovative Modelling Approaches for Production Systems to Raise Validatable Efficiency*, Technologien für die intelligente Automation 8, https://doi.org/10.1007/978-3-662-57805-6_6

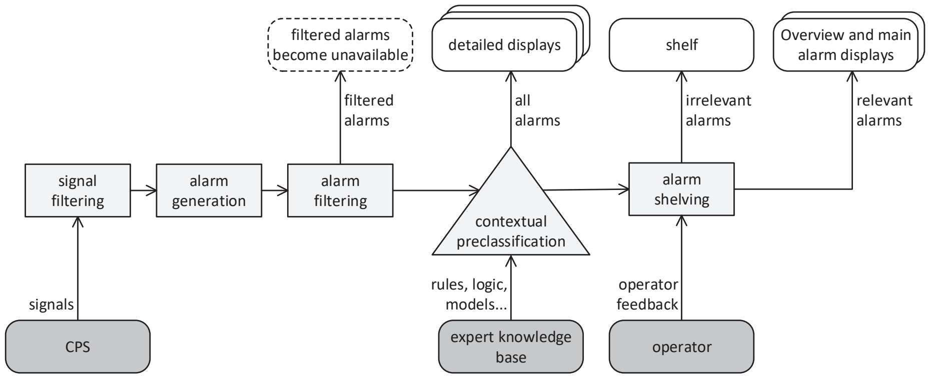


Fig. 1: Typical alarm filtering and shelving system extended by contextual preclassification step (from: [16]).

Diagnosis of a failure of an industrial plant is a non-trivial task that requires extensive knowledge of causalities between symptoms produced by the system [15]. Model-based methods often require explicit representation of expert knowledge about the system and about possible failures, which is rarely available to the extent that is necessary for the method to provide reliable results. In these cases shallow data-driven approaches are more applicable. Data-driven approaches are based purely on the data obtained from the system, possibly with rudimentary expert knowledge inserted when available to improve the results. Data-driven methods directly analyse, manage and reduce the alarm annunciation and therefore flooding [2], without a semantic representation of the system. Multiple approaches exist to this end, drawing from the data mining fields such as sequence identification and pattern recognition [18, 4], correlation analysis [21] or visualisation [13]. Many of these approaches utilise flood similarity measure of some kind, e.g., [20, 3].

Alarm flood detection and clustering is a data-driven approach to handle alarm flooding. An operator assistance system can detect a newly annunciated alarm flood, compare it to the previously seen floods and identify the most similar cluster. If the history of flood of the plant has annotations, such as a log of repairs done to remedy the reason of an original flood, the system can make a suggestion to the operator regarding the fault diagnosis and repair procedure.

A major challenge in creating such a system is determining how similarity between alarm floods should be defined. A multitude of similarity measures (and, analogously, distance measures) exists in the field of data mining and clustering [5]. Another challenge is, that real industrial alarm data is difficult to work with, e.g. because of a high volume of alarms, or because of poor alarm system design. While a certain similarity measure might work for a certain application, there is no guarantee it will work in other application scenarios, as there is not systematic method for quantifying the usefulness of a given similarity measure in the industrial setting.

We here propose a semi-formal approach to answering that issue by defining an experimental design to validate the behaviour of a distance measure in regard to alarm flood clustering. Analysis of the behaviour of the distance measure can then help choose the most suitable distance measure. We also reproduce and extend the alarm flood detection and clustering approach by Ahmed et al. [1] with additional similarity measures based on Term frequency-inverse document frequency (TF-IDF) representation [12] and Levenshtein distance [14], apply these measures to a large real industrial alarm log and evaluate them using our validation method.

Our study shows that the flood distance measure in [1] behaves significantly different from the other analysed distance measures; in particular, the other measures produce a more stable clustering in the presence of noise in the data.

Methodology for detecting floods and computing distance measures is given in Section 2, as well as our proposed approach for validating the behaviour of clustering results using alarm floods synthetically generated from real alarm flood data. Section 4 presents and discusses results of empirical validation on a real industrial dataset. We summarize the results and conclude in Section 5.

2 Clustering methodology

The approach for alarm flood similarity measure analysis is illustrated in Figure 2. Alarm flood similarity measure analysis consists of seven steps divided into two stages: (1) clustering performed once on alarm floods detected in the alarm log and (2) repeatable generation of synthetic alarm floods and clustering. Analysis is preceded by the acquisition of alarm log A from the Cyber—Physical Production System (CPPS) performed by component C1, as described in Section 2.1. Component C2 is responsible for detecting the alarm floods which as described in Section 2.2. Stage 1 clustering (component C4) is performed only once on the set of alarm floods detected in the alarm log F^O and yields a clustering solution S^O. Clustering methodology is described in Section 2.3. Stage 2 is performed multiple times and begins with synthetic flood generation (component C3) which yields a set of synthetic floods F^S. Synthetic floods are clustered alone by component C6 which yields solution S^S, as well as merged with the original floods forming F^{OS} and clustered together by component C5, yielding solution S^{OS}. The procedure for generating synthetic floods is described in Section 3.1. The process is repeated an arbitrary number of times and each round is evaluated according to the metrics described in Section 3.

2.1 Alarm log acquisition

Alarms triggered within a CPPS are processed by an alarm logging system (component C1) within its data acquisition and management system. Alarms are displayed to the operator and saved into a historical database referred to as a historical alarm log A. Recorded alarms are characterised by, at the very least, alarm ID, alarm trigger time and alarm acknowledgement time. Alarm log can also contain additional information such as a description, location and other details of the alarm.

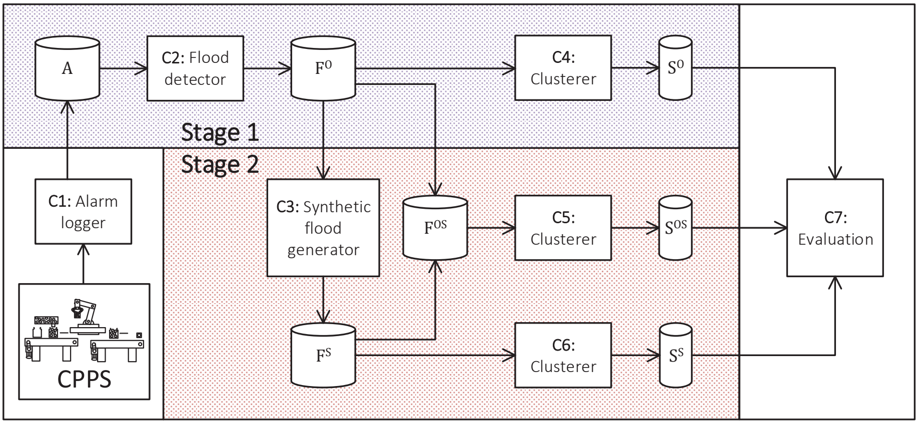


Fig. 2: Approach for validating alarm flood similarity measures.

2.2 Flood detection and preprocessing

First step of the analysis is performed once for a historical alarm log.

Flood detector (component C2) detects the floods based on the alarm flood definition. According to the industry standard for Management of Alarm Systems for Process Industries, an alarm flood begins when the alarm annunciation rate exceeds 10 alarms per 10 minutes and ends when the alarm annunciation rate drops under 5 alarms per 10 minutes [9], see Figure 3c.

However, the specific methodology for flood detection is up to interpretation.

Firstly, the alarm logs should be preprocessed to account for the lingering and chattering alarms (see Figure 3a). Ambiguity arises in the case of lingering alarms that are active for a long time before the flood is considered to begin. Since they might be relevant to the root cause of the flood, we include such alarms in the flood if they were triggered no longer than a lingering threshold t_l before the flood start (see Figure 3b). Moreover, chattering alarms falsely increase the number of alarms within a time period so they are merged before flood detection.

Redundant alarms convey the same information, see Figure 3a, while unnecessarily cluttering the operator display. As a most simple example, two alarms triggered based on the same condition are redundant. In more complex situations, a redundant alarm is triggered based on a condition which is directly caused by an event that triggers another alarm, and is not influenced by any other factors. Redundant alarms are more difficult to deal with, requiring an analysis and refinement of the alarm system or using an advanced reasoning to detect causalities between alarms.

Floods are detected using a sliding window and the outcome is a set of original floods F^O.

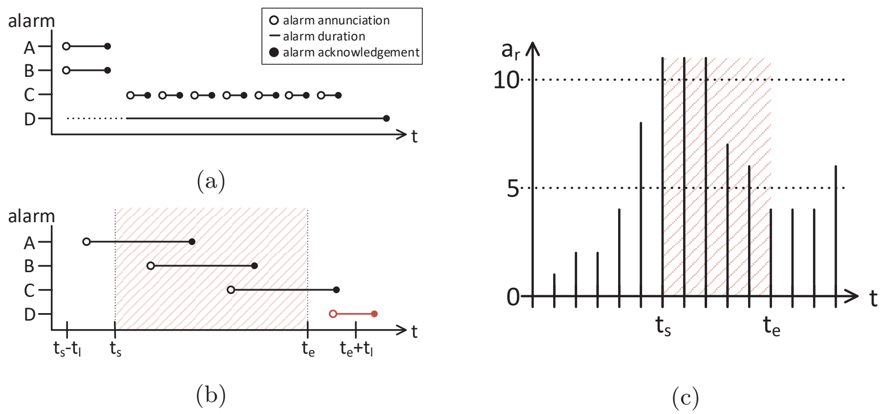


Fig. 3: (a) Undesirable alarms caused by imperfections in alarm system design: A, B - redundant alarms, C - chattering alarm, D - lingering alarm. (b) Flood detection from a more detailed point of view, where a dilemma arises which alarms exactly should be included. Alarms A, B, C are included in the flood, while alarm D is not. t_s - alarm flood start, t_e - alarm flood end, t_l - lingering alarm inclusion margin. (c) Flood detection: a_r - alarm rate (alarms active in 10 minute period), t - 10-minute time periods, t_s - alarm flood start, t_e - alarm flood end.

2.3 Alarm flood clustering

Original flood set F^O is clustered to obtain the original clustering solution S^O (see Figure 2 component C4). Density-based spatial clustering of applications with noise (DBSCAN) [6] is a clustering algorithm that intrinsically deduces the most-fitting number of clusters. It is based on the concept of density, where points within a specified distance threshold ϵ to each other are considered to belong to a dense area—a cluster. Points that are distant from dense areas are considered outliers and are gathered in a separate group. Preliminary experiments showed, that it is a disadvantage to predefine the number of clusters, because the number of natural clusters in the data is expected to change with different distance measures. Therefore, to perform an unbiased comparison of distance measures, we use the DBSCAN algorithm which is not biased to a fixed number of clusters and exclude the outliers in our analysis.

Choice of a distance measure is the second, after the clustering algorithm itself, most critical aspect when performing clustering, and the focus of this paper. Each distance measure requires a specific data representation and four distance measures are analysed: Jaccard distance [11] on a bag of words representation, distance based on the frequency of consecutive alarms [1], Euclidean distance on TF-IDF representation [12] and Levenshtein distance [14]. Chosen measures are focused either on the appearance of alarms (Jaccard distance and TF-IDF representation) or on the order of alarms (frequency of consecutive alarms and Levenshtein distance); in the latter, the absolute time distance is not considered.

Jaccard distance (J) Jaccard distance is the ratio between alarm types occurring only in one of the two floods, and the number of alarm types in both floods. Each flood f_i is represented as a binary vector $f_i = (a_1, a_2, \ldots, a_m)$, where m is the number of unique alarm identifiers in the complete alarm log and a_j is a binary value representing whether an alarm appeared in the flood, regardless of its count. The Jaccard distance between floods f_i and f_j is given by

$$J_{ij} = \frac{|f_i \text{ xor } f_j|}{|f_i \text{ or } f_j|}, \tag{1}$$

where $|x|$ returns the number of true values in vector x. Alarm types that are absent from both floods are irrelevant for Jaccard distance. This measure was used as preprocessing in [1].

Frequency of consecutive alarms (F) This measure was proposed in [1] based on a simplification of first-order Markov chains. Each flood is represented as a matrix of counts of each pair of alarms appearing consecutively,

$$P = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ f_{m1} & f_{m2} & \cdots & f_{mm} \end{bmatrix}, \tag{2}$$

where f_{ij} is the frequency of alarm a_j being annunciated directly after alarm a_i in a given alarm flood. Then, the distance between two floods can be calculated as a distance between their P matrices, e.g. using Frobenius distance.

Term frequency-inverse document frequency (T) TF-IDF is a measure often used in natural language processing to weight terms in a document according to how frequent and discriminative they are with respect to a document collection. We apply TF-IDF to weight alarms in alarm floods with respect to the collection of all floods. TF-IDF is calculated for each alarm a and flood f as

$$\text{tf-idf}(a, f) = \text{tf}(a, f) * \text{idf}(a). \tag{3}$$

Term frequency is calculated as

$$\text{tf}(a, f) = \frac{f_{a,f}}{|f|}, \tag{4}$$

where $f_{a,f}$ is the number of annunciations of alarm a in flood f and $|f|$ is the total number of alarm annunciations in f.

Inverse document frequency is calculated as

$$\text{idf}(a) = \log_e \frac{|F|}{|\{f \in F \mid a \in f\}|}, \tag{5}$$

where $|F|$ is the total number of floods. TF-IDF score is calculated for every alarm type and every flood in the log and yields a flood representation in the form of a vector of length m, the total number of unique alarm signatures.

Two floods f_i and f_j can then be compared using a distance measure such as Euclidean distance between two vectors:

$$d(f_i, f_j) = \sqrt{\sum_{k=1}^{m} (\text{tf-idf}(a_k, f_i) - \text{tf-idf}(a_k, f_j))^2}. \quad (6)$$

Levenshtein distance (L) This metric counts the amount of "edits" that are needed to transform one sequence into another one, where an edit is a symbol insertion, symbol deletion, or a symbol substitution. To apply this metric, alarm floods are represented as sequences of symbols, which in turn represent unique alarm types. The Levenshtein distance $d(|f_i|, |f_j|)$ between floods f_i and f_j is calculated recursively, where the distance for the first x and y symbols of f_i and f_j, respectively, is calculated as follows:

$$d(x,y) = \begin{cases} \max(x,y) & \text{if } \min(x,y) = 0 \\ \min \begin{cases} d(x-1,y)+1 \\ d(x,y-1)+1 \\ d(x-1,y-1)+\mathbf{1}_{f_i(x) \neq f_j(y)} \end{cases} & \text{otherwise,} \end{cases} \quad (7)$$

where $\mathbf{1}_{condition}$ is the indicator function. The distance score is normalised over the length of floods.

2.4 Distance matrix postprocessing

Distance matrix is optionally postprocessed using a Jaccard similarity measure threshold t. The rationale for postprocessing is that only floods that have a significant number of alarms in common can be assigned a low distance value [1]. To ensure that, distance values for each pair of floods are filtered based on the value of the Jaccard distance for this pair. For any distance measure d, distance value d_{ij} remains unchanged if the Jaccard distance between the corresponding floods d^J_{ij} is lower than the threshold t; otherwise, the distance is replaced with the maximum value $d_{ij} = 1$:

$$\hat{d}_{ij} = \begin{cases} d_{ij} & \text{if } d^J_{ij} < t, \\ 1 & \text{otherwise.} \end{cases} \quad (8)$$

Postprocessed distance matrix $\hat{d}$ yields clustering solution $\hat{S}$.

3 Evaluation methodology

At the core of the proposed approach, the original flood set F^O is concatenated with the synthetic flood set F^S to create a joint set F^{OS}. Procedure to generate synthetic floods is described in section 3.1. The clustering result of the join set, S^{OS} is used to evaluate the behaviour of the distance measures. Clustering (see Figure 2 components 5 and 6) is performed analogously to the original floods, as described in 2.3. We propose to evaluate the clustering solutions in regard to the distance measure behaviour using two measures: cluster membership of synthetic floods m_1 (see section 3.2) and cluster stability, based on adjusted Rand index [17], in five variants: R_1, R_2, R_3, R_4 and R_5 (se section 3.3).

3.1 Synthetic flood generation

Second step of the analysis is focused on generating synthetic alarm floods that can be included in clustering to evaluate the distance measures. Synthetic floods are generated based on the existing set of original floods F^O. Each original flood f^O is used to create one synthetic flood f^S. The original alarm flood is modified in three ways to create the synthetic flood, as illustrated in Figure 4: (i) by addition of randomly chosen alarms, (ii) by removal of randomly chosen alarms and (iii) by transposing randomly chosen pairs of alarms.

Those three modifications correspond to the expected possible variations in an industrial alarm log, which stem e.g. from delays on the bus or data acquisition sampling rate. For simplicity, we always apply an equal amount, at least one, of all three modifications to each flood. The number of modifications is varied throughout the experiments to obtain different synthetic flood sets, ranging from very similar to dissimilar to the original floods. We represent the degree of modification as a percentage of the number of alarms in a flood that has been modified.

This way of modification of floods from real datasets is chosen according to our domain knowledge and experience with floods and their variation in the industrial setting.

3.2 Cluster Membership of Synthetic Floods

The first validation approach is the fraction of synthetic floods that is assigned to the same cluster as their original flood. It is calculated as

$$m_1 = \frac{|f_i^S : c(f_i^S) = c(f_i^O)|}{|F^S|}, \tag{9}$$

where f_i^S is a synthetic flood generated from original flood f_i^O, $c(f)$ is the cluster flood f was assigned to and F^S is the set of all synthetic floods. This measure is calculated for each synthetic flood with respect to its mother flood, without considering (potentially random) similarities to other original floods. Therefore, if a synthetic flood by chance becomes more similar to a different original flood than its mother flood, it will not affect the results.

3.3 Cluster Stability

We can consider the original flood clustering results to be the ground truth, or the "target", as in the supervised machine learning validation methods. This assumption is made only for the purpose of validation of the similarity measure behaviour.

Each synthetic flood has a known mother flood, and is expected to be treated similarly by the clustering algorithm if it was generated with a low degree of modification; i.e., the synthetic flood is expected to have similar distance scores to other floods as its mother flood, and therefore to be clustered alike. Hence, the synthetic floods are given the same target as their mother floods. On the other hand, if the synthetic flood was generated using a high degree of modification it is expected to be treated differently by the clustering algorithm than its original flood.

Results of clustering original and synthetic floods together can be compared to that target solution. Adjusted RAND index is a well-known measure for quantifying partition agreement between clustering solutions, while disregarding the actual cluster number. Adjusted RAND index for two partitions $C_1 = \{c_0, c_1, ..., c_n\}$ and $C_2 = \{c_0, c_1, ..., c_n\}$ of items is calculated as

$$R = \frac{a+b}{a+b+c+d}, \tag{10}$$

where a is the number of pairs of items that are in the same cluster in C_1 and in C_2, b is the number of pairs of items that are in different clusters in C_1 and in C_2, c is the number of pairs of items that are in the same cluster in C_1 but in different clusters in C_2, and d is the number of pairs of items that are in different clusters in C_1 but in the same cluster in C_2.

We specify the following five different variants of cluster stability to perform the analysis. Cluster stability R_1 compares the cluster membership of original floods and synthetic floods when clustered together. Cluster stability R_2 is used to quantify the change in original flood partitioning when clustered with and without the synthetic floods. Cluster stability R_3 quantifies the difference between the expected outcome, which is the result of the original flood clustering, and the obtained solution. Cluster stability R_4 compares the cluster memberships of the original and the synthetic floods when clustered separately. Finally, cluster stability R_5 compares the solutions for the original floods obtained with and without the postprocessing step described in section 2.4.

$$R_1 = R(S^{OS}(0,n), S^{OS}(n+1, n+m)), \tag{11}$$

$$R_2 = R(S^O, S^{OS}(0,n)), \tag{12}$$

$$R_3 = R(S^{OS}, S^O \cdot S^O), \tag{13}$$

$$R_4 = R(S^O, S^S), \tag{14}$$

$$R_5 = R(S, \hat{S}), \tag{15}$$

where $S^O = s_0^O, ..., s_n^O$ is the clustering solution for the set of original floods $F^O = f_0^O, ..., f_n^O$, $S^S = s_0^S, ..., s_m^S$ is the clustering solution for the set of synthetic floods $F^S = f_0^S, ..., f_m^S$, $S^{OS} = s_0^{OS}, ..., s_n^{OS}, s_{n+1}^{OS}, ..., s_{n+m}^{OS}$ is the clustering solution for the joint set of original and synthetic floods $F^{OS} = f_0^{OS}, ..., f_n^{OS}, f_{n+1}^{OS}, ..., f_{n+m}^{OS}$, $S^{OS}(0,n)$ and $S^{OS}(n+1, n+m)$ denote the joint solution subsets corresponding to the original and synthetic floods respectively and $\hat{S}$ denotes a solution obtained from a postprocessed distance matrix.

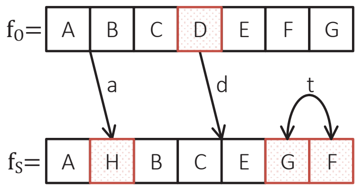


Fig. 4: Creation of a synthetic flood f_s from an original flood f_o composed of alarms A, B, C, D, E, F, G through three edits: a - addition, d - deletion, t - transposition.

Table 1: Specification of performed experiments.

Exp.	Dist. measure	Postproc.
1	Jaccard	No
2	Jaccard	Yes
3	Frequency	No
4	Frequency	Yes
5	TF-IDF	No
6	TF-IDF	Yes
7	Levenshtein	No
8	Levenshtein	Yes

4 Empirical evaluation results

The following results extend those already published in [7]. The experimental data set is a 25-day alarm log from a production plant from the manufacturing industry, consisting of 15 k annunciations of 96 alarm types. The flood detection algorithm has been modified to account for the lingering alarm problem (cf. Section 2.2) and yielded 166 alarm floods with an average length of 61 alarms. We perform similarity measure analysis as described in the methodology section and analyse the behaviour of the distance measure as it changes with adding synthetic floods to the dataset and influences the structure of the clusters. DBSCAN clustering is used in the experiments and since it does not require specification of the number of target clusters, it is possible to observe how do the synthetic floods change the structure of the data, for example by creating new inherent clusters.

4.1 Visualization on a demonstrative set of 25 floods

Fig. 5 visually demonstrates clustering and validation methodology on a reduced dataset of 25 floods. Floods are arranged on X- and Y-axis in the same order, and pixels indicate degree of similarity, where white means highest distance, and strong colour means equality (which occurs mostly on the diagonal).

Row (a) presents original distance matrices of distance measures J, F, T, L. Jaccard distance identified four floods as exactly the same (solid square in the image), while all other distance metrics show that they are in fact not identical: albeit they are composed of the same alarms, they differ in the number and order of their annunciations.

Row (b) shows distance matrices postprocessed according to 2.4. Clearly this filters out many values in the matrices, in particular in the case of TF-IDF many low distance values are reset to highest distance. This property of TF-IDF can be explained, because the principle of TF-IDF is to put more weight on terms that occur more often in one flood and that occur less often in other floods. Therefore, to

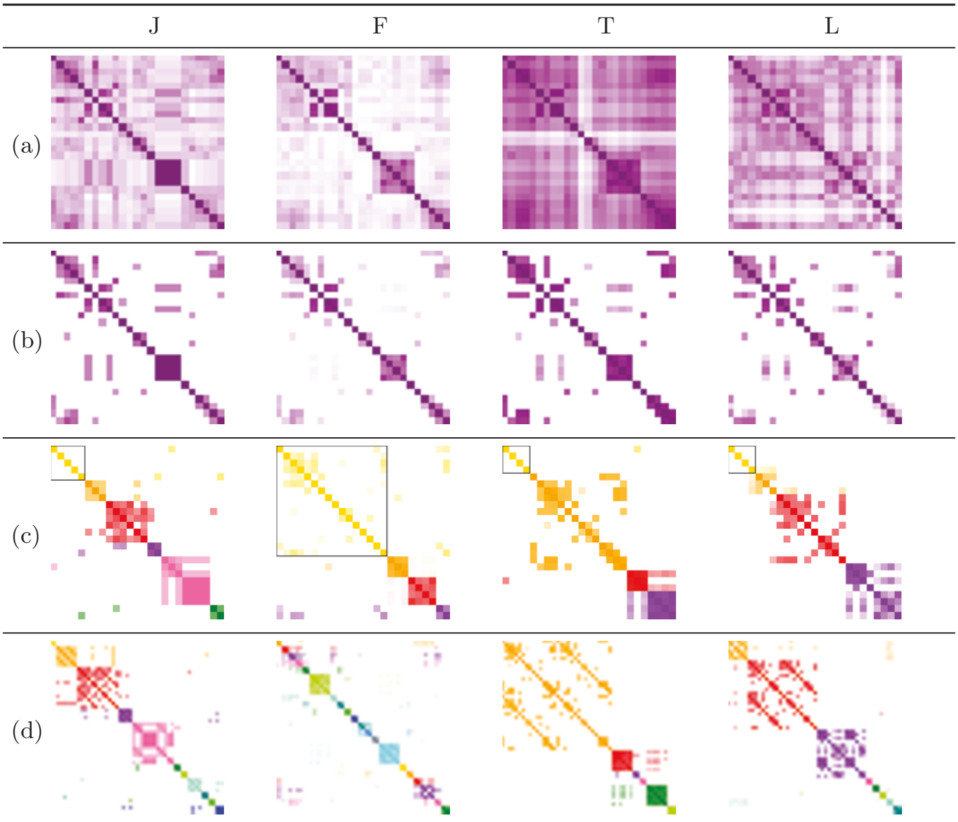


Fig. 5: Distance matrices obtained at every stage of the experiments for measures J, F, T, and L for a reduced dataset of 25 floods. Pixels indicate pairwise similarity between floods which are arranged on X- and Y-axis in the same order: stronger colours show lower distance. Rows show (a) original flood distance matrices; (b) distance matrices postprocessed using Jaccard distance threshold; (c) clustering solutions using distances from (b), coloured according to the cluster number, with outliers in the top left cluster highlighted by a black border; and (d) clustering solutions after adding a 10% modified synthetic flood for each flood.

obtain high distance, a pair of floods needs to contain a distinct set of alarms that has low frequency in other floods. In our dataset this rarely happens, concretely it mainly happens in short floods, where the term frequency of rare alarms contributes a large value to the distance measure.

Row (c) presents the DBSCAN clustering results on distance matrices of row (b), where floods have been rearranged and coloured according to the cluster number. The top left cluster represents outliers: their distance to other floods was under the ϵ threshold and therefore they were not assigned to any cluster.

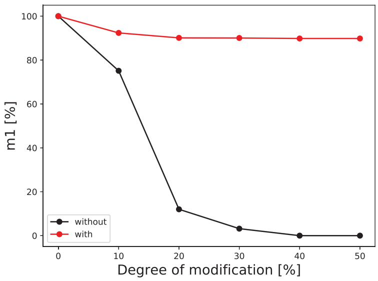


Fig. 6: Measure m_1 - cluster membership of synthetic floods calculated with and without outliers for frequency of consecutive alarms distance measure.

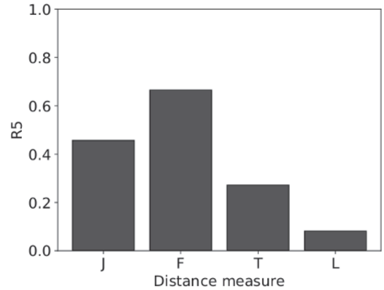


Fig. 7: Measure R_5 - cluster stability between clustering solutions with and without postprocessing calculated for the original flood set.

Row (d) presents clustering results after introducing synthetic floods with 10% modification. Synthetic floods cause changes in the cluster structure, although the structure of original clusters is mostly retained. As in this case, synthetic floods are quite similar to their respective mother floods, many of the outlier floods are clustered together with their synthetic counterpart and form two-flood clusters.

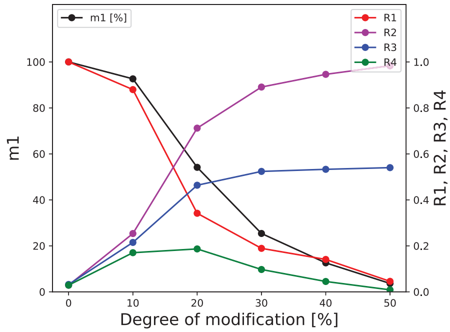
4.2 Clustering with synthetic floods on the full dataset

Further experiments were performed on the whole alarm flood dataset. Each distance measure is evaluated with and without the postprocessing step, as listed in table 1. For validation experiments we generate experimental sets of synthetic floods with a degree of modification (the amount of edits) ranging between none and 50%. Each experiment is repeated 10 times and the measure values are averaged.

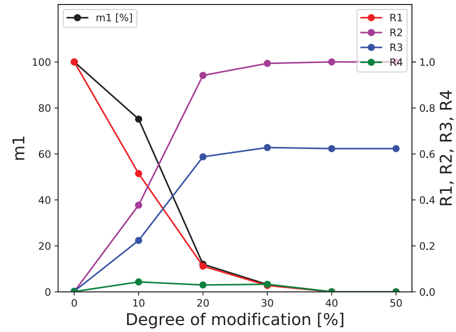
Figure 6 shows cluster membership of synthetic floods measure m_1 calculated for the frequency of consecutive alarms distance measure under two conditions: including and excluding the outliers. Since the amount of outliers is high in a real industrial dataset, including them in the analysis distorts the results. The number of classifications that are correct from the perspective of m_1 measure (so number of synthetic floods assigned to the same cluster as the corresponding original flood) is very high because the outliers constitute such a large cluster and the synthetic floods based on the outliers also are assigned to the outlier cluster. Therefore, in further analysis we focus on the floods that were not outliers.

Figure 7 presents the cluster stability between solutions obtained from unchanged distance matrices and postprocessed distance matrices for the original flood set. Low values of R_5 indicate that the solutions obtained with and without postprocessing are not consistent. Results imply that postprocessing heavily influences the results.

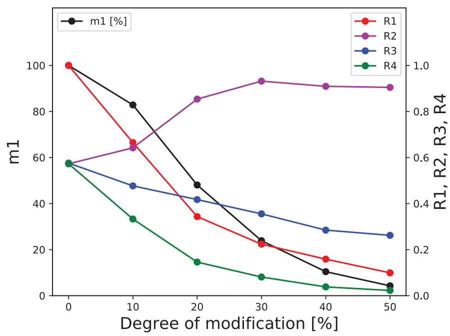
Figures 8a to 8d present the evaluation measures m_1, R_1, R_2, R_3 and R_4 for experiments with varying degrees of modification used to generate the synthetic



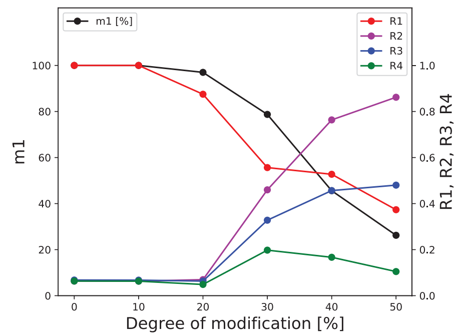
(a) Experiment 1 - Jaccard distance measure without postprocessing.



(b) Experiment 3 - Frequency of consecutive alarms distance measure without postprocessing.



(c) Experiment 5 - TF-IDF distance measure without postprocessing.

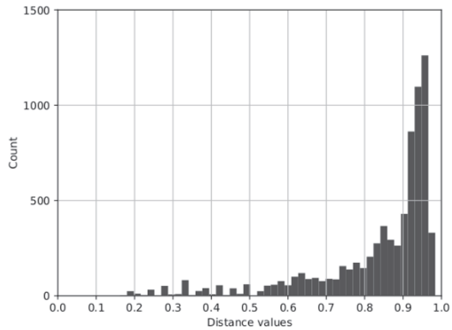


(d) Experiment 7- Levenshtein distance measure without postprocessing.

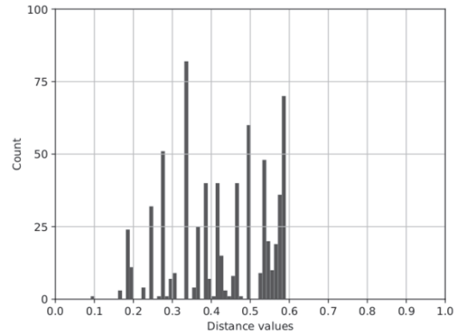
Fig. 8: Experimental results.

flood sets. Intuitively, identical floods will be clustered together, and indeed for every distance measure the m_1 and R_1 measures for synthetic floods with no modifications are maximal. As the degree of modification is raised, m_1 and R_1 both decrease. The decrease is most abrupt for frequency of consecutive alarms distance measure, which implies that this measure is most sensitive to even small variation in the data.

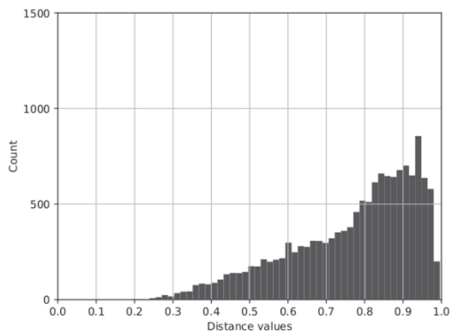
On the other hand, measures R_2 and R_3 increase with the degree of modification. As the synthetic floods become more and more different from the original floods, the clustering solution of the original floods resembles more the solution obtained in step 1 in the process. That means the synthetic floods no longer sufficiently resemble the original floods. Measure R_3 does not reach the upper limit of 1.0 like measure R_2 does. That is due to the parameter of the clustering algorithm which specifies the minimum number of samples to form a valid cluster. The set of original floods is smaller and therefore the minimum number of samples to form



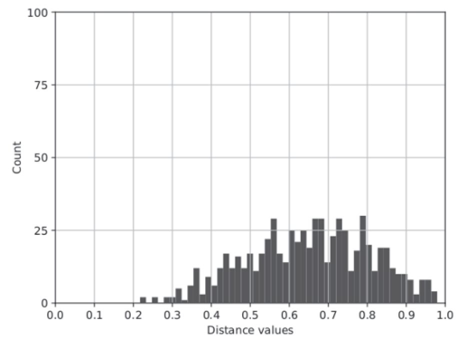
(a) Experiment 1



(b) Experiment 2



(c) Experiment 3.



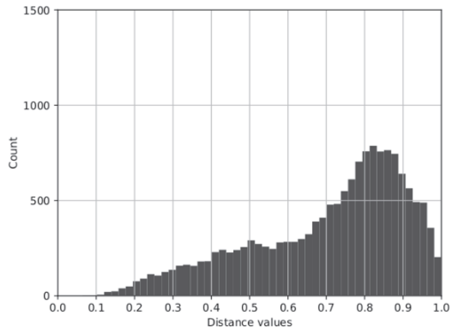
(d) Experiment 4.

Fig. 9: Comparison of distance value histograms before (left) and after (right) postprocessing for distance measure: a,b - Jaccard, c,d - Frequency of consecutive alarms. Postprocessed histograms (right) ignore the distance values of 1.

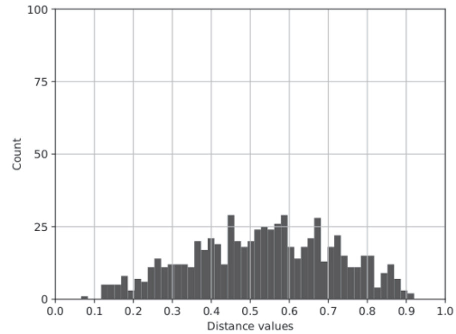
a cluster might not be reached for some subsets of floods, while after adding the synthetic floods, the minimum is reached and clusters are formed.

The effect of postprocessing is analysed further by comparison of histograms of distance matrices before and after postprocessing (Figures 9 and 10). During postprocessing we replace with 1 all the values that have the Jaccard distance value higher than the threshold $t = 0.6$. This means above a certain distance we assume it is maximum distance. The resulting histograms are dominated by these "1" values so we do not show them. We apply this postprocessing to the other distance matrices as well. We always use Jaccard distance as the criterion whether to reset a distance value to 1 (this is the methodology described in [1]) and therefore histograms for the other distance measures show values above 0.6.

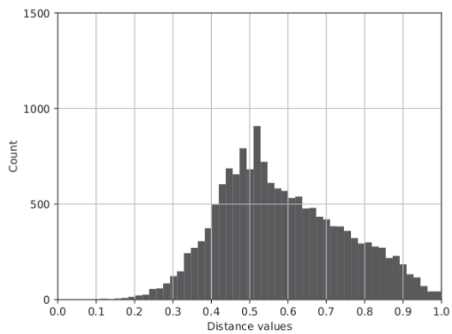
In the case of Jaccard distance measure (Figure 9b), postprocessing simply removes all the values in range of $(0.6, 1.0)$. While the postprocessed Levenshtein distance matrix (Figure 10d) is consistent with the Jaccard measure (in the sense that very few distance values remain in the range of $(0.6, 1.0)$), the two other



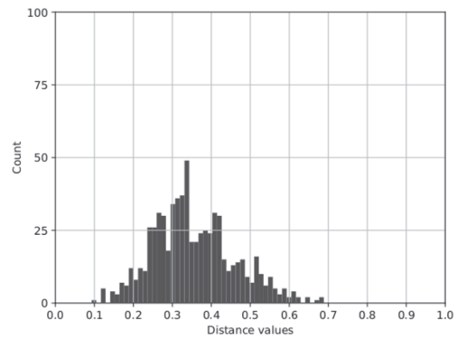
(a) Experiment 5.



(b) Experiment 6.



(c) Experiment 7.



(d) Experiment 8.

Fig. 10: Comparison of distance value histograms before (left) and after (right) postprocessing for each distance measure: a,b - TF—IDF, c,d - Levenshtein. Post-processed histograms (right) ignore the distance values of 1.

measures show that alarm floods that have many IDs in common may in fact be quite distant in the terms of (i) how frequently alarms appear in order (Figure 9d) and (ii) the most discriminative alarms (Figure 10b).

5 Conclusion

In this paper, we continue the analysis of similarity measures that can be used in alarm flood clustering [7].

The paper presents a methodology for validating similarity measures to help choose a measure which is best suited for similarity-based approaches used in alarm flood analysis. An example of a similarity-based approach is case-based reasoning, where a new alarm flood is compared to a database of known cases to suggest a course of action to the operator [8]. Choice of a similarity measure is a difficult problem, which is normally solved arbitrarily by an expert. Synthetically generated alarm floods create a controlled environment for observing the behaviour and sensitivity of the similarity measure to expected variations in the data, which is not possible using industrial data sets which are generated under unknown conditions and limited in size. The validation process has been shown using a real industrial dataset.

We have previously shown that the measure introduced in [1] produces very different results than our newly introduced measures and results suggest that the measure of [1] is less favourable. TF-IDF representation with Euclidean distance has shown the most balanced results.

Since the alarm logging systems are flawed we can assume the alarm log data to exhibit some degree of variability. For example, there may be delays on the bus or sampling may be too slow so that many alarms are logged with the same timestamp but not necessarily the correct order. Two of the analysed measures (frequency of consecutive alarms and Levenshtein distance) rely heavily on the order of alarms in the data. Results show that frequency of consecutive alarms distance measure is most sensitive to variations in data, while Levenshtein distance is not because it allows transpositions of adjacent alarms within a flood.

Results show that postprocessing heavily influences the results. It has been shown that floods containing many of the same alarm IDs may in fact be very distant when considering other characteristics, such as the order of the alarms or the discriminating value of alarm IDs.

Moreover, DBSCAN clustering appears to produce more meaningful results because of its adaptive choice of the number of clusters.

In the future work, using an annotated dataset (e.g. data generated in a controlled simulation environment) could help further analyse these effect and establish whether postprocessing is viable. Furthermore, additional similarity measures can be analysed, including measures that take absolute time distance into account.

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 678867.

References

1. Ahmed, K., Izadi, I., Chen, T., Joe, D., Burton, T.: Similarity analysis of industrial alarm flood data. In: IEEE Transactions on Automation Science and Engineering (Apr 2013)
2. Bergquist, T., Ahnlund, J., Larsson, J.E.: Alarm reduction in industrial process control. In: Proc. IEEE Conference on Emerging Technologies and Factory Automation. vol. 2, pp. 58–65 (2003)
3. Charbonnier, S., Bouchair, N., Gayet, P.: A weighted dissimilarity index to isolate faults during alarm floods. Control Engineering Practice 45, 110–122 (2015)
4. Charbonnier, S., Bouchair, N., Gayet, P.: Fault template extraction to assist operators during industrial alarm floods. Engineering Applications of Artificial Intelligence 50, 32–44 (2016)

5. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer (2009)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD. pp. 226–231. AAAI Press (1996)
7. Fullen, M., Schüller, P., Niggemann, O.: Defining and validating similarity measures for industrial alarm flood analysis. In: IEEE 15th International Conference on Industrial Informatics (INDIN) (2017)
8. Fullen, M., Schüller, P., Niggemann, O.: Semi-supervised case-based reasoning approach to alarm flood analysis. In: Machine Learning for Cyber Physical Systems (ML4CPS) (2017)
9. Instrumentation, Systems, and Automation Society: ANSI/ISA-18.2-2009: Management of Alarm Systems for the Process Industries (2009)
10. Izadi, I., Shah, S.L., Shook, D.S., Chen, T.: An introduction to alarm analysis and design. In: IFAC SAFEPROCESS. pp. 645–650 (2009)
11. Jaccard, P.: Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines. Bulletin de la Société Vaudoise des Sciences Naturelles 37, 241–272 (1901)
12. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28, 11–21 (1972)
13. Laberge, J.C., Bullemer, P., Tolsma, M., Reising, D.V.C.: Addressing alarm flood situations in the process industries through alarm summary display design and alarm response strategy. Intl. J. of Industrial Ergonomics 44(3), 395–406 (2014)
14. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
15. Niggemann, O., Lohweg, V.: On the diagnosis of cyber-physical production systems: State-of-the-art and research agenda. In: Proc. AAAI. pp. 4119–4126. AAAI Press (2015)
16. Norwegian Petroleum Directorate: YA-711 Principles for alarm system design (2001)
17. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical association 66(336), 846–850 (1971)
18. Vogel-Heuser, B., Schütz, D., Folmer, J.: Criteria-based alarm flood pattern recognition using historical data from automated production systems (aps). Mechatronics 31, 89–100 (2015)
19. Wang, J., Yang, F., Chen, T., Shah, S.L.: An overview of industrial alarm systems: Main causes for alarm overloading, research status, and open problems. IEEE Transactions on Automation Science and Engineering 13(2), 1045–1061 (2016)
20. Wang, J., Li, H., Huang, J., Su, C.: A data similarity based analysis to consequential alarms of industrial processes. Journal of Loss Prevention in the Process Industries 35, 29–34 (2015)
21. Yang, F., Shah, S., Xiao, D., Chen, T.: Improved correlation analysis and visualization of industrial alarm data. ISA Transactions 51(4), 499–506 (2012)