This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2797247, IEEE Access

Sequence Mining based Alarm Suppression

Gyula Dorgo, Janos Abonyi, *Member, IEEE*

***Abstract*—Despite the high-pace improvement of industrial process automation, the management of abnormal events still requires human actions. Alarm systems are becoming crucial in providing situation-specific information to the decreasing number of operators. The key role of an alarm management system is to ensure that only the currently significant alarms are annunciated. The design of alarm suppression rules requires the systematic analysis of the process and its control system. We give an overview of the recently developed data-driven techniques and show that the widely applied correlation based methods utilize a static view of the system. To provide more insight into the process dynamics and represent the temporal relationships among faults, control actions and process variables we propose of a multi-temporal sequence mining based algorithm. The methodology starts with the generation of frequent temporal patterns of the alarm signals. We transform the multi-temporal sequences into Bayes classifiers. The obtained association rules can be used to define alarm suppression rules. We analyze the dataset of a laboratory-scale water treatment testbed to illustrate that multi-temporal sequences are applicable for the description of operation patterns. We extended the benchmark simulator of a vinyl acetate production technology to generate easily reproducible results and stimulate the development of alarm management algorithms. The results of detailed sensitivity analyses confirm the benefits of the application of temporal alarm suppression rules which are reflecting the dynamical behaviour of the process.**

***Index Terms*—alarm management, process safety, frequent pattern mining, data mining**

I. Introduction

OPTIMIZING economic performance within environment and safety related constraints became the primary challenge of control systems [1]. Alarm management systems aim to minimize physical and economic loss through operator intervention in response to an abnormal situation [2]. According to the Engineering Equipment and Materials Users' Association (EEMUA) [3] the purpose of an alarm system is to redirect the operator's attention towards plant conditions requiring timely assessment or action. Therefore, properly designed and operated alarms help the operator to keep the processes in the normal operation range by indicating the presence of abnormal situations. In correspondence with this, alarm management means the efficient design, implementation, operation, and maintenance of industrial process alarms.

According to the guidelines of EEMUA, 300 alarms per day (one alarm in every five minutes) are manageable for an operator. In most of the European refineries, the number of alarms significantly exceeds this recommended level. For example, in a European refinery, the average number of alarms is 14250, with a peak of 26650 alarms, but numbers like these are not unusual [4].

MTA PE Lendulet Complex Systems Monitoring Research Group, University of Pannonia, Veszprem, H-8200, Hungary e-mail: janos@abonyilab.com

Typical alarm management performance metrics presented in Table I confirm, industrial metrics are far away from the specified goals [5].

TABLE I
TYPICAL ALARM PERFORMANCE METRICS AND TARGETS [5]

Metric	Target	Action limit
Average alarm rate per operator (alarms per day)	<288	>432
Average alarm rate per operator (alarms per hour)	<12	>18
Average alarm rate per operator (alarms per 10 minutes)	1-2	>3
Percent of 10-minute periods containing >10 alarms	<1%	>5%
Maximum number of alarms in a 10 minute period	≤10	>10
Percent contribution of top 10 most frequent alarms	<1% to ∼5%	>20%
Percent of time the system is in flood	<1%	>5%

There can be several reasons for bad alarm management performances. The number of alarms often increases due to the easily definable alarm trip points in distributed control systems (DCS). Today far more alarms are configured with often meaningless alarm levels for alarm variables that should not be alarmed at all, as defining alarms has no significant cost. Therefore a high number of nuisance alarms occur that do not require any corrective actions from the operators, as no real abnormalities appeared. This can cause the so-called *cry wolf effect* [6], the alarms lose their awareness-raising effect and operators are more likely to miss the relevant alarms that are buried by irrelevant ones. Bliss *et al.* substantiated the existence of the cry-wolf effect for alarm responses [7], which highlights the problem of informativeness of alarm systems. Sorkin and Woods showed that the system performance can be highly improved in light of the operators' workload characteristics [8]. Meyer *et al.* have performed an experimental study and suggested that the occurrence of hazard warnings depends on the operator's character and that the diagnostic value of a warning system decreases for better operators [9]. It is interesting to note that the issue of informativeness and alarm response has been also discussed in healthcare applications [10].

A. The definition and design of proper alarms

The design of alarm thresholds is the top priority to reduce the number of false or missed alarms (alarms without the presence of abnormal situation or alarms, which do not appear in case of abnormality). The proper definition of alarm thresholds does not guarantee a fully functional alarm management system, since there are a high number of alarms that cannot work properly with a static configuration of thresholds.

In some cases alarm limits should change according to process operating mode, therefore mode based alarm system [11] or dynamic alarm management [12] might be needed. Other main contributors of high alarm numbers are the *chattering alarms*. According to ANSI/ISA-18.2 (2009), any alarm occurring more than three times in a one-minute-period can be considered as a chattering alarm. In a more informal context, any alarm that appears with a disturbingly high frequency can be regarded as a chattering alarm [13]. To avoid chattering deadbands [14], [15], delay-timers [16], or filtering [17] can be used. A summary of methods for the improvement of alarm systems is given by Izadi *et al.* [18].

B. Advanced alarm reduction techniques

Although guidelines suggest having only one alarm for a single abnormal event, the number of interacting components make it almost infeasible to avoid redundant alarms. The main idea is that overlapping and redundant alarms should be grouped and these groups should be used for the detection of failures, prediction of future events and alarm suppression (see Figure 1) [2]. In the following we present a brief overview of methods developed to support this concept.

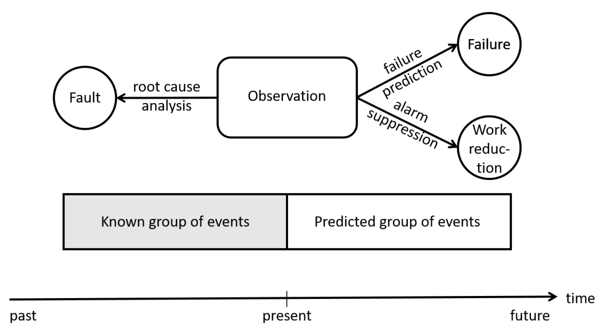

Fig. 1. Grouping of events can be useful in fault detection, root cause analysis and alarm suppression

1) Correlation analysis based techniques: The main idea of correlation analysis is to explore alarms always occurring together within a short period of time [19]. The results of this analysis can be used for the reduction of operator workload as has been reported in a Japanese ethylene plant [20]. Kondaveeti *et al.* used binary and Gaussian kernel based representations for the visualisation of clustered alarms [21], [22]. Yang *et al.* [23] also took into consideration the correlation delay and found that among the tested 22 similarity measures, the Sorgenfrei coefficient was the most effective for the comparison of binary represented alarm signals. As the estimated Sorgenfrei and Jaccard coefficients were found to be too small and the distribution of correlation delays require a large database this approach has been improved further in ref. [24]. Wang *et al.* [25] combined similarity analysis and causal relationships detection for the identification of consequential alarms and their evolution path. Grouping of massive alarm data in distributed parallel alarm management systems also utilizes alarm similarity measures [26]. More recently, Soares *et al.* investigated the alarm rate of a natural gas processing plant and applied alarm prioritization and clustering methods (correlation analysis, cluster analysis, and principal component analysis) to reduce the number of alarms [27].

2) Pattern analysis based techniques: The key idea of pattern analysis based techniques is that advanced data mining algorithms can extract useful patterns that can be used to formalize alarm suppression rules. Kordic *et al.* [28] grouped alarms using a context-based segmentation approach to find a pattern of correlated alarms. To avoid the disturbing effect that occurs when an alarm triggers other alarms this method was further developed in [29]. This triggering alarm may return before the end of the pattern sequence, therefore the method requires marking a target tag in advance to set the starting point of segmentation. Folmer and Vogel-Heuser proposed an automatic alarm sequence generating algorithm to find the possibilities to redesign the alarm management systems. However, the authors admit the deficiency of the algorithm, as it is not able to compare different alarm sequences even if they differ in only a single alarm in the sequence [30]. This problem can be solved by a modified Smith-Waterman algorithm which utilizes the time stamp information of alarm tags to calculate the similarity index of alarm floods [31]. The algorithm uses these similarity indexes to classify the alarm floods to similar patterns instead of identifying exactly the same sequences. This work was extended with new scoring functions, dynamic programming, backtracking and alignment generation procedures in order to align multiple alarm flood sequences [32]. At the application of dynamic time warping for the pairwise alignment of the sequences it can be also assumed that the alarm sequences are generated from first-order Markov chains [33]. Recent results focuses also on the applicability of the concept. Hu *et al.* proposed a novel framework with the developed toolbox for smart data analytics of alarm and process data and causality analysis [34], while an online pattern matching based reduction of incoming alarm floods was reported in [35].

C. Motivation and outline of the paper

The previously presented detailed overview highlighted that current state of alarm management is static and lacks the detailed probability analysis of extracted suppression rules. In the present paper we would like to introduce a methodology resolving these issues.

To provide more insight into the process dynamics, pattern-based techniques are getting developed. In our work we make a step further, we propose method to handle the temporal relationships among the alarms, control actions and process variables.

The main contributions of our work are:

- The mining of frequent temporal sequences requires a special algorithm. We significantly extend our previous work [36] with the probabilistic interpretation based design of the alarm suppression rules.
- We transform the multi-temporal sequences into Bayes classifiers that can be directly used to define alarm suppression rules.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2797247, IEEE Access

- The real-life applicability is demonstrated via the analysis of an openly available dataset obtained from a laboratory-scale water treatment testbed.
- We extended the benchmark simulator of a vinyl acetate production technology to generate easily reproducible results and further stimulate the development of alarm management algorithms.
- We elaborate a detailed sensitivity analysis to demonstrate the effects of the parameters of the sequence mining algorithm to the number of the identified alarm suppression rules.
- To stimulate further research, the resulted MATLAB codes of the proposed sequence mining and alarm analysis algorithms are openly available at the website of the authors (www.abonyilab.com).

The roadmap of the paper is as follows. The sequence mining problem, the suggested reliability measures, and the pattern mining algorithm are presented in Section II-A, Section II-B and Section II-C respectively. The workflow of the proposed methodology is summarized in Section II-D. We demonstrate the real life applicability and effectiveness of the proposed sequence mining algorithm in the description of process dynamics of a scaled-down version of an industrial water treatment process in III-A. The detailed analysis is based on simulated data. The studied system and its simulator are described in Section III-B. We extended the simulator to make it suitable to study alarm management problems. These results are documented in Section III-C. The structure of the studied temporal database is described in Section III-D. We illustrate the causal connections between alarms in Section III-E. Problematic faults have similar consequences. The statistical measures and the complexity and similarity analysis of the faults are evaluated in Section III-F. The applicability of the extracted alarm suppression rules is studied in a sensitivity analysis in Section III-G. Based on the discussion of the results we clarify the applicability and the limitations of the methodology in Section III-H.

II. Methodology

A. Multi-temporal representation of alarm sequences

Alarm and warning signals can be treated as *states* of the technology. Each state (denoted by s) is represented by $< pv, a >$ data couples, where pv is the index of the process variable and a is the attribute showing the process variable's value related to the alarm and warning limits, such as $a \in \{Low\ A, Low\ W, High\ W, High\ A\}$, where A stands for Alarm and W stands for Warning. For example the description of a state can be represented as follows: $s_e := < Column\ Top\ Temperature, High\ A >$.

An *event* is the time interval in which the defined state occurs, denoted by e. Therefore an event can be represented by a triplet, such as $< s,\ st,\ et >$, where s is the state, which is taking place in a time interval between st starting time and et ending time. Note that each state can correspond to one or more events as well. An event can relate to the state s_e as follows: $< s_e,\ 12,\ 14 >$.

In case an event log is not available, it is necessary to create one, and the time series of the process values have to be converted to an event log database. In the present context, we use the limits of the individual variables to define events. The definition of such thresholds is not trivial, for the conventional approaches see Section I-A, but in most of the cases, process-relevant knowledge and sensitivity tests are used. After the definition of alarm and warning thresholds, we can find the time series of the faulty operation periods, by checking the process variables whether they exceed these limits. With the obtained event start and end points, as it is represented in Figure 2, an event log database can be created by sorting these events into ascending order based on their starting time. Therefore in Figure 2, two events can be defined, $e_1 = < Temperature_{LowA},\ st_{e1},\ et_{e1} >$ and $e_2 = < Temperature_{HighA},\ st_{e2},\ et_{e2} >$, where $Temperature_{LowA}$ and $Temperature_{HighA}$ are the states of the specified object.

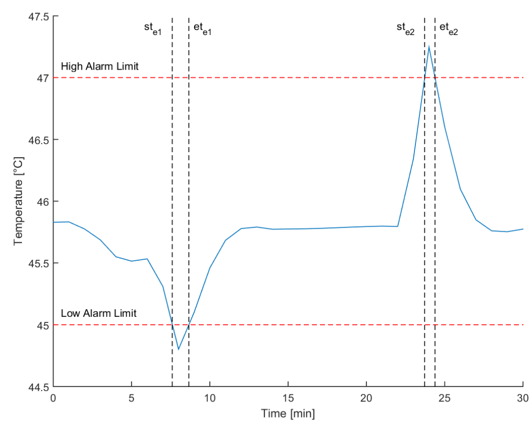

Fig. 2. Example for the time series of a process variable with the lower and higher thresholds indicated

The D_T database shows an example for an event log database as can be seen in Figure 3 and Table II. In the defined temporal database we can distinguish high and low alarm states by adding a one or a two extension to their IDs, respectively (the alarm tags used later are described in Appendix B).

TABLE II
Example D_T event log database

Event id	State id	Starting time	Ending time
e_1	s_1	1	4
e_2	s_2	2	6
e_3	s_3	2	6
e_4	s_1	5	8
e_5	s_4	5	15
e_6	s_2	10	15
e_7	s_5	11	14
e_8	s_2	16	18

For the determination of event correlation it is important to define a *time window*. In the case of a (st_1, et_1) time interval of event e_1, the time lag (window) defined as $C_1 = (st_1, et_1 +$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2797247, IEEE Access

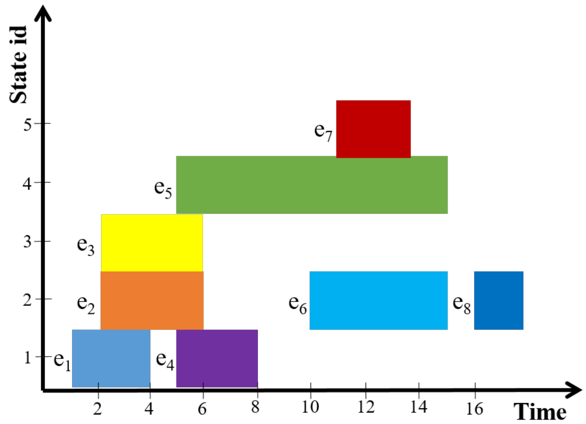

Fig. 3. A visual representation of the events of different (process) states in time. The horizontal axis represents the time and the rows of the vertical axis represent the states. The vertical length of the bars with different colours indicates the length in time of the given event.

$window$) is the window constraint of e_1. If we have an e_2 with the starting point, such that $st_2 \ \epsilon \ C_1$, than e_2 satisfies the window constraint of e_1. The value of the window parameter sets the time distance in which two event can be correlating. Given the D_T database with a time window of 4, e_6 satisfies the window constraint of e_4; some kind of correlation can be defined. Given two events, $e_1, \ e_2$, with the time intervals (st_1, et_1) and (st_2, et_2) respectively, we can define a temporal connection between them, if they are connected by one of the four temporal predicates as follows:

- If $st_1 = st_2$ and $et_1 = et_2$, then $e_1 \ equal \ e_2$,
- If $0 \ \leq st_2 - et_1 \leq \ window$, then $e_1 \ before \ e_2$
- If $st_2 < st_1 < et_1 \leq et_2$, then $e_1 \ during \ e_2$
- If $st_1 \leq st_2 < et_1 < et_2$ or $st_1 < st_2 < et_1 \leq et_2$, then $e_1 \ overlap \ e_2$

We will use the notation E, B, D and O for $equal$, $before$, $during$ and $overlap$ respectively.

Using this approach, we can define temporal instances as $\phi \ := \ e_1 \ \overset{R}{\Rightarrow} \ e_2$, where $R \ \epsilon \ \{ \ E, \ B, \ D, \ O\}$ is a temporal predicate. For example, in the case of D_T temporal database: $e_1 \overset{O}{\Rightarrow} e_2$, $e_2 \overset{E}{\Rightarrow} e_3$, $e_7 \overset{D}{\Rightarrow} e_6$, $e_6 \overset{B}{\Rightarrow} e_8$. We represent the set of n states denoted by $S = (s_1, s_2, ..., s_n)$, and the set of temporal predicates $\{E, \ B, \ D, \ O\}$. A pattern of $k+1$ states is connected with k temporal predicates, is called a k-length temporal pattern (the number of predicates is k), or a $k+1$-state temporal pattern (the number of states is $k+1$) and is denoted as Φ_k. According to this we can have:

- $k \ = \ 0$-length patterns, where $\Phi_0 \ := \ s_0, \ s_0 \ \epsilon \ S$ (we call this trivial pattern with only one sate, e.g. "low temperature", a degenerated temporal pattern)
- $k \ = \ 1$-length pattern is formulated as $\Phi_1 \ := \ (\Phi_0 \overset{R_1}{\Rightarrow} s_1) := (s_0 \overset{R_1}{\Rightarrow} s_1), \ R_1 \ \epsilon \ \{E, \ B, \ D, \ O\}$
- In general $k \geq 2$-length pattern is formulated as $\Phi_k := (\Phi_{k-1} \overset{R_k}{\Rightarrow} s_k) \ := \ (s_0 \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} s_2 \overset{R_3}{\Rightarrow} ... \overset{R_k}{\Rightarrow} s_k), \ R_j \ \epsilon \ \{E, \ B, \ D, \ O\}$, where $j = 1, 2, ..., k$.

If we have a temporal pattern e.g. $\Phi := s_i \overset{R}{\Rightarrow} s_j$ and a temporal instance $\phi := e_{ip} \overset{R}{\Rightarrow} e_{jq}$ where e_{ip} and e_{jq} are the events related to s_i and s_j, respectively, then we call the temporal pattern Φ supported by the temporal instance ϕ.

The patterns with 3 or more states (or 2 predicates) are called multi-temporal patterns. Based on this, given a multi-temporal pattern $\Phi_k := (\Phi_{k-1} \overset{R_k}{\Rightarrow} s_k) := (s_0 \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} s_2 \overset{R_3}{\Rightarrow} ... \overset{R_k}{\Rightarrow} s_k), \ R_j \ \epsilon \ \{E, \ B, \ D, \ O\}, \ j = 1, 2, ..., k$, then a pattern $\Phi' := s_{j1} \overset{R_j 2}{\Rightarrow} s_{j2} \overset{R_j 3}{\Rightarrow} s_{j3} ... \overset{R_j m}{\Rightarrow} s_{jm}$, where $0 \leq j_1 \leq j_2 \leq j_3 \leq ... \leq j_m \leq k$, and $j_1, j_2, ..., j_m$ is a series of sequential natural numbers, then Φ' is called a sequential sub-pattern of Φ. In the D_T example temporal database the $\Phi_1 := s_1 \overset{O}{\Rightarrow} s_2$ is a sequential sub-pattern of the $\Phi_2 := s_1 \overset{O}{\Rightarrow} s_2 \overset{B}{\Rightarrow} s_5$ pattern.

TABLE III
THE D_T EXAMPLE EVENT LOG DATABASE TRANSFORMED INTO D'_T. THE COLUMNS OF THIS CONVERTED DATABASE ARE THE INDIVIDUAL STATES, WHILE THE ROWS BELOW EACH STATE ARE THE TIME INTERVALS IN THAT THE STATE OCCURS.

s_1	s_2	s_3	s_4	s_5
$[1, 4]$	$[2, 6]$	$[2, 6]$	$[5, 15]$	$[11, 14]$
$[5, 8]$	$[10, 15]$			
	$[16, 18]$			

B. Probabilistic interpretation of temporal patterns

The estimation of the number of annunciated alarms and the risk associated with the suppression of a warning signal requires the probability based interpretation of the temporal patterns. The key idea is that the probability of an alarm is proportional to the support of the $k = 0$-length sequence as can be seen in Equation 1, where $supp(s_i)$ stands for the number of supporting events of each state.

$$P(s_i) \backsimeq supp(s_i) \tag{1}$$

We can easily determine the number of the supporting events of states by counting the number of instances under each state in a transformed database, as can be seen in the D'_T temporal database in Table III. The columns of this converted database are the individual states, while the rows below each state are the time intervals in that the state occurs.

To characterise the frequency of the given event, the number of temporal instances can be related to time intervals. In the present approach, the number of temporal instances is normalised by the number of temporal instances of the most frequent state according to Equation 2.

$$support(\Phi) = supp(\Phi)/|E| \tag{2}$$

where $|E| = \underset{j=1,2,...,N}{max}(|E_j|)$, N is the number of states in the D_T temporal database, E_j is the set of events supporting state s_j, $|E_j|$ is the cardinality of E_j (the number of events in E_j). According to Table III, the degenerated pattern, s_1, is supported by two temporal instances, therefore $supp(s_1) \ = \ 2$, and $|E| \ = \ max(2, 3, 1, 1, 1) \ = \ 3$. Then $support(s_1) \ = \ 2/3$. The support value is used to measure the frequency of each pattern, therefore it is essential in the determination of frequent patterns. Given a defined support threshold ($minSupp$), determined as an input parameter of the mining algorithm, the Φ sequential pattern is called frequent pattern if $supp(\Phi) \geq minSupp$. The main goal in sequential pattern mining is finding all the frequent patterns for a given $minSupp$.

A consequence of this, that the $k > 0$-length patterns are proportional as well to the degenerated pattern's (k=0-length sequences) support. The probability of a transition between two states is consequent to the support value of the appropriate k=1-length sequence.

Alarm suppression rules can be defined when the temporal instances of the states are not independent, so the probability of a $P(\Phi_k) \coloneqq P(s_0 \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} s_2 \overset{R_3}{\Rightarrow} ... \overset{R_k}{\Rightarrow} s_k)$ sequence can be calculated by the chain rule:

$$
\begin{aligned}
P(\Phi_k) &= P(s_1|s_0) \times P(s_2|s_0 \overset{R_1}{\Rightarrow} s_1) \times ... \\
&... \times P(s_k|s_0 \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} s_2 \overset{R_3}{\Rightarrow} ... \overset{R_{k-1}}{\Rightarrow} s_{k-1})
\end{aligned} \tag{3}
$$

Equation 3 shows that, the probability of the occurrence of a k-length sequence can be calculated from the occurrence of the $k-1$-length sequence and from the conditional probability denoted by $P(s_k|\Phi_{k-1})$, where Φ_{k-1} is the $k-1$-length sub-pattern of Φ_k in a given database (therefore, based on the definition Φ can be written as $\Phi_{k-1} \overset{R_{k-1}}{\Rightarrow} s_k$):

$$
P(s_k|\Phi_{k-1}) = \frac{P(\Phi_k)}{P(\Phi_{k-1})} = \frac{supp(\Phi_k)}{supp(\Phi_{k-1})} \tag{4}
$$

It should be noted, that this equation represents the confidence of the rule for transition between states, denoted in the $\Phi_k \coloneqq (\Phi_{k-1} \overset{R_k}{\Rightarrow} s_k)$ k-length sequence:

$$
conf(\Phi_k) = \begin{cases} \frac{supp(\Phi_k)}{supp(\Phi_{k-1})} \times conf(\Phi_{k-1}) & R_k \neq D \\ \frac{supp(\Phi_k)}{supp(s_k)} & R_k = D \\ 1 & \Phi_{k=0} \end{cases} \tag{5}
$$

The support gives a measure of the relevance of the given sequence and the confidence measures its reliability. It is important to mention, that $conf(\Phi_k)$ therefore gives the confidence of the whole Φ_k sequence, while the confidence of the transition between states denoted as $conf(\Phi_{k-1} \overset{R_k}{\Rightarrow} s_k) = P(s_k|\Phi_{k-1})$ as described before, thus the two quantity does not equal.

When information about the type of fault is also available, then the sequences of events can be defined as the consequence of a c_j type fault. Therefore the probability of this event sequence can be defined by the $P(c_j \overset{R_0}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} ... \overset{R_k}{\Rightarrow} s_k)$ probability. According to the principle of the alarm suppression rules, the occurrence of a failure and the sequence of the occurring events are not independent. Therefore the probability of the whole sequence is described by the chain rule:

$$
\begin{aligned}
P(c_j \overset{R_0}{\Rightarrow} \Phi_k) &= P(c_j) \times P(s_1|c_j) \times P(s_2|c_j \overset{R_1}{\Rightarrow} s_1) \times ... \\
&... \times P(s_k|c_j \overset{R_1}{\Rightarrow} s_1 \overset{R_2}{\Rightarrow} ... \overset{R_{k-1}}{\Rightarrow} s_{k-1})
\end{aligned} \tag{6}
$$

It is essential to examine the dependency of the antecedent and consequent parts of the rules. To evaluate the dependence, we can define a measure called *improvement*. The improvement can be calculated by the fraction of the confidence of the transition between states and the support of the added state:

$$
\begin{aligned}
Improvement(\Phi_k) &= \frac{P(s_k|c_j \overset{R_0}{\Rightarrow} \Phi_{k-1})}{P(s_k|c_j)} = \\
&= \frac{conf((c_j \overset{R_0}{\Rightarrow} \Phi_{k-1}) \overset{R_k}{\Rightarrow} s_k)}{support(s_k)} = \\
&= \frac{support(c_j \overset{R_0}{\Rightarrow} \Phi_{k-1} \overset{R_k}{\Rightarrow} s_k)}{support(c_j \overset{R_0}{\Rightarrow} \Phi_{k-1}) \times support(s_k)}
\end{aligned} \tag{7}
$$

C. The multi-temporal mining algorithm

To generate informative patterns from temporal databases we utilised the algorithm published in [36]. The implementation of the method is fine-tuned to make it more convenient and applicable for alarm management purposes. Since we are interested in patterns that can be useful for alarm suppression, the resulted patterns were also filtered based on the $minConf$ value. To stimulate further research, the resulted MATLAB codes of the proposed sequence mining and alarm analysis algorithms are openly available at the website of the authors (www.abonyilab.com).

Algorithm 1 Multi-Temporal Mining method

Require: D_T: Temporal database
  $minSupp$: Support threshold
  $minConf$: Confidence threshold

```
F_0 = 0
for every state s in D_T do
    if supp(s) ≥ minSupp then
        F_0 = F_0 ∪ {s}
    end if
end for
C_1 = {s_i =R=> s_j | s_i ∈ D_T, s_j ∈ F_0}
i=1
while F_{i-1} ≠ 0 do
    F_i = 0
    for every sequence Φ ∈ C_i do
        if supp(Φ) ≥ minSupp and
           conf(Φ) ≥ minConf then
            F_i = F_i ∪ {Φ}
        end if
    end for
    C_{i+1} = {s_i =R=> s_j | s_i ∈ F_i, s_j ∈ F_0}
    i=i+1
end while
```

The pseudo-code of the described algorithm can be seen in Algorithm 1. The $k+1$ patterns are generated step by step by combining all the k-length patterns (degenerated patterns at the beginning of the mining algorithm) and the events in the D_T temporal database while checking their relationship to the defined time window. The appearing sequences are stored until we cannot create more sequences with $k+1$ elements. We examine them concerning the specified $minSupp$ and $minConf$ conditions, as only the sequences meeting these criteria can be defined as frequent and reliable patterns. Patterns violating these conditions are discarded. We repeat this process until we cannot create more sequences with $k+1$ element.

D. Workflow of the methodology

Figure 4 shows the workflow of the methodology. Firstly, we have to define the events that we would like to examine. We can use process log files of alarms or we can create discrete events from time series. There are cases when we do not have log files from previous operation periods, but nowadays many of the industrial plants have a detailed dynamic simulator either for modeling or operator training purposes. In the case of the examination of unknown faults or in the case of the lack of process log files, we can implement malfunctions in these simulators and record the following events. With the use of this database, we can apply the described multi-temporal mining algorithm and create the frequent patterns of events. Here we have two options.

If we know the root cause of the events, therefore the database contains the malfunctions as well, we can characterize the faults and investigate the spillover effect of them.

In the case of industrial alarm log files we usually do not know the root cause of the analyzed events, but in these cases we can still analyze the correlation of the alarm pairs that frequently occur together through the support value of the two-event-long sequences. In addition, in these cases we can calculate the confidence of transition between events and define alarm suppression rules.

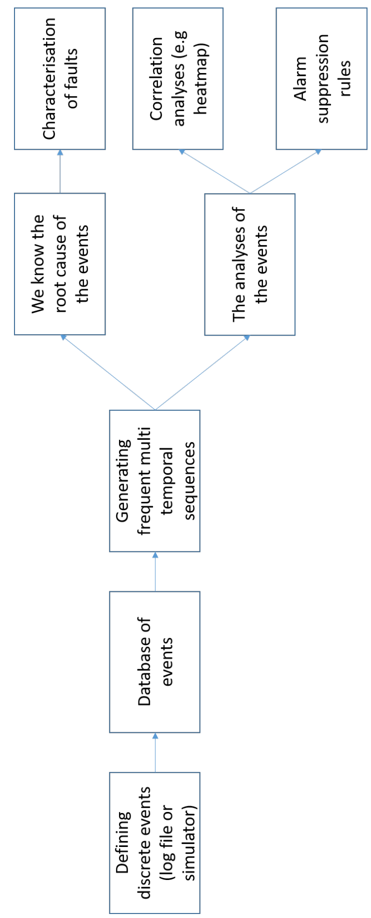

Fig. 4. The schematic workflow of the proposed methodology

III. RESULTS AND DISCUSSIONS

A. The multi-temporal representation of a dynamic process

The proposed multi-temporal sequences describe process dynamics in a reduced-information-content form. This representation is useful when we would like to get a deeper insight into the process dynamics based on historical data and only the (alarm) log databases are available for longer time periods.

We would like to illustrate the effectiveness of this representation using an openly available dataset, called the Secure Water Treatment testbed (SWaT) made by the iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design [37]. The testbed was designed to support the research of cyber-physical systems and represents a scaled-down version of an industrial water treatment plant. The data collection process lasted for 11 days. In the first seven days the testbed operated without malfunctions, so this part of the dataset is available for the determination of normal operating patterns, and in the rest four days SWaT was under attack, this part of the dataset can be considered the faulty operation. In total, 946.722 samples comprising of 51 attributes were collected over the 11 days of the collection process including signals from sensors (tank levels, flow meter's, water properties, pressure) and actuators (water pumps, motorized valves and ultraviolet dechlorinators).

We demonstrate the applicability of multi-temporal sequences on a simple example related to the raw water storage tank at first part of the testbed. The tank has two sensors measuring the inflow to the tank and the water level inside the tank and two binary control inputs related to the switching on or off the inflow and the outflows. In case of the measured sensor variables, events are recorded when the given variable drops below the lower or increases above the higher threshold of the variable. The thresholds are the margins of the lower and upper 5 % of operation range. In case of the actuators, the events were defined based on the binary control input signals. Figure 5 shows a normal operation period with the indication of the occurring discrete events and their time intervals.

The proposed multi-temporal mining algorithm can determine the frequently occurring operation patterns. With values of 0.2 of support and confidence thresholds and 2-minute-long time window the algorithm finds the pattern of operation (it finds reasonable patterns with other input values as well, but with these parameters it gives a single sequence that nicely represents the operation pattern): $Level\ High \overset{D}{\Rightarrow} Outflow\ OFF \overset{O}{\Rightarrow} Inflow\ OFF \overset{O}{\Rightarrow} Measured\ Inflow\ Low \overset{B}{\Rightarrow} Level\ Low$.

The detected sequence describes well the normal operation pattern of the storage tank. During the outflow from the tank is off the level will reach the high state (notice that the during temporal predicate starts with the shorter event, which is incorporated into the other event). Thanks to the high liquid level, the outflow from the tank is turned on (this state is the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2797247, IEEE Access

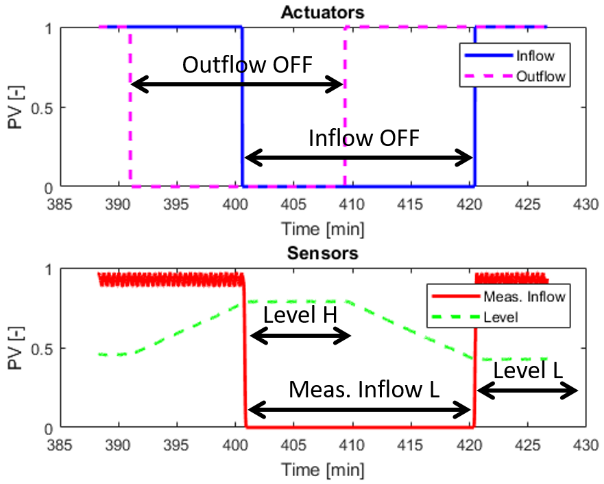

Fig. 5. The actuator (inflow and outflow) and sensor (measured inflow and tank level) values of the raw water storage tank of Secure Water Treatment testbed made by iTrust. The time intervals of the recorded discrete events are indicated by black arrows with the name of the recorded event above it: Outflow off, Inflow off, Level High, Measured Inflow Low, Level Low. The horizontal axis shows the time, while the vertical axis shows the related process value (PV).

complementer of the turned-off state, therefore it is not defined as an event) and the inflow to the tank is turned off, which is resulted in low measured inflow as well. Finally, we will reach the low-level state.

The application example demonstrates the effectiveness of the multi-temporal sequences in the description of dynamic processes. If we consider the alarm and warning signals of a process the indicators of process state, we can describe the frequently occurring operational patterns with the use of the described algorithm. However, to demonstrate the applicability of the proposed sequence-based representation in the field of alarm suppression in a more detailed form, we designed a simulation example that we present in the following section.

B. Description of the Vinyl Acetate Process Simulator

The drawback of the alarm management studies is that the results are not reproducible. To provide a benchmark for further research, we developed a well-documented simulation study. In the following, the description of the simulator and the generation of log files are described in details.

Since the MATLAB simulator of the vinyl acetate (VAc) process contains 27 controlled and 26 manipulated variables, this benchmark system is complex enough to test alarm management and fault diagnosis algorithms [38]. The process contains 10 basic unit operations and seven chemical components (ethylene (C_2H_4), oxygen (O_2) and acetic acid (HAc, CH_3COOH) are converted to vinyl acetate ($CH_2{=}CHOCOCH_3$) with water (H_2O) and carbon dioxide (CO_2) byproducts, ethane (C_2H_6) enters with the ethylene feed as inert). The scheme of the production technology can be seen in Figure 6.

The vaporizer is implemented as a well-mixed unit with seven components, with a gas input containing the mixture of the fresh C_2H_4 stream and the absorber vapor effluent, and a

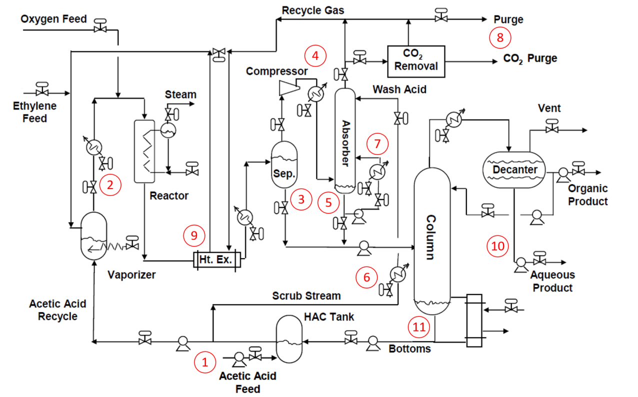

Fig. 6. Modified version of the figure presented by Chen *et al.* [38]. Flow chart of the VAc production technology (The numbers in circle (red) show the type of the implemented fault, see Table IV)

liquid input from the HAc tank. The catalytic plug flow reactor is implemented as a distributed system with ten elements in the axial direction. Inside of the reactor the exothermic reactions of Equation 8 and 9 take place.

$$\begin{aligned} C_2H_4 + CH_3COOH + 1/2\ O_2 \rightarrow \\ \rightarrow CH_2 = CHOCOCH_3 + H_2O \end{aligned} \tag{8}$$

$$C_2H_4 + 3\ O_2 \rightarrow 2\ CO_2 + 2\ H_2O \tag{9}$$

The process contains a feed-effluent heat exchanger (FEHE), where a small time constant is added to the exit temperature sensors to simulate the dynamics of the process. After a pressure letdown valve (which is not shown in Figure 6), the effluent is led to a separator. The separator is modeled as a partial condenser, and the leaving liquid and gas stream flow rates are calculated with a steady state equilibrium-flash equation. The gas stream enters to the bottom part of the absorber unit after compression. The absorber is divided into two parts. The liquid stream of the bottom part is the liquid stream leaving the top part and a circulation stream. The gas inlet of the top part is leaving from the bottom part of the absorber, while the liquid inlet comes from the HAc tank. There is a CO_2 removal system implemented after the absorber. A gas removal system takes place before the azeotropic distillation tower to remove all the light components from the inlet of the tower which comes from the bottoms of the separator and absorber units. It is modeled as an ideal component separator, which completely separates the gas components (O_2, CO_2, C_2H_4, C_2H_6) to send them back to the inlet of the compressor, while the liquid stream (VAc, H_2O, HAc) enters the distillation tower. The column is highly nonlinear, with 20 theoretical stages, whose liquid holdup can vary. After the condenser, a decanter is implemented for the separation of the liquid phases. The liquid recirculation stream and the HAc inlet stream are mixed in the HAc tank.

C. Defining and implementing faults

The original MATLAB model of the simulator contained five disturbances (1.) step change in the composition of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2797247, IEEE Access

C_2H_6 in the fresh C_2H_4 feed stream from 0.001 to 0.003 mole fraction, 2.) loss of column feed for 5 minutes, 3.) loss of fresh HAc feed stream for 5 minutes, 4.) Loss of fresh O_2 feed stream, 5.) an analyzer is off-line (except the O_2 analyzer)) [38]. Károly and Abonyi [36] studied the effects of three malfunctions (1.) Loss of *HAc* feed, 2.) Loss of O_2 feed and 3.) Loss of column feed), and studied the effect of product changes with the use of the so-called Operating State Matrix (OSM), containing the randomly generated values of the following manipulated variables:

- Operating state start time (min)
- Operating state end time (min)
- Setpoint of the reactor output temperature (150-165 °C)
- H_2O composition in the column's bottom (9 - 18%)
- Vaporizer feed (2.2 - 2.6 $\frac{kmol}{min}$)
- Change of the C_2H_6 concentration of the C_2H_4 feed from 0.1% to 0.3% (not range based, only two states)

For the efficient and precise work with the VAc simulator, we revised the simulator and inserted additional faults related to the controllers, as the manipulated value of the process variable remained at a constant value for a specified time. The duration of these faults follows a lognormal distribution.

The values of the manipulators in the case of faults can be seen in Table IV.

In the case of injected malfunctions an Error Event Matrix (EEM) is created randomly with the following content:

- Fault occurrence time (min)
- Fault end time (min)
- Fault ID for the identification of fault

Each malfunction type is chosen based on a uniform distribution and the duration is determined using the lognormal distribution described before. The occurrence of each fault is generated between the $10th$ and $T_{simend} - 100th$ minute of the simulation. The OSM and EEM record the input data of the randomly generated experiments.

D. Definition of process alarms and generation of their log files

The alarm limits were determined based on a simulation containing 150 different process states (according to the Operator State Matrix, OSM), with no faults implemented. The overcome of the minimal and maximal values for each process variables were used for the determination of events on each process variables.

First of all, with the utilization of the VAc simulator, time series were created for each fault in separate runs. All of the faults were implemented in 200 different, 1-hour-long operating states with lognormal time interval distribution as it was described before. The time series were then transformed to event log database using the defined alarm thresholds. When the aim of the analysis is the improvement of HAZOP analysis and the characterization of the consequences of faults, the known faults should also be included as events in this database. This database forms the input of the multi-temporal mining algorithm. The obtained sequences were filtered, as we were interested only in the sequences containing the implemented faults, at the beginning of the characteristic sequences. When

TABLE IV

MANIPULATOR VALUES IN CASE OF FAULTS (MEASUREMENT UNITS ARE NEGLECTED AS THE ORIGINAL CODE AND DOCUMENTATION DOES NOT CONTAIN THEM NEITHER)

Tag of fault	Controlled variable	Manipulated variable	Man. variable value in case of fault
1	HAc Tank Level	HAc fresh feed Flow Rate	0.3
2	Heater Exit Temp	Reactor Preheater Heat Flow	2000
3	Separator Level	Separator Liquid Exit Flow Rate	0
4	Compressor Exit Temp.	Compressor Heater Heat Flow	20000
5	Absorber Level	Absorber Liquid Exit Flow Rate	0
6	Circulation Stream Temp.	Absorber Scrub Heat Flow	5000
7	Scrub Stream Temp.	Circulation Cooler Heat Flow	1000
8	C_2H_6 in the Gas Recycle	Purge Flow Rate	0
9	FEHE Hot Exit Temp.	Bypass Flow Rate	0.4
10	Decanter Aqueous Level	Aqueous Product Flowrate	0
11	Coloumn Bottom Level	Coloumn Bottom Exit Flowrate	0

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2797247, IEEE Access

the aim of the analyses is the reduction of incoming alarms, the faults are not included in the event database, as we are only interested in the frequent operating patterns. In this case, we generate event sequences from this database.

We tested how the number of generated sequences depends on the window parameter, as this reflects temporal causality of the implemented faults. The results were calculated with the use of the database containing the faults as well, using 0.1 and 0.2 of support and confidence values, respectively. Figure 7 illustrates that the number of generated sequences show a saturation as the function of the time window. This is reasonable as the effect of a fault decays in time in optimal cases.

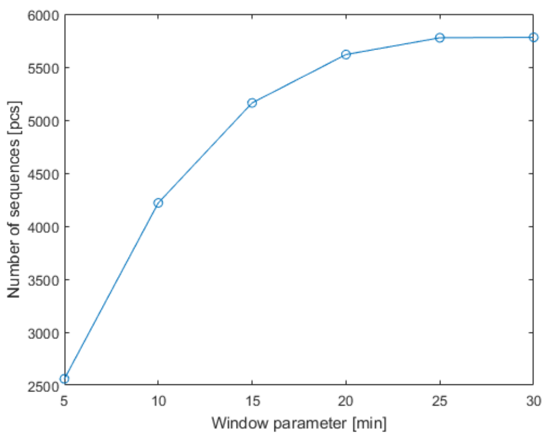

Fig. 7. The number of frequent sequences shows a saturation with increasing time window

The number of generated sequences highly depends on the chosen support and confidence threshold values as well. The horizontal axis of Figure 8 shows the value of the investigated threshold, the other threshold is set to 0.2 during the analysis, the window parameter is set to 20 minutes. The first vertical axis on the left shows the number of sequences for the support threshold analyses, while the second vertical axis on the right shows the results of the confidence threshold analysis. According to Figure 8 the number of frequent sequences decreases significantly as the function of the increasing threshold values.

It is important to mention that the confidence of transition between states is the most critical measure that determines the applicability of an alarm suppression rule. For the presented results 20-minute time window, 0.1 and 0.2 support and confidence threshold values were used respectively, however, in this section we will also demonstrate the effects of these parameters in a detailed sensitivity analysis.

E. Correlation of occurring alarms

Advanced alarm management algorithms are often based on correlated alarms, as correlation analysis can explore the hidden connections between alarms. This casual connection can be illustrated by a heatmap, visualizing the intensity of the alarm tags occurring together in the defined order. Figure 9 shows the support value of the $k = 2$-length sequences. The

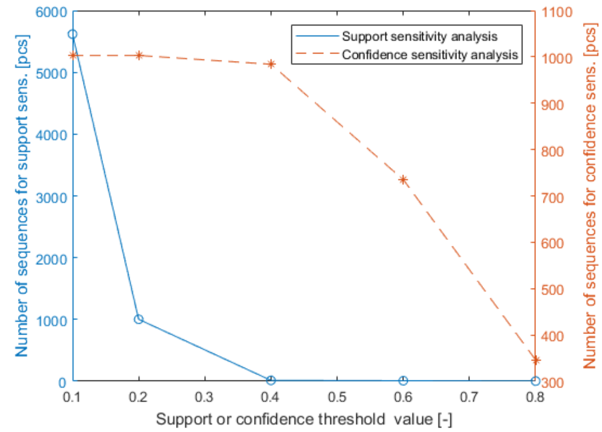

Fig. 8. The number of generated sequences in case of different support and sensitivity threshold values. The investigated threshold value is indicated on the horizontal axis, the other threshold is set to 0.2. The first vertical axis on the left shows the number of sequences for the analysis of the support threshold, while the second vertical axis on the right shows the results of the confidence threshold analysis.

support values are calculated with the ignorance of temporal relationships, therefore the support values of the same alarm pairs connected with different temporal predicates are summarized. The database without the faults was used for the analysis.

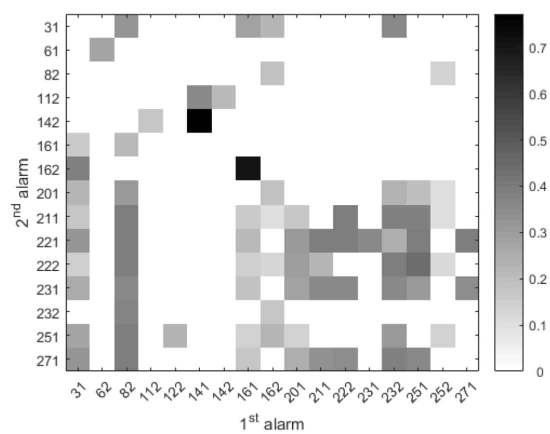

Fig. 9. The heatmap shows the support values for the $k = 2$-length sequences. The support values are calculated with the ignorance of temporal relationships, therefore the support values of the same alarm pairs connected with different temporal predicates are summarised. Axis x shows the first, while axis y shows the second alarm tag of the sequence.

Figure 9 highlights that some alarms can be followed by more alarm tags. Due to this uncertainty, there is a need for longer sequences, because the knowledge of the history of states and the transition between them can increase the probability of the next event (see the improvement measure in Equation 7).

F. Characterization of faults based on their complexity

The sequence database holds an enormous amount of information about the dynamics of the process. Using this database

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2797247, IEEE Access

we can look for problematic or complicated faults, which are hard to identify based on their particular sequences, or for faults that cause a spillover effect on the process. With this, we can help the improvement of safety-related analysis methods, e.g., hazard and operability analysis. Creating such studies is usually highly labor-intensive, as a multidisciplinary team creates scenarios for possible malfunctions in brainstorming meetings. The team tries to identify all the potential process faults with their consequences, and give recommendations for handling them each by each. However, this technique requires long hours of human work and relies entirely on the expert knowledge of team members. With the use of the proposed methodology, the time and workforce requirement of such techniques can be reduced.

To analyze the consequences of specific faults, the sequences of the event database containing the faults as well were filtered to the ones containing only one fault at the beginning of the sequence. The statistic evaluation of the obtained sequences was carried out as can be seen in Table V (the No. of faults follows the order presented in Table IV). The higher number of characteristic sequences reflects that it is harder to identify the root cause of the malfunction, while longer sequences mean that the given fault causes a spillover effect on the process triggering a series of alarms. The support and confidence values show the measure of how relevant and reliable is the obtained information, respectively.

Table V shows that most of the faults have only a few types of following frequent alarm patterns, the only exception is Fault 3, the fault of the separator liquid exit flow rate.

Faults with fewer types and shorter length of characteristic sequences are easier to identify, and the information held by these sequences can usually be treated as trivial by the process experts. Faults with longer length of characteristic sequences are much harder to predict with expert knowledge. Figure 10 visually illustrates the characterization of faults based on their number of characteristic sequences and maximal length of them. The horizontal axis shows the number of sequences that can follow the given malfunction, while the vertical axis shows the maximal length of these sequences. The figure does not contain faults that do not generate frequent alarm sequences and does not contain Fault 3 neither, as the number of possible sequences after this malfunction has a higher order of magnitude. The more complicated sequences can be seen in the upper right quadrant of the graph, while the ones at bottom left corner can be considered as a more easily identifiable malfunction.

The complexity of the analysis of alarm sequences can be well illustrated with the examination of the time distribution of sequences, according to Figure 11. If we consider only the simple time distribution of single alarms, as can be seen in the upper two graphs of Figure 11, we can see that the alarm time distribution can vary in time and does not definitely follows the lognormal time distribution of the implemented faults. In the case of longer sequences, as it is presented in the bottom four graphs of the figure, experience shows longer time periods with the increasing sequence lengths, which is probably caused by the spillover effect of the processes.

It is also interesting to examine how hard is to distinguish

TABLE V
THE STATISTICAL EVALUATION OF THE ALARMS AND SEQUENCES

Tag of Fault	Number of Characteristic Seq's	Max. Length of Seq's	Max. Confidence of Seq's	Max. Support of Seq. with Max. Conf.	First Event after Failure in Seq. with Max. Conf.	Length of Following Seq. after Fault
1	7	3	0.975	0.377	162	1
2	3	2	1	0.387	61	1
3	1357	8	1	0.387	82	1
4	9	3	1	0.387	112	1
5	5	2	0.96	0.371	122	1
6	3	2	1	0.387	141	1
7	3	2	1	0.387	161	1
8	0	0	-	-	-	-
9	0	0	-	-	-	-
10	1	1	1	0.387	232	1
11	1	1	0.445	0.172	242	1

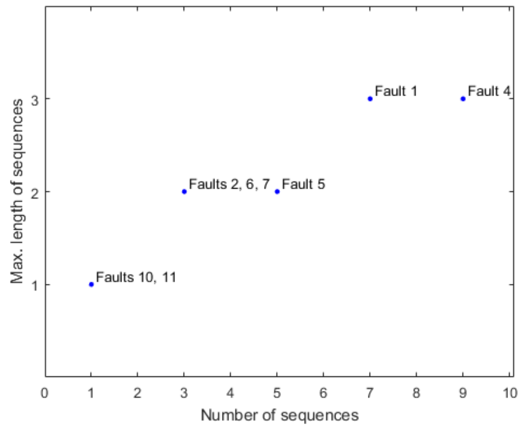

Fig. 10. The visual classification of sequences based on their complexity (the No. of faults follows the order presented in Table IV). The more complicated faults are at the upper right corner of the figure, as these faults can be followed by different and often long series of events. The easily identifiable alarms are at the bottom left corner consequently. The figure does not contain faults that do not generate frequent alarm sequences and Fault 3, as the number of possible sequences after this malfunction has a higher order of magnitude.

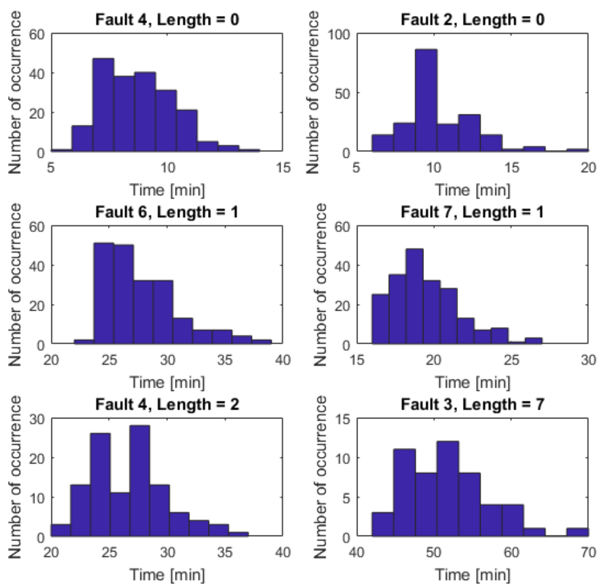

Fig. 11. The time distribution of selected sequences. Even the time distribution of shorter sequences presented at the upper graphs can vary significantly in time, but the spillover effect of the processes is well illustrated by the longer sequences in the bottom graphs. $k = 0$-length patterns are degenerated patterns with only one state, without temporal predicate, as a pattern of $k+1$ states is connected with k temporal predicates, is called a k-length temporal pattern.

different faults based on their caused alarm types. The difficulty comes when the consequences of these faults are similar. We defined a similarity measure to investigate this problem by comparing the annunciated alarm tags caused by each fault, this measure is presented in Equation 10.

$$Sim_{i,j} = \frac{|E_i \cap E_j|}{|E_i \cup E_j|} \tag{10}$$

Table VI shows the counted (nonzero) similarity ($Sim_{i,j}$) indexes. According to this, the problematic (similar) faults from the view of root cause analysis are the 1-7 and 4-6 fault pairs, therefore the first pair is the fault of HAc fresh feed flow rate and the fault of circulation cooler heat flow, while the second pair is the fault of the compressor heater heat flow and the fault of the absorber scrub heat flow.

TABLE VI
THE COUNTED (NONZERO) SIMILARITY INDEXES OF EACH FAULT PAIRS. ACCORDING TO THE RESULTS, THE 1-7 (THE FAULT OF HAC FRESH FEED FLOW RATE AND THE FAULT OF CIRCULATION COOLER HEAT FLOW) AND 4-6 (THE FAULT OF THE COMPRESSOR HEATER HEAT FLOW AND THE FAULT OF THE ABSORBER SCRUB HEAT FLOW) FAULT PAIRS ARE HARD TO DISTINGUISH BASED ON THE INCOMING ALARMS.

Fault number	Fault number	Similarity measure
1	3	0.231
1	7	0.667
3	5	0.071
3	7	0.154
3	10	0.077
4	6	0.667

The distinction of these similar faults can be carried out with the inspection of the appearing alarms (events) as it is presented in Table VII. In both fault pairs, one of the faults (Fault 1 and 4) holds one more alarm type, which can be used for the identification of the root cause of the events.

TABLE VII
THE ALARMS (EVENTS) APPEARING IN THE CASE OF SIMILAR FAULTS (THE NAME OF ALARM TAGS CAN BE SEEN IN TABLE IX)

Fault Tag	Alarms
1	31, 161, 162
4	112, 141, 142
6	141, 142
7	161, 162

The causal relationship of Faults 4 and 6 is obvious if we take into consideration the physical meaning of them: Fault 4 is the manipulation of the compressor heater heat flow, while Fault 6 is the manipulation of absorber scrub heat flow. Both of them controls the temperature of the given equipment, and they are located close to each other. Both of them triggers the high and low alarm of the circulation stream temperature, but Fault 4 also causes a high alarm for the compressor exit temperature. Fault 1 is the manipulation of HAc fresh feed flow rate, and Fault 7 stands for the circulation cooler heat flow. Similarly to the latter, the lack of HAc storage can cause problems with the scrub stream temperature as well, as in this case the flow rate of the scrub stream decreases and the applied heat flow can over or under heat it. The low HAc feed causes low alarm on the HAc tank level as well.

In case of similar processes, the first appearing alarm can be suggestive of the root cause as well. Table VIII shows the number of the type of first alarms appearing after the occurrence of the process faults. Based on this fault types 10 and 11 can be identified relatively obviously, but Fault 1, 4 and especially 3 is hard to determine as they can have different alarms right after the malfunction.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2797247, IEEE Access

TABLE VIII
THE NUMBER OF THE FIRST ALARMS APPEARING AFTER EACH FAULTS

Fault Tag	First alarm types
1	3
2	2
3	10
4	3
5	2
6	2
7	2
8	0
9	0
10	1
11	1

G. Development of alarm suppression rules

The fundamental idea of alarm suppression is that incoming alarms with a high confidence should not be annunciated, and the reduced number of alarms can reduce the workload of the operators. Sequence-based alarm suppression rules could be more beneficial than simple correlation-based methods since the information about the past events can increase the probability of the occurrence of the next event. The transition between the states of the process can be predicted well with the use of the conditional probability described in Equation 4.

For alarm suppression purpose we do not need to know the fault that caused the given sequence, it is enough to explore the alarms occurring together and obtain their sequence of occurrence. We know the conditional probability of the transition of a Φ_{k-1} sequence to the s_k state, or in this case alarm, therefore we have to determine a confidence threshold for this last transition at the end of the sequence in order to define alarm suppression rules. To assure a high rate of confidence of state transition at the last transition of the sequences we set this threshold to 0.8. E.g. the sequence rules of ten alarms (with tags 31, 82, 141, 162, 211, 222, 231, 232, 251, 271) meet these criteria, Figure 12 illustrates the distribution of the length of these sequences.

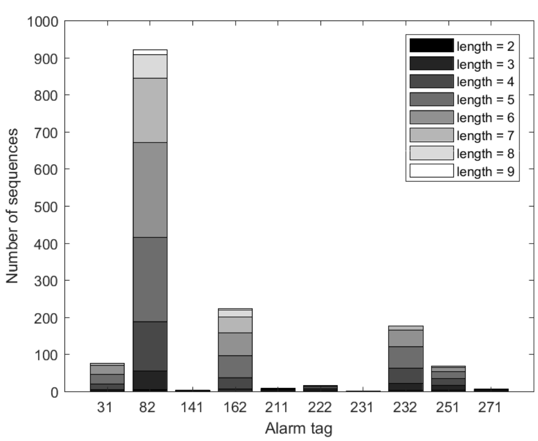

Fig. 12. The number and length of sequences applicable for the suppression of the given alarms

Alarm 82 has a significantly high number of applicable sequences, while for example the alarm 141 and 231 have only a few.

We can examine the ratio of the suppressed and nonsuppressed alarms as can be seen in Figure 13.

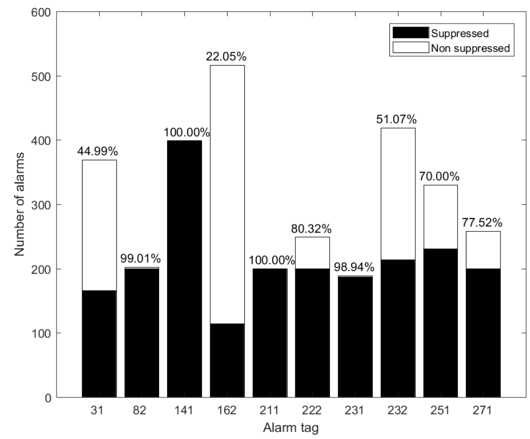

Fig. 13. The ratio of the suppressed and nonsuppressed alarms, the percent of the suppressible alarms as the ratio of all occurring alarms of the given tag is given at the top of each bar.

It is interesting to see that alarm 141 had only four applicable sequences, but all of the alarms could be suppressed with the application of these sequence rules.

H. Limitations

Although in most of the cases the effectiveness of alarm management is measured based on the alarm rates, many studies have shown that the real problem is related to the informativeness of alarms [7], [8], [9], [10]. The concept of our method is that predictable alarms do not bring new information to the operators and the predictability can be evaluated based on the confidence of the alarm suppression rules. Although the number of properly suppressed alarms can be estimated based on the support and the confidence of the rules, these variables should be tuned by making a tradeoff between suppression power and fault diagnostic sensitivity.

The critical question of the present methodology is the sensitivity of the alarm suppression for the window, and for the support and confidence thresholds. The number of generated sequences are investigated, the results are presented in Section III-D. However, it is important to mention that these parameters influence the amount of generated sequences, but the predictive accuracy of the presented method is just the question of the confidence of the last transition in these sequences. This gives the probability of the occurrence of the last alarm, which can be suppressed. If we choose a confidence threshold for this last transition of 0.8 to suppress alarms, we ostensibly miss the 20 % of times that the alarms occur by themselves. For the handling of this issue, during the online application in the future, the times when not the predicted alarm happens the system will display it and only the correct predictions will be suppressed. According to these limitations, the generated alarm suppression rules must be revised by a safety expert who supervises the transformation of the extracted sequences into alarm suppression rules, since evaluating the risk related to a wrongly suppressed alarm requires detailed process relevant knowledge.

IV. Conclusion

Alarm management is a crucial task of improving process safety and reduction of operator workload. The log files generated by process control systems can provide valuable information for the exploration of causal relationships between alarms signals, which can be used to form alarm suppression rules and support root cause analysis. To incorporate the dynamics of the process into the identification of the alarm suppression rules, we proposed a multi-temporal sequential pattern mining-based algorithm. The analysis of a laboratory-scale water treatment system showed that the algorithm could efficiently map the operational pattern of the raw water storage tank from the set of discrete events. Since we wanted to provide a more sophisticated and detailed analysis of the applicability of the method, we extended the simulator of a vinyl-acetate production technology to support and ensure the reproducibility of our results and motivate future research of alarm management and fault diagnosis techniques.

Based on well-documented scenarios we elaborated a detailed sensitivity analysis to evaluate the effects of the parameters of the algorithm (time window, support and confidence threshold values) on the number of suppressed alarms and prediction accuracy. The results of sequence time distribution analysis highlighted that a malfunction has a spillover effect on the process, which cannot be neglected during the definition of alarm thresholds and general hazard analysis principles. The proposed probabilistic interpretation of the model helps to evaluate how suppressible are the alarms. The tuning of the parameters makes a trade-off between the number of suppressed alarms and the confidence in the prediction accuracy. In the studied dataset ten alarm tags met the chosen 0.8 value of state transition threshold and using the identified alarm suppression rules approximately 67.4 % of these alarms could be suppressed.

The results confirmed that the proposed methodology is applicable in evaluating similarities of the faults and the significance and accuracy of the alarm suppression rules extracted from the frequent temporal patterns.

Appendix A
Abbreviations

HAZOP - hazard and operability study
EEMUA - Engineering Equipment and Materials Users' Association
m - mean of the lognormal distribution
v - variance of the lognormal distribution
μ - the mean of the associated normal distribution
σ - the standard deviation of the associated normal distribution
T_{simend} - the end time of the simulation
OSM - Operating State Matrix
EEM - Error Event Matrix
s - *state* of the technology
pv - index of the process variable
a - the attribute showing the process variable's value related to the alarm and warning limits, such as $a \in \{Low\ A, Low\ W, High\ W, High\ A\}$
A - alarm
W - warning
e - event
st - starting time of an event
et - ending time of an event
D_T - an example for an event log database
$window$ - *time window* for the determination of event correlation
E - equal temporal predicates between events
B - before temporal predicates between events
D - during temporal predicates between events
O - overlap temporal predicates between events
R - arbitrary temporal predicate between events
Φ - a temporal pattern of events
ϕ - a temporal instance of events
Φ_k - notation of a k-length temporal pattern
$support(\Phi)$ - support value of a sequence
$supp(\Phi)$ - supporting events of a sequence
$|E|$ - the maximal number of events supporting each states in the D_T temporal database
N - the number of states in the D_T temporal database
$minSupp$ - the defined support threshold of the mining algorithm
$minConf$ - the defined confidence threshold of the mining algorithm
$conf(\Phi)$ - the degree of confidence for the given sequence
$P_{F,1,2,3}$ - the probability of the occurrence of the given fault
PV - process value
$Sim_{i,j}$ - Similarity measure of fault type i and j

Appendix B

During the numbering of the different alarm tags, each of the monitored variables were numbered in an ascending order from 1 to 27. The alarm tags were determined using Equation 11.

$$s_{id} = No.\ of\ variable \times 10 + a \tag{11}$$

The $No.\ of\ variable$ refers to the number in order of the given variable in the described ascending order, while a is the attribute showing the process variable's value related to the alarm and warning limits, here:

- $a = 1$, if the given alarm is a low alarm
- $a = 2$, if the given alarm is a high alarm

The obtained alarm tags can be seen in Table IX.

Acknowledgment

The research has been supported by the National Research, Development and Innovation Office NKFIH, through the project OTKA 116674 (Process mining and deep learning in

TABLE IX
THE NOTATION OF THE DIFFERENT ALARM TAGS

Name of Alarm	Tag of Low Alarm	Tag of High Alarm
$\%O_2$ in the Reactor Inlet	11	12
Gas Recycle Stream Pressure	21	22
HAc Tank Level	31	32
Vaporizer Level	41	42
Vaporizer Pressure	51	52
Heater Exit Temperature	61	62
Reactor Exit Temperature	71	72
Separator Level	81	82
Separator Temperature	91	92
Separator Vapor Flowrate	101	102
Compressor Exit Temperature	111	112
Absorber Level	121	122
Absorber Scrub Flowrate	131	132
Circulation Stream Temperature	141	142
Absorber Circulation Flowrate	151	152
Scrub Stream Temperature	161	162
$\%CO_2$ in the Gas Recycle	171	172
$\%C_2H_6$ in the Gas Recycle	181	182
FEHE Hot Exit Temperature	191	192
$\%H_2O$ in the Column Bottom	201	202
5^{th} tray Temperature	211	212
Decanter Temperature	221	222
Decanter Organic Level	231	232
Decanter Aqueous Level	241	242
Column Bottom Level	251	252
Liquid Recycle Flowrate	261	262
%VAc E-3	271	272

the natural sciences and process development). Gyula Dorgo was supported by the UNKP-14-2-I New National Excellence Program of the Ministry of Human Capacities

REFERENCES

[1] H. Pasman, "Chapter 8 - extended process control, operator situation awareness, alarm management," in *Risk Analysis and Control for Industrial Processes - Gas, Oil and Chemicals*, H. Pasman, Ed. Oxford: Butterworth-Heinemann, 2015, pp. 355 – 381. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128000571000080

[2] B. Mehta and Y. Reddy, "Chapter 21 - alarm management systems," in *Industrial Process Automation Systems*, B. Mehta and Y. Reddy, Eds. Oxford: Butterworth-Heinemann, 2015, pp. 569 – 582. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128009390000218

[3] E. Equipment, M. U. Association, E. Equipment, and M. U. A. Staff, *Alarm Systems: A Guide to Design, Management and Procurement*, ser. EEMUA publication. E E M U A (Engineering Equipment & Materials Users Association), 2015. [Online]. Available: https://books.google.hu/books?id=SrY3MwAACAAJ

[4] P. Grosdidier, P. Connor, B. Hollifield, and S. Kulkarni, "A path forward for dcs alarm management," *Hydrocarbon Processing*, vol. 82, no. 11, pp. 59 – 64, 2003.

[5] K. VanCamp, "Alarm management by the numbers," *Chemical Engineering Essentials for the CPI Professional, Automation & Control*, 2016.

[6] S. Breznitz, *Cry Wolf: The Psychology of False Alarms*. Lawrence Erlbaum Associates, 1984. [Online]. Available: https://books.google.hu/books?id=rfR-AAAAMAAJ

[7] J. Bliss, R. D. Gilson, and J. E. Deaton, "Human probability matching behaviour in response to alarms of varying reliability," vol. 38, pp. 2300–12, 12 1995.

[8] R. D. Sorkin and D. D. Woods, "Systems with human monitors: A signal detection analysis," *HumanComputer Interaction*, vol. 1, no. 1, pp. 49–75, 1985. [Online]. Available: http://dx.doi.org/10.1207/s15327051hci0101_2

[9] J. Meyer and Y. Bitan, "Why better operators receive worse warnings," *Human Factors*, vol. 44, no. 3, pp. 343–353, 2002, pMID: 12502153. [Online]. Available: https://doi.org/10.1518/0018720024497754

[10] M. Rayo and S. Moffatt-Bruce, "Alarm system management: Evidence-based guidance encouraging direct measurement of informativeness to improve alarm response," vol. 24, 03 2015.

[11] S. Bhaumik, J. MacGowan, and V. Doraj, "Mode based alarm solutions at syncrude canada," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 653 – 656, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405896315011234

[12] J. Wang and T. Chen, "Main causes of long-standing alarms and their removal by dynamic state-based alarm systems," *Journal of Loss Prevention in the Process Industries*, vol. 43, pp. 106 – 119, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950423016301279

[13] I. S. of Automation and A. N. S. Institute, *ANSI/ISA-18.2-2009, Management of Alarm Systems for the Process Industries*. ISA, 2009. [Online]. Available: https://books.google.hu/books?id=qyo4twAACAAJ

[14] N. A. Adnan, I. Izadi, and T. Chen, "On expected detection delays for alarm systems with deadbands and delay-timers," *Journal of Process Control*, vol. 21, no. 9, pp. 1318 – 1331, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959152411001296

[15] A. J. Hugo, "Estimation of alarm deadbands," *IFAC Proceedings Volumes*, vol. 42, no. 8, pp. 663 – 667, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016358530

[16] H. Zang, F. Yang, and D. Huang, "Design and analysis of improved alarm delay-timers," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 669 – 674, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S240589631501126X

[17] I. Izadi, S. L. Shah, D. S. Shook, S. R. Kondaveeti, and T. Chen, "A framework for optimal design of alarm systems," *IFAC Proceedings Volumes*, vol. 42, no. 8, pp. 651 – 656, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016358517

[18] I. Izadi, S. L. Shah, and T. Chen, "Effective resource utilization for alarm management," in *49th IEEE Conference on Decision and Control (CDC)*, Dec 2010, pp. 6803–6808.

[19] F. Yang, S. L. Shah, and D. Xiao, "Correlation analysis of alarm data and alarm limit design for industrial processes," in *Proceedings of the 2010 American Control Conference*, June 2010, pp. 5850–5855.

[20] M. Noda, F. Higuchi, T. Takai, and H. Nishitani, "Event correlation analysis for alarm system rationalization," *Asia-Pacific Journal of Chemical Engineering*, vol. 6, no. 3, pp. 497–502, 2011. [Online]. Available: http://dx.doi.org/10.1002/apj.575

[21] S. R. Kondaveeti, I. Izadi, S. L. Shah, and T. Black, "Graphical representation of industrial alarm data," *IFAC Proceedings Volumes*, vol. 43, no. 13, pp. 181 – 186, 2010, 11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667015325283

[22] S. R. Kondaveeti, I. Izadi, S. L. Shah, T. Black, and T. Chen, "Graphical tools for routine assessment of industrial alarm systems," *Computers & Chemical Engineering*, vol. 46, pp. 39 – 47, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135412002268

[23] Z. Yang, J. Wang, and T. Chen, "Detection of correlated alarms based on similarity coefficients of binary data," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 4, pp. 1014–1025, Oct 2013.

[24] W. Hu, J. Wang, and T. Chen, "A new method to detect and quantify correlated alarms with occurrence delays," *Computers & Chemical Engineering*, vol. 80, pp. 189 – 198, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135415002094

[25] J. Wang, H. Li, J. Huang, and C. Su, "A data similarity based analysis to consequential alarms of industrial processes," *Journal of Loss Prevention in the Process Industries*, vol. 35, pp. 29 – 34, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950423015000674

[26] D. Li, J. Hu, H. Wang, and W. Huang, "A distributed parallel alarm management strategy for alarm reduction in chemical plants," *Journal of Process Control*, vol. 34, pp. 117 – 125, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959152415001584

[27] V. B. Soares, J. C. Pinto, and M. B. de Souza, "Alarm management practices in natural gas processing plants," *Control Engineering Practice*, vol. 55, no. Supplement C, pp. 185 – 196, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066116301514

[28] S. Kordic, P. Lam, J. Xiao, and H. Li, "Associative data mining for alarm groupings in chemical processes," in *Proceedings of the 2007 International Conference on Intelligent Systems and Knowledge Engineering*, A. Press, Ed.

[29] ——, *Analysis of Alarm Sequences in a Chemical Plant*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 135–146. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88192-6_14

[30] J. Folmer and B. Vogel-Heuser, "Computing dependent industrial alarms for alarm flood reduction," in *International Multi-Conference on Systems, Sygnals Devices*, March 2012, pp. 1–6.

[31] Y. Cheng, I. Izadi, and T. Chen, "Pattern matching of alarm flood sequences by a modified smithwaterman algorithm," *Chemical Engineering Research and Design*, vol. 91, no. 6, pp. 1085 – 1094, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263876212004261

[32] S. Lai and T. Chen, "A method for pattern mining in multiple alarm flood sequences," *Chemical Engineering Research and Design*, vol. 117, pp. 831 – 839, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0263876215002257

[33] K. Ahmed, I. Izadi, T. Chen, D. Joe, and T. Burton, "Similarity analysis of industrial alarm flood data," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 452–457, April 2013.

[34] W. Hu, S. L. Shah, and T. Chen, "Framework for a smart data analytics platform towards process monitoring and alarm management," *Computers & Chemical Engineering*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0098135417303551

[35] S. Lai, F. Yang, and T. Chen, "Online pattern matching and prediction of incoming alarm floods," *Journal of Process Control*, vol. 56, no. Supplement C, pp. 69 – 78, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0959152417300100

[36] R. Karoly and J. Abonyi, "Multi-temporal sequential pattern mining based improvement of alarm management systems," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2016, pp. 003 870–003 875.

[37] J. Goh, S. Adepu, K. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," 10 2016.

[38] R. Chen, K. Dave, T. J. McAvoy, and M. Luyben, "A nonlinear dynamic model of a vinyl acetate process," *Industrial & Engineering Chemistry Research*, vol. 42, no. 20, pp. 4478–4487, 2003. [Online]. Available: http://dx.doi.org/10.1021/ie020859k

Gyula Dorgo graduated with bachelor's (2016) and master's (2017) degree in chemical engineering with a specialization in process engineering from the University of Pannonia. His research interest covers the areas of goal-oriented optimization, event analysis and data science applications in process mining. His current research focuses on the applications of process mining tools in alarm management for the improvement of process safety.

Janos Abonyi is a full professor at the Department of Process Engineering at the University of Pannonia in computer science and chemical engineering. He received MEng and PhD degrees in chemical engineering in 1997 and 2000 from the University of Veszprem, Hungary. In 2008, he earned his Habilitation in the field of Process Engineering, and the DSc degree from the Hungarian Academy of Sciences in 2011. In the period of 1999-2000 he was employed at the Control Laboratory of the Delft University of Technology (in the Netherlands). Prof. Abonyi has co-authored more than 240 journal papers and chapters in books and has published five research monographs and one Hungarian textbook about data mining. His research interests include complexity, process engineering, quality engineering, data mining and business process redesign.