Journal of Loss Prevention in the Process Industries 62 (2019) 103959

Contents lists available at ScienceDirect

Journal of Loss Prevention in the Process Industries

journal homepage: www.elsevier.com/locate/jlp

A data-driven alarm and event management framework

Pankaj Goel[a,b,c], E.N. Pistikopoulos[c,*], M.S. Mannan[b,1], Aniruddha Datta[a,**]

[a] Department of Electrical and Computer Engineering, United States
[b] Mary Kay O'Connor Process Safety Center, Artie Mc-Ferrin Department of Chemical Engineering, United States
[c] Texas A&M Energy Institute, Texas A&M University, College Station, TX, 77843, United States

ARTICLE INFO

Keywords:
Alarm management
Data mining and analysis
Abnormal situation management (ASM)
Visualization

ABSTRACT

Industrial systems are monitored and controlled by sensors and actuators. The signals from these instruments are mostly configured as alarms in the control system which alerts the plant operators in case of an abnormal process event. A higher number of alarms appearing on the operator screen signifies poor system performance and results in additional workload on the operators and may lead to abnormal situations. These situations can further escalate to a catastrophic incident if not managed properly. There are several solutions available to address the challenges related to the alarms and their management as a part of alarm management life-cycle process defined in industrial standards and guidelines. With the advent of Open Process Automation (OPA) concept and requirement of Real-time Operational Technology (OT) services there is a need to develop solutions based on open source software platforms. To address this, an alarm management framework is proposed, that integrates the alarm management life-cycle concept provided in ANSI/ISA-18.2 with data mining and analysis methods applied on alarm and event logs generated from a control system. The framework involves four distinct levels - design, rationalize, advance, and intelligent. A methodology to benchmark an alarm system by calculating Key Performance Indicators (KPIs) is formulated, results are obtained and shown visually by designing a tool using an open source software platform (R and Python). With the use of data mining and analysis, the methodology/tool benchmarks the system for the defined metrics for alarm systems as KPIs, identifies the bad actors, provides insights about the alarm system to manage the alarm flooding. We use the results to address the key issues and thereby improve the overall efficiency of the alarm system. A historical real industrial Alarm and Event (A&E) log data-set is used as an example to demonstrate the potential application of the proposed alarm management tool, to arrive at reliable decision, contributing to better alarm management and safer operations.

1. Introduction

Alarm systems play an imperative role in operating a process in the safer region and serve as a layer of protection in preventing the escalation of a process upset to an abnormal situation. *"An alarm is an audible and/or visible means of indicating to the operator an equipment malfunction, process deviation, or abnormal condition requiring a timely response"* (ANSI/ISA-18.2, 2016). In a recently published study (Jain et al., 2018a), 76% of the respondents mentioned alarm rate to be an important factor in determining or predicting the process upset events. Alarm systems act as an early detection indicator that alerts the operator about the abnormal process condition (Jain et al., 2017; Goel et al., 2017b; Pariyani et al., 2012) or malfunction of equipment in service. Designing an alarm system includes understanding the process, developing a master alarm database, defining operator actions and implementing in process control systems. The process of designing, implementing, understanding and operating systems of alarms is known as alarm management. Industrial process systems are monitored and controlled by sensors and actuators. Due to the ease in configuration techniques available at the software level, most of these sensors are configured as alarms in the control system. This has resulted in a higher number of alarms, poor system performance, additional workload on operators, and in some cases has led to abnormal situations. These situations can further escalate to catastrophic incidents, if not managed properly (Jain et al., 2018b).

Alarm flooding is one of the causes due to which critical process alarms are being overlooked or judgmental errors being made by the operator. This results in losses for the operating company which may

* Corresponding author.
** Corresponding author.
E-mail addresses: stratos@tamu.edu (E.N. Pistikopoulos), datta@ece.tamu.edu (A. Datta).
[1] Deceased 11 September 2018.

https://doi.org/10.1016/j.jlp.2019.103959
Received 28 April 2019; Received in revised form 10 July 2019; Accepted 12 September 2019
Available online 21 September 2019
0950-4230/ © 2019 Elsevier Ltd. All rights reserved.

include both direct losses such as production loss, equipment damage or indirect losses such as reputation, fines,*etc.* which are sometimes humongous. Hence, there is a critical need to design and develop techniques to manage the alarm flooding challenges using the process information and alarm and event data. Both academia and industry have made efforts in addressing the issues of reducing alarm flooding and operator work-load, enhance alarm design and effective alarm management strategies. The literature survey shows there are several methods derived in the past related to developing tools for operator assistance (Carrera and Easter, 1991), methods for alarm shelving (Burnell and Dicken, 1997), co-relation analysis (Higuchi et al., 2009), finding redundant alarms (Folmer and Vogel-Heuser, 2012), alarm flood classification (Lucke et al., 2018; Charbonnier et al., 2016; Lai et al., 2017; Dorgo et al., 2018), similarity analysis (Yang et al., 2012, 2013; Wang et al., 2015; Ahmed et al., 2013), visualization (Kondaveeti et al., 2012), pre-processing of alarm data (Mannani et al., 2019), pattern mining and detection of alarm flood sequence (Niyazmand and Izadi, 2019; Hu et al., 2018a), integrating process data into alarm analysis with the help of a tool developed in matlab (Hu et al., 2018b). A detailed list can be found in (Goel et al., 2017b; Lucke et al., 2019). From the discussion above, we can conclude that there are very few integrated tools/solutions available to address the issues related to alarm management based on analyzing the alarm and event logs generated from the control system.

Also, with the advent of Open Process Automation (OPA) concept and requirement of Real-time Operational Technology (OT) services, there is a need to develop solutions based on open source software platforms. To address these, an alarm management framework is proposed, that integrates the alarm management life-cycle concept provided in ANSI/ISA-18.2 with data mining and analysis methods applied on alarm and event logs generated from a control system. The organization of the paper is as follows: Section 2 introduces the proposed framework involving four distinct levels - design, rationalize, advance, and intelligent. Users can categorize their current efforts into one of the stages and work forward towards improving the alarm systems. Section 3 describes the application of data mining and analytics (an offline data mining and analysis method) to address the issues related to alarm flooding and management in industrial facilities. The method benchmarks the system for the Key Performance Indicators (KPIs) defined in (ANSI/ISA-18.2, 2016) & (EEMUA-191, 2013), identifies the bad actors, provides insights about the alarm system to manage the alarm flooding, and thereby improving the overall efficiency of the system. Section 4 demonstrates the effectiveness of the proposed method using historical real industrial Alarm and Event (A&E) log data-set followed by the conclusions and future work in Section 5. The main contributions of this work include:

1. A novel framework to categorize the process of the alarm management system in an industrial facility (design, rationalize, advance and intelligent). This framework can help end users categorize their alarm management program. For a facility, it is important to start the alarm management program with designing the alarm systems, rationalizing and then using the proposed methodology to remove bad actors and reduce alarm flooding. The framework follows a life-cycle approach, which includes bench-marking the alarm system and following the re-design and re-rationalize steps if required.
2. An integrated method to calculate KPIs and generate visualization plots. This provides an overall better approach to analyze the Alarm and Event logs and disseminate information to the user to take appropriate corrective actions to improve the overall alarm management program.
3. A data-mining and analysis based alarm analysis tool on open source software platform to address the issues related to alarm management. We use data set from a real industrial plant to demonstrate the features of the tool.
4. The proposed method provides an opportunity for the users to design and develop tools in an open source software platforms (R Core Team, 2018; Rossum, 1995) and make them easily more useable and scalable based on the various processes and business requirements. Some of the key features of the proposed tool are: web-based application tool, manual data file upload or automatic data file update, interactive visualization and a friendly user-interface, minimum design cost.

2. Alarm management framework

International Society of Automation (ISA) defines an alarm system as:*"the collection of hardware and software that detects an alarm state, communicates the indication of that state to the operators, and records changes in the alarm state"* (ANSI/ISA-18.2, 2016). Alarm management is defined as *"set of processes that ensures an effective alarm system"*. A typical industrial facility is operated by a control system (DCS or PLC or both), a Human- Machine Interface (HMI) is used to display the information related to process variables, operation and alarms. In some special cases such as in safety systems or a critical equipment the alarms and some critical controls are duplicated on an annunciator panel. Alarm historians are used to record and store the information comprising of alarms messages, process changes, and operator actions for future records and references. An effective alarm management program can help the users in operating the plant at both optimum and safer levels resulting in lower losses, increased throughput and higher quality.

One of the biggest challenges in alarm management for industrial facilities is the alarm floods.*"Alarm flooding is a condition during which the alarm rate is greater than what the operator can effectively manage (e.g. more than 10 alarms* per *10 min)"* (ANSI/ISA-18.2, 2016; EEMUA-191, 2013). Alarm floods result in abnormal situations and in some cases the situation may escalate to catastrophic incidents. Fig. 1 illustrates few major incidents that occurred in the past three decades where alarms or alarm system management were reported as one of the root causes. In past few years, there have been several standards and guidelines developed comprising of instructions for the users to design and manage the alarm systems. Additionally, these documents provide detailed requirements related to certain KPIs for bench-marking the alarm systems. Some guidelines related to technical methodologies and steps are provided as a part of separate technical reports. Usually, the alarm management process in an industrial facility includes designing and rationalizing the systems. Advanced alarming techniques are used in only few cases. The use of data mining and analysis systems to harness the information from historians to improve alarm management system is very scarce.

To address the various issues of alarm management, the authors propose an integrated alarm management framework based on ANSI/ISA 18.2 alarm management life-cycle and data mining and analysis methods as shown in Fig. 2, which consists of four stages: Design, Rationalize, Advance and Intelligent. It is important to note the different actions and stages are required to be performed and implemented sequentially. The life-cycle is not a one time process and requires continual improvement and implementation. The four stages of the proposed alarm management framework are described below:

- **Design:** The design stage is the most critical part of an effective alarm management program. During the design stage, users design their alarm systems based on available standards, guidelines and best practices used in the industry, their own organization. It includes developing the philosophy document and the design requirements. A preliminary master alarm database is generated which includes the potential list of alarms for a facility. While designing alarm systems the important characteristics to be built into the design are (Goel et al., 2017b):
  1. Every designed alarm should require an operator action.
  2. Priority - priority selection for the alarms according to the rule of

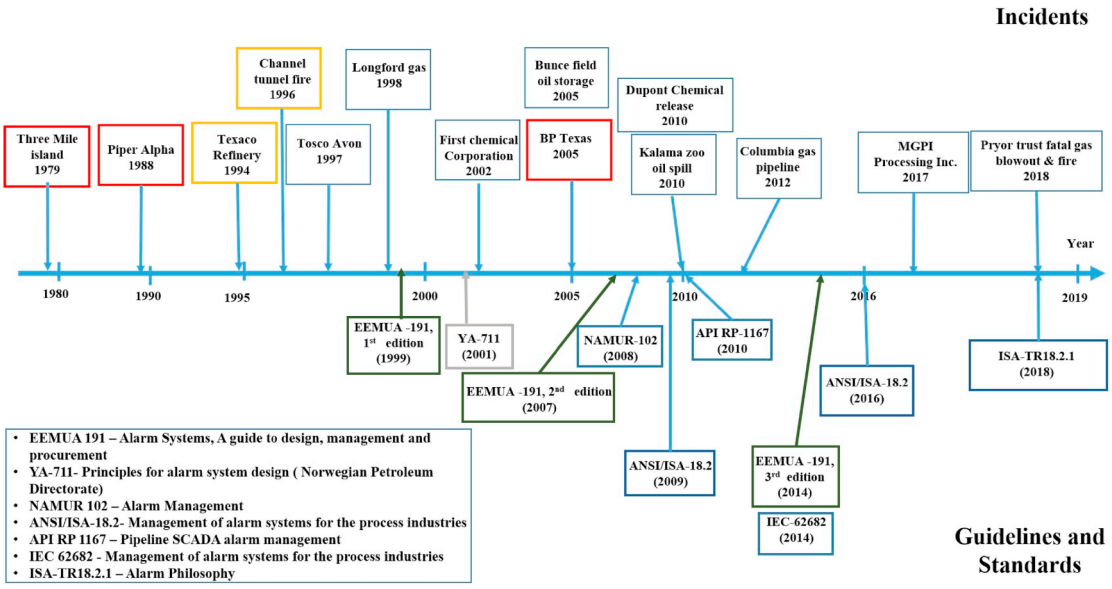


Fig. 1. Evolution of alarm management (modified and adopted from (Goel et al., 2017b)).

85/15/5 for (Low/Medium/High) priorities.

3. Uniqueness - alarm indicates the details about a single unique process parameter with a suitable description to understand the alarm.
4. Timeliness - alarm appears on time and provide appropriate time for operator to detect, diagnose and act.
5. Relevance - alarm relevance for the operational value.

- **Rationalize:** The rationalize stage includes tasks of alarm classification, prioritization, rationalization and documentation. The rationalization process requires inputs such as alarm philosophy document and list of all potential alarms (initial master alarm database) which is generated during the design phase. The rationalization step provides the result in the form of master alarm database with design requirements. There may be a significant reduction in the configured alarms and the nuisance alarms after rationalization. The rationalization process can be implemented both during the design of a new system or to an existing running system to improve the installed alarm system.
- **Advance:** This stage includes advanced alarming techniques to manage the alarm floods in case of process changeover conditions such as start-up and shut-down. In such situation, suppression techniques are used (Rothenberg, 2009), which work on the principle of hiding or masking the alarms which are not relevant to the operator after a particular process event. These techniques include: alarm shelving (manual suppression) initiated by the operator, designed suppression (automatic suppression) based on operating states or plant conditions (also known as static suppression) such as a reactor start-up and out of service alarms (also known as dynamic suppression) such as a compressor trip or an out of service equipment.
- **Intelligent(based on data analytics):** With advancement in digitization more data is being collected and stored daily by the operating companies. The energy industry is becoming 'data rich'. Data analytics is the key enabler to find out insights from the raw data for more informed business and operational decisions. Enormous amounts of data is generated by sensors and actuators for process operations, automatic or manual actions and safety that is stored in 'data-warehouses' or 'data lakes'. Data mining and analysis methods can be used to reveal the relationships between the data sets. The information generated from such methods can be shown to the user with the help of visualization tools. This provides an opportunity to derive business intelligence from the data and improve overall system performance (Goel et al., 2017a). Fig. 3 highlights key phases of analysis to support improved operations, process safety and risk management. The process starts with collecting data and performing descriptive analysis to understand what is going on; a diagnostic analysis to understand why it is happening. In certain cases we can use a predictive approach to predict the future and the last step is prescriptive analysis involving real-time analysis and reporting. The value contribution and complexity of the solution increases at each step. The use of advanced analytics and expert knowledge is required in these cases to derive informed decisions and business intelligence. To perform analytics on a problem we need to follow a life-cycle approach as shown in Fig. 4. The process starts with identifying the purpose of the study and questions which require answer or analysis. After, this step the data is collected from various sources and aggregated to ensure the availability of required data and information for the study. The next step includes developing the methodology and perform analysis to find meaningful information and results. The obtained results are interpreted by experts and then disseminated to the end users for final evaluation. After evaluation if there are any changes required in the approach,

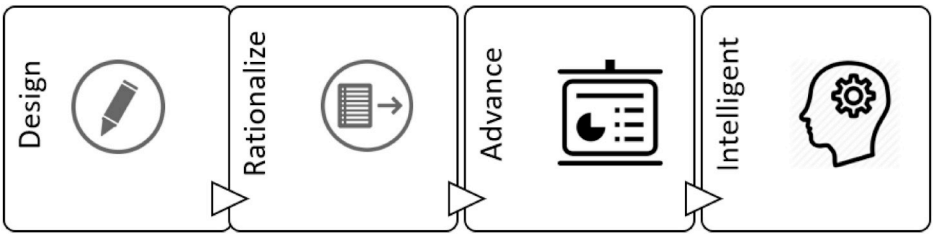


Fig. 2. Alarm management framework.

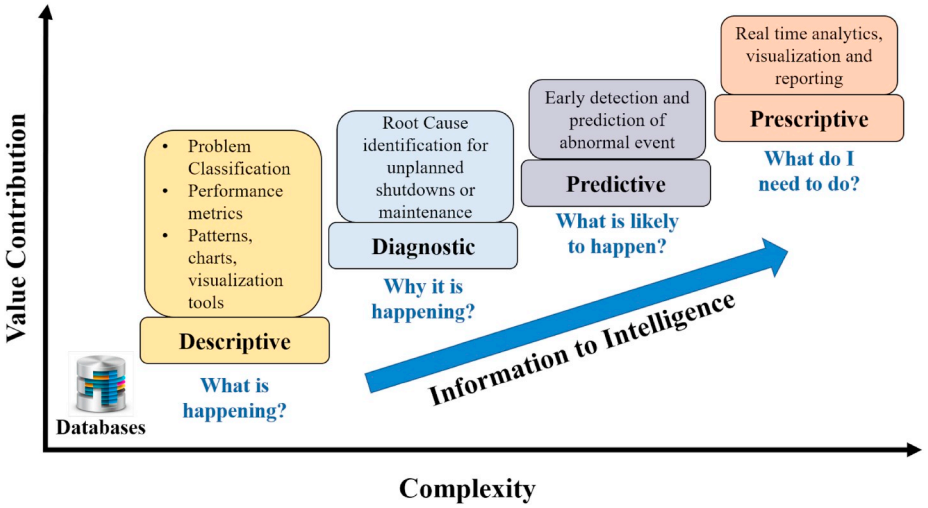


Fig. 3. Data to business intelligence.

the appropriate information is shared to update the purpose or evaluation. For the purpose of this study the data analysis life-cycle stages are defined as: (1) Purpose definition: address the issue related to alarm management, develop a methodology that can be used for analyzing alarm and event log; (2) Data Collection: For this study the data is collected from a DCS historian of a real industrial plant in appropriate template or format; (3) Data analysis: various methods developed for this stage are shown in Section 3; (4) Interpretation, dissemination and evaluation of results which are described in Section 4. In case of alarm management, the use of analytic tools can provide an opportunity to find out the bad-actors, improve the overall alarm management system and operate the plant systems at an optimized level. In this paper, a method is developed for offline analysis for alarm and event logs and explained in next sections. The information generated from this stage can be used in other stages to enhance the alarm management program.

3. Problem formulation

An alarm serves as a medium to communicate the abnormal process event to the operator. A sensor is used to measure the process value and the output is wired as an analog or digital signal to the control system (DCS, PLC or ESD system). The transmitted signal is processed by the programs written and stored in electronic circuits, logic boards (also known as controllers) serving as the automated brain of the system. Post-processing the generated information is displayed to the operator on HMI displays in the form of process operations screens and alarm display screen. The operator uses the information to assess the process operation and takes necessary actions when needed. During the alarm activation state (equation (1)), the operator reviews the state of process variables and other related plant information, comprehends the current plant state, detect and diagnose the abnormal process situation and carry out the required actions to bring the process back to normal state. The operator's action activates a sequence of automated operation including a desired control action from the controller to the final control element (also known as actuator) which brings back the process to a desired operating range (Mehta and Reddy, 2014). These events are captured in Alarm & Event log of a control system (DCS/PLC) historian as:

- ALM: alarm appeared,
- ACK: alarm acknowledged by an operator,
- RTN: process variable returned to a normal state.

Alarm and Event log data captured can provide very relevant and useful information regarding bad actors (*"an alarm that is suspect and cannot be relied upon to deliver accurate information to the operator, such as stale, chattering, duplicate or suppressed alarms"* (Grosdidier et al., 2003)), flood sequences which will result in better alarm management performance. To demonstrate this, we have developed a methodology that uses an Alarm and Event log information and the previous knowledge of the process operation. This methodology includes following steps:

- **Step 1: Acquire and validate the data-set from alarm historian:** Alarm and Event log is stored in a historian of a process control system. This step includes acquiring the data from the historian in a relevant file format. The generated file is used in the next step.
- **Step 2: Pre-process the data-set to ensure correctness:** The alarm data-set not only includes the information of the process alarms, in some cases they include other information such as system failure alarms, hardware failure alarms too. It is important to filter these alarms from the process alarms. Also, time stamping needs to be checked to ensure the date and time stamp for each alarm is captured in the system. This process is shown in Algorithm 1. The output of this level is an Alarm and Event log which is used in subsequent steps.
- **Step 3: Analyze and evaluate alarm flood clusters, patterns and KPI information:** The next step is to find the alarm floods from the Alarm and Event log generated in pre-processing step. To ensure the correct values are collected and used again the data-set is checked for quality and flood sequences are identified and KPIs for the alarm system are calculated as mentioned in ISA 18.2 standard and EEMUA 191 guideline.
- **Step 4: Provide information to the users with visualization tools:** This information is necessary to understand the system performance. This step includes selection of appropriate methods and designing the visualization screens post analysis of the complete alarm and event logs.

Fig. 5 highlights the complete process presented in this paper. For the purpose of the study, an alarm event (a_i) is defined as a binary-valued variable such that

$$a_i(t) = \begin{cases} 0, & \text{if } O(t) \in O_p, \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

Which means that alarm is inactive (0) whenever the value of the operating condition $O(t)$ is in normal operating condition O_p and active (1) in case of a deviation. An alarm can be configured as Low/Low-Low

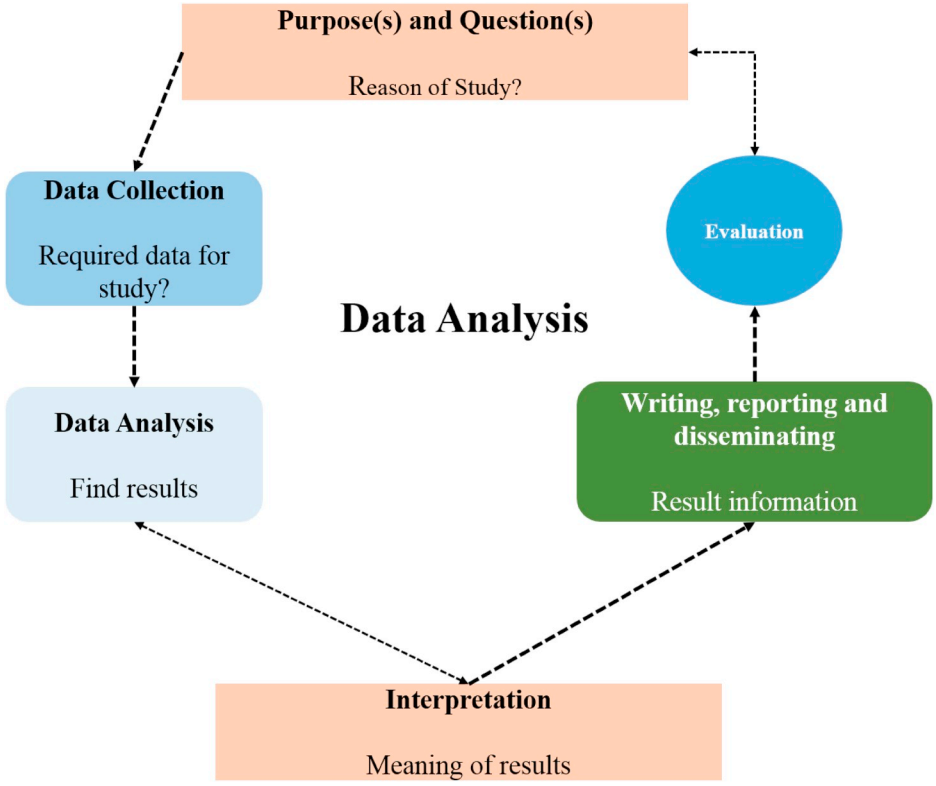


Fig. 4. Data analysis as a life-cycle.

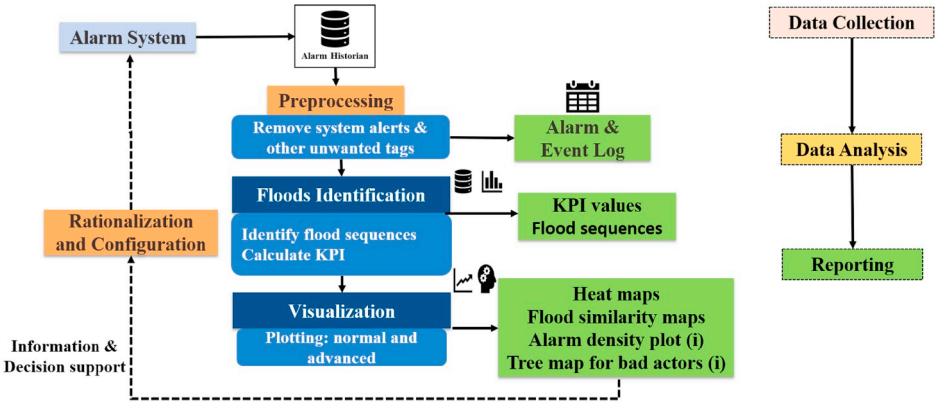


Fig. 5. Methodology framework.

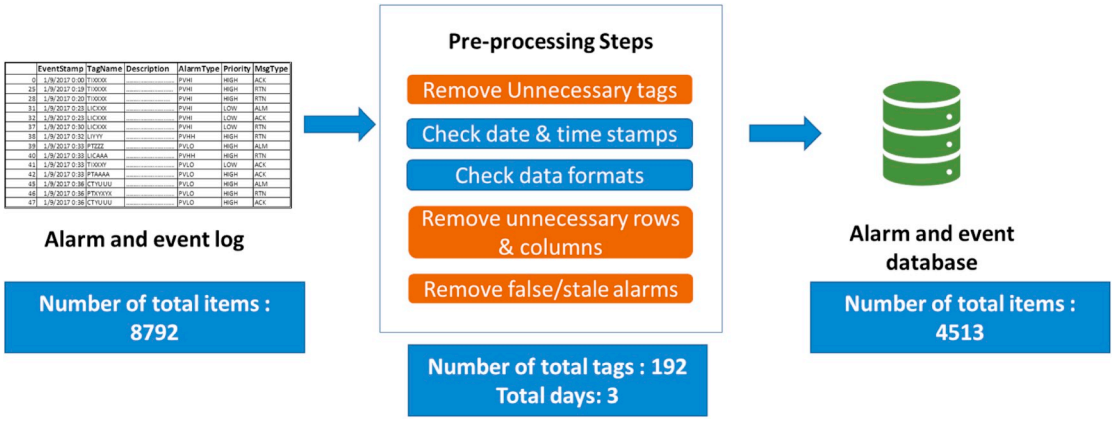


Fig. 6. Alarm and Event log Pre-processing.

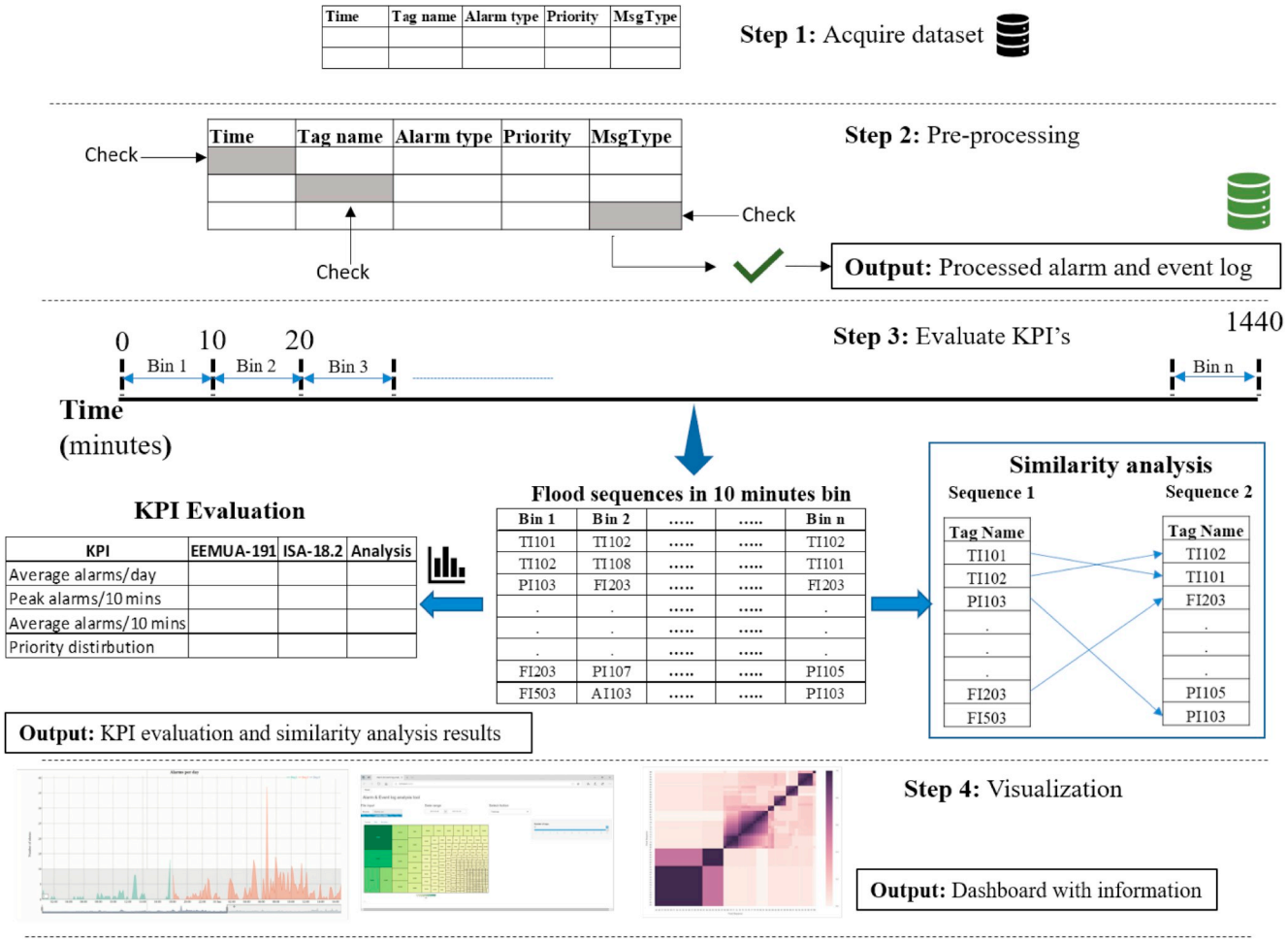


Fig. 7. Illustrative motivating example.

	EventStamp	TagName	Description	AlarmType	Priority	MsgType
0	1/9/2017 0:00	TIXXXX	.....	PVHI	HIGH	ACK
25	1/9/2017 0:19	TIXXXX	.....	PVHI	HIGH	RTN
28	1/9/2017 0:20	TIXXXX	.....	PVHI	HIGH	RTN
31	1/9/2017 0:23	LICXXX	.....	PVHI	LOW	ALM
32	1/9/2017 0:23	LICXXX	.....	PVHI	LOW	ACK
37	1/9/2017 0:30	LICXXX	.....	PVHI	LOW	RTN
38	1/9/2017 0:32	LIYYY	.....	PVHH	HIGH	RTN
39	1/9/2017 0:33	PTZZZ	.....	PVLO	HIGH	ALM
40	1/9/2017 0:33	LICAAA	.....	PVHH	HIGH	RTN
41	1/9/2017 0:33	TIXXXY	.....	PVLO	LOW	ACK
42	1/9/2017 0:33	PTAAAA	.....	PVLO	HIGH	ACK
45	1/9/2017 0:36	CTYUUU	.....	PVLO	HIGH	ALM
46	1/9/2017 0:36	PTXYXYX	.....	PVLO	HIGH	RTN
47	1/9/2017 0:36	CTYUUU	.....	PVLO	HIGH	ACK

Fig. 8. Sample data-set attributes.

alarm (where the value of the measured variable is below the operating condition limit) or High/High-High alarm (where the value of the measured variable is higher than the operating condition limit). A flood event, in this case, is defined as:

$$A = [a_1, a_2, \ldots, a_n], \tag{2}$$

where symbol [.] indicates a sequence, |A| is cardinality of the sequence, number of alarms is denoted by (n), and a_i is the alarm event occurring in a chronological order with a time stamp (t). a_i can be represented by a tuple with multiple attributes

$$a_i = (e_i, t_i, at_i, p_i, m_i) \tag{3}$$

where

- (e_i) is alarm tag (such as FT, PT, TT, *etc.*),
- (t_i) is the time stamp of the alarm occurred (HH:MM:SS or 11:00:25 format),

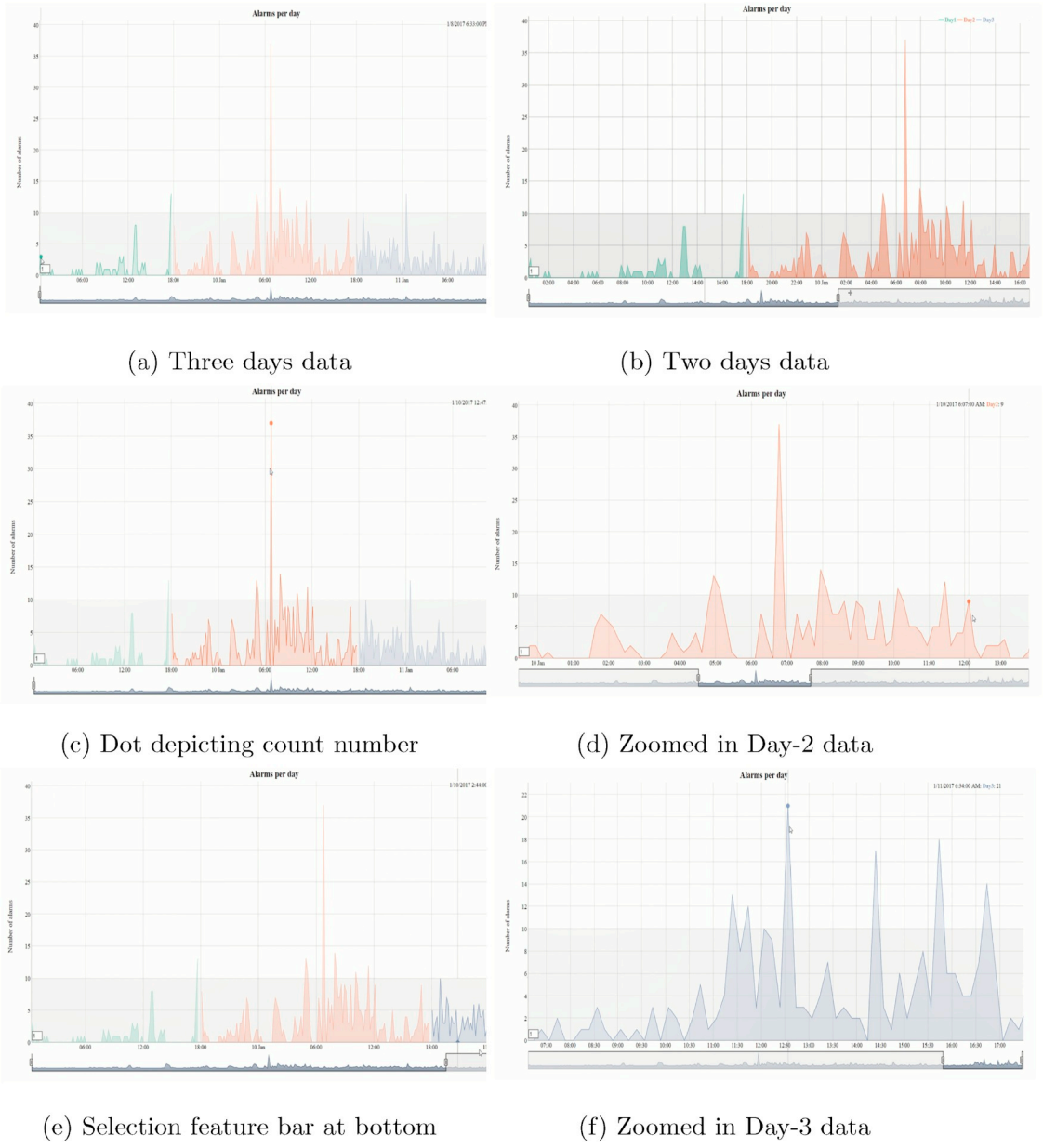


Fig. 9. Interactive alarm/day plots showing alarm information for three days (day1 - green, day2 - orange, day3 - blue) with date time and alarm values on top right corner of each plot, scroll option for zoom selection at bottom, and a gray colored line highlighting number of alarms=10 depicting alarm floods in case the value exceeds. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

- (at_i) is alarm type (Low, Low-low, High, HIgh-High *etc.*),
- (p_i) is priority setting for each alarm (Low, Medium, High),
- (m_i) is the message type generated (ALM, ACK, RTN).

To detect the alarm floods in the database alarm rate is calculated. Alarm rate is defined as number of alarms during a time period. The alarm rate $R(t)$ at a time t is defined as:

$$R(t) = \sum_{n=1}^{|A|} \sum_{k=\Delta T+1}^{t} a_i(k) \tag{4}$$

here, ΔT is the difference between time stamp of an alarm tag in seconds and 600 s (10 min interval).

By using this alarm rate, the identification of alarm floods can be done by comparing the rate with a pre-defined threshold.

An indexing variable (ζ) can be used to show the presence of an alarm flood as:

$$\zeta(t) = \begin{cases} 1, & \text{if } R(t) \geq \tau_s \text{ and } \zeta(t-1) = 0 \\ 0, & \text{if } R(t) < \tau_e \text{ and } \zeta(t-1) = 1 \end{cases} \tag{5}$$

here, 0: No alarm flood condition

1: Alarm flood condition

$R(t)$: is the rate calculated from equation 4

τ_s: alarm count threshold (10 alarms over 10 min).

τ_e: alarm count threshold (five alarms over 10 min) with threshold conditions as defined in (ANSI/ISA-18.2, 2016).

Algorithm 1. Pre-processing algorithm

Input: Alarm & Event dataset from an industrial process
Output: Processed Alarm & Event log (AE)
Read ← file with A = [a_1, a_2, ...,a_n]
Check - a_n for (e_n, t_n, at_i, p_i, m_i)
Remove - a_n with missing values

Remove - a_n if $a_n \in$ SYS OR Non-process tag
Return → Processed Alarm & Event log (AE)

When an Alarm and Event log is captured from a control system, it includes alarms other than process variables (Non-process tags), such as system alarms, hardware alarms, *etc.* It is required to remove such alarms to obtain the correct picture of the process system. Hence, these alarms are removed as a part of the pre-processing of the database. The date and time stamps for each alarm and event are checked. While performing the analysis, it is important to check and validate correct data format too. The available data is checked for data format accuracy after converting it into the data-frames. Any unnecessary rows and columns which don't contain complete information or with bad values are removed after consultation with an expert. The overall pre-processing method is shown in Fig. 6. Once the pre-processing is complete an alarm log with only process variables is available for analysis. This log includes both normal sequence and the flood sequence of alarms. With the help of data mining and analysis methods, the KPIs are calculated.

The average alarm rate per day is calculated by finding the total number of alarms appearing per day

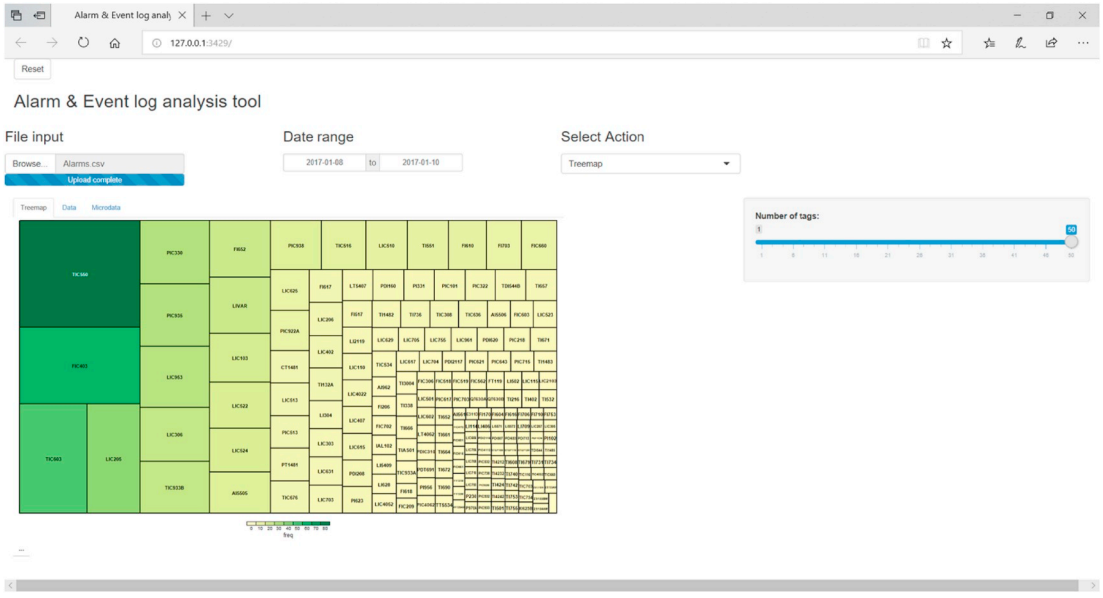
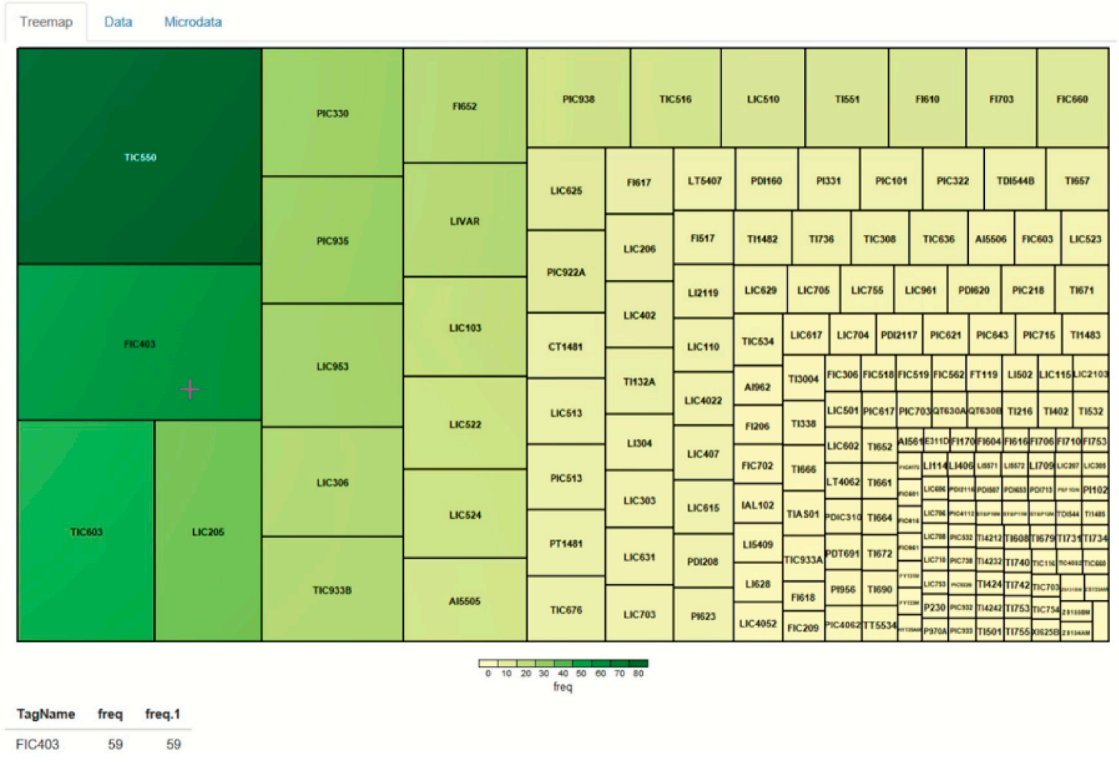


Fig. 10. Tool screen shot.



TagName	freq	freq.1
FIC403	59	59

Fig. 11. Tree map showing tag information.

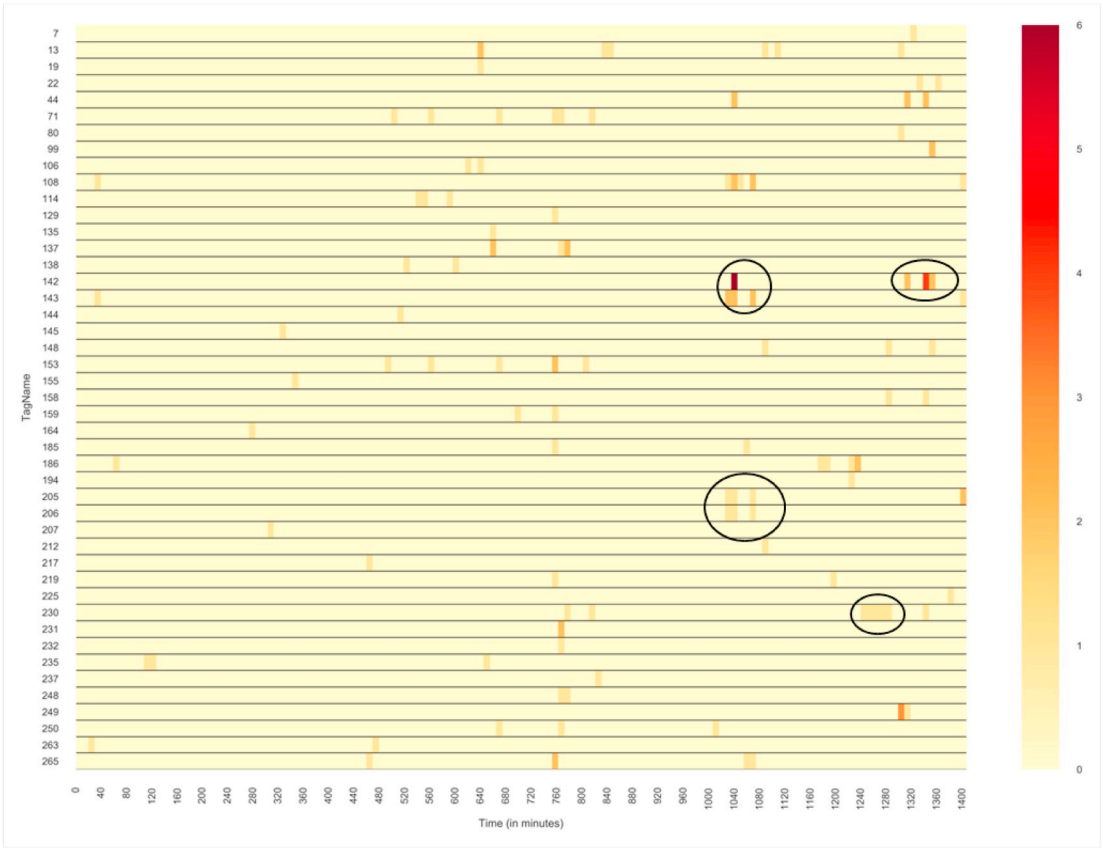


Fig. 12. Day 1 tag information.

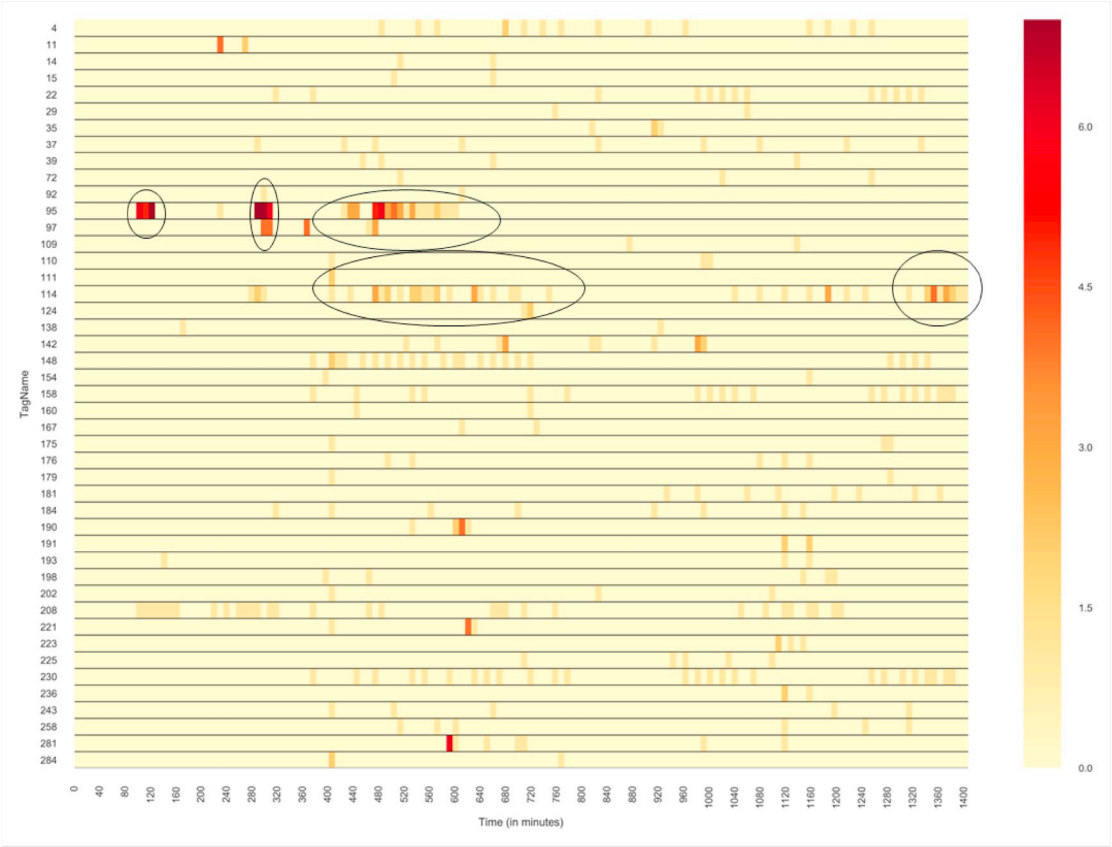


Fig. 13. Day 2 tag information.

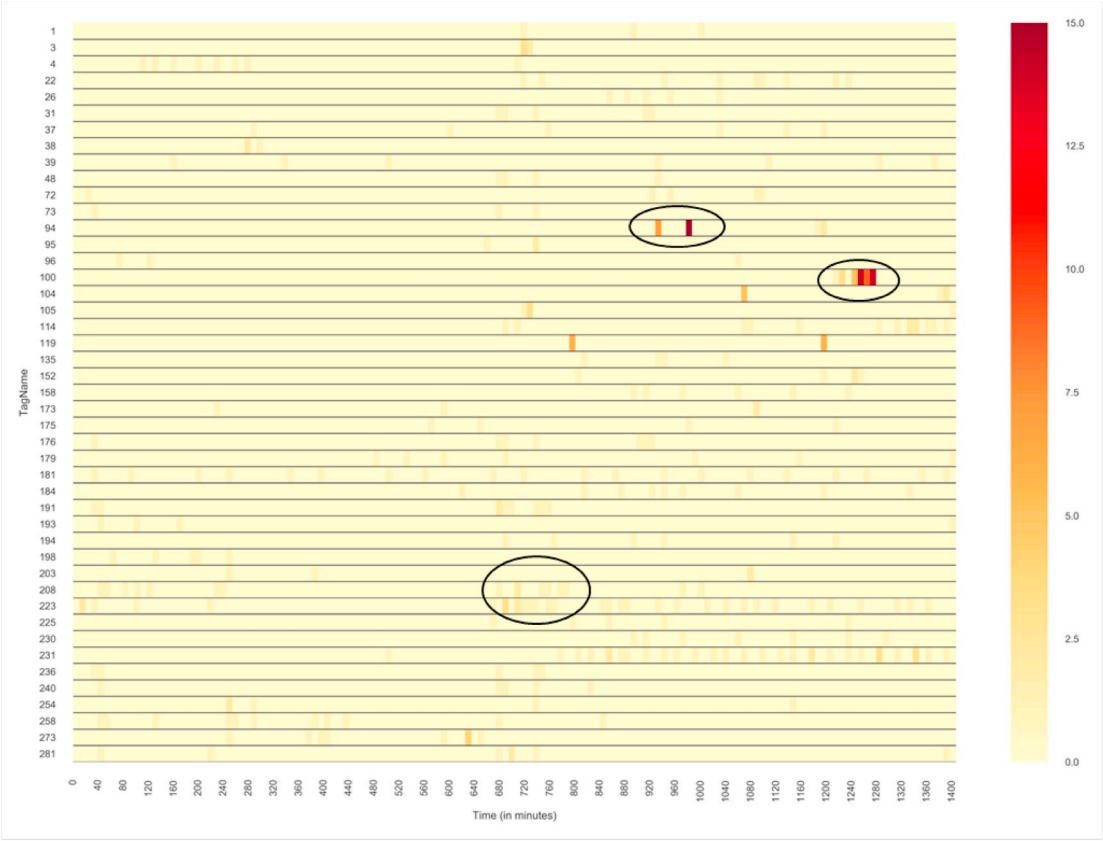


Fig. 14. Day 3 tag information.

Table 1
Data-set details.

Details	Values
Total items (alarms + events) before pre-processing	8793
Total items after pre-processing	4513
Total number of process tags	192
Total number of days	3

$$R(t)_{avg.} = \sum_{n=1}^{|A|} \sum_{t=0}^{1440} a_i(k) \,(\text{when } a_i(k) = \text{ALM}) \tag{6}$$

The peak alarm rate is calculated as the largest value of alarm rate $R(t)$ over a period of 10-min interval

$$R(t)_{peak} = Max. \left(\sum_{n=1}^{|A|} \sum_{t=0}^{10} a_i(k) \right) (\text{when } a_i(k) = \text{ALM}) \tag{7}$$

The priority distribution for the data-set is calculated by counting the number of alarms appearing for a particular priority

$$A_{dis.} = \sum_{n=1}^{|A|} p_i(k) \,(\text{when } p_i(k) = \text{Low, Medium, High}) \tag{8}$$

Algorithm 2. Flood sequences and plotting algorithm

Input: Processed alarm & Event log (AE)
Output: Clustered alarm flood sequences
Read ← file (AE),
Check - a_n for (e_n and t_n)
Cluster C_k: k = 1,2,3, ...,K such that C_k = (A_i: $i \in$ 1,2,..., N)
for each (C_k) find similarity
Return → Clustered alarm sequences

The flood sequences generated from Algorithm 3 are used to find out the similarity between the alarm floods for a pair of alarm flood sequences. The normalized similarity index (normalized between 0 and 1) for the alarm sequences is calculated based on the Jaccard index as given in (Fullen et al., 2017), where given two alarm sequences X and Y, the alarm similarity index is calculated by:

$$SI(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{9}$$

The normalized similarity index is plotted in the form of heat maps, where darkest color depicts the highest similarity between two alarm sequences.

A motivating illustrative example depicting the proposed method

Table 2
KPI analysis for the alarm and event log.

KPI	EEMUA-191	ISA 18.2	Analysis result
Average alarms/day	<144	~155	Day1:142 Day2:504 Day3:526
Peak alarms/10 min	<10	≤10	Day1:11 Day2:39 Day3:41
Average alarms/10 min	1	~1	Day1:1 Day2:2.1 Day3:2.2
Priority distribution (L/M/H)	80/15/5	80/15/5	60/30/5/5(E)

Fig. 15. Similarity analysis for alarm flood clusters (part I).

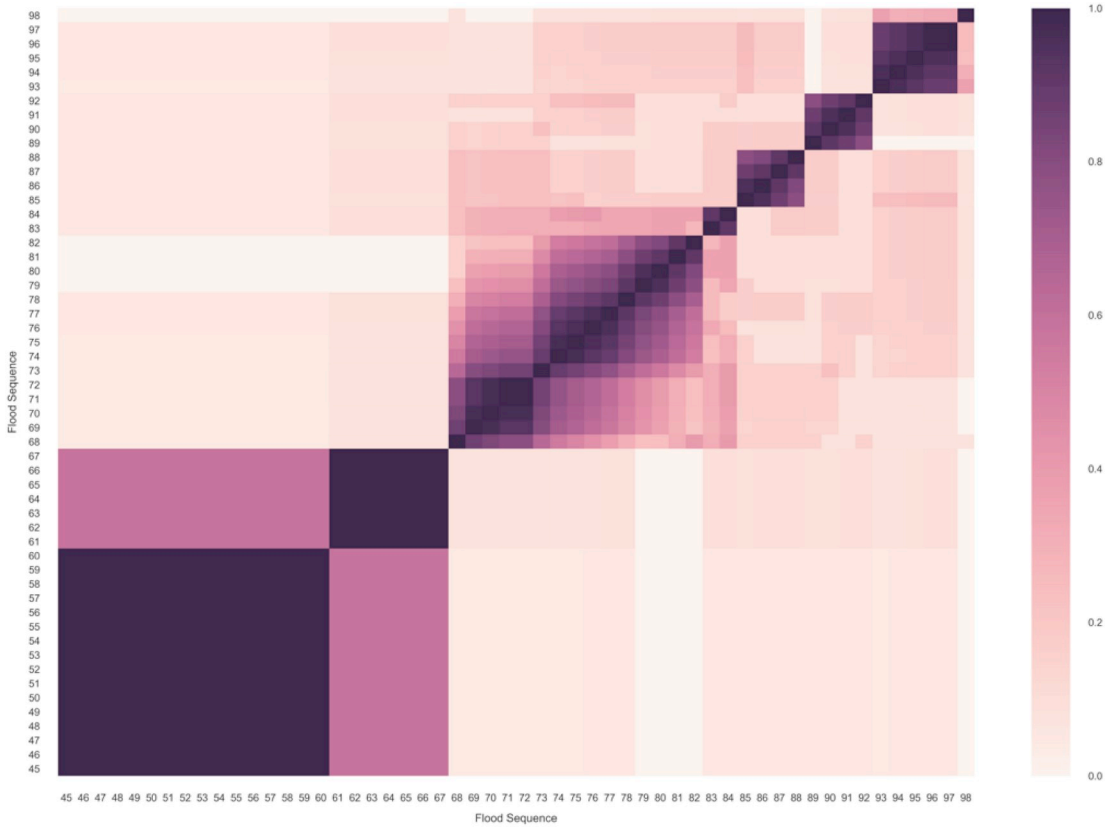


Fig. 16. Similarity analysis for alarm flood clusters (part II).

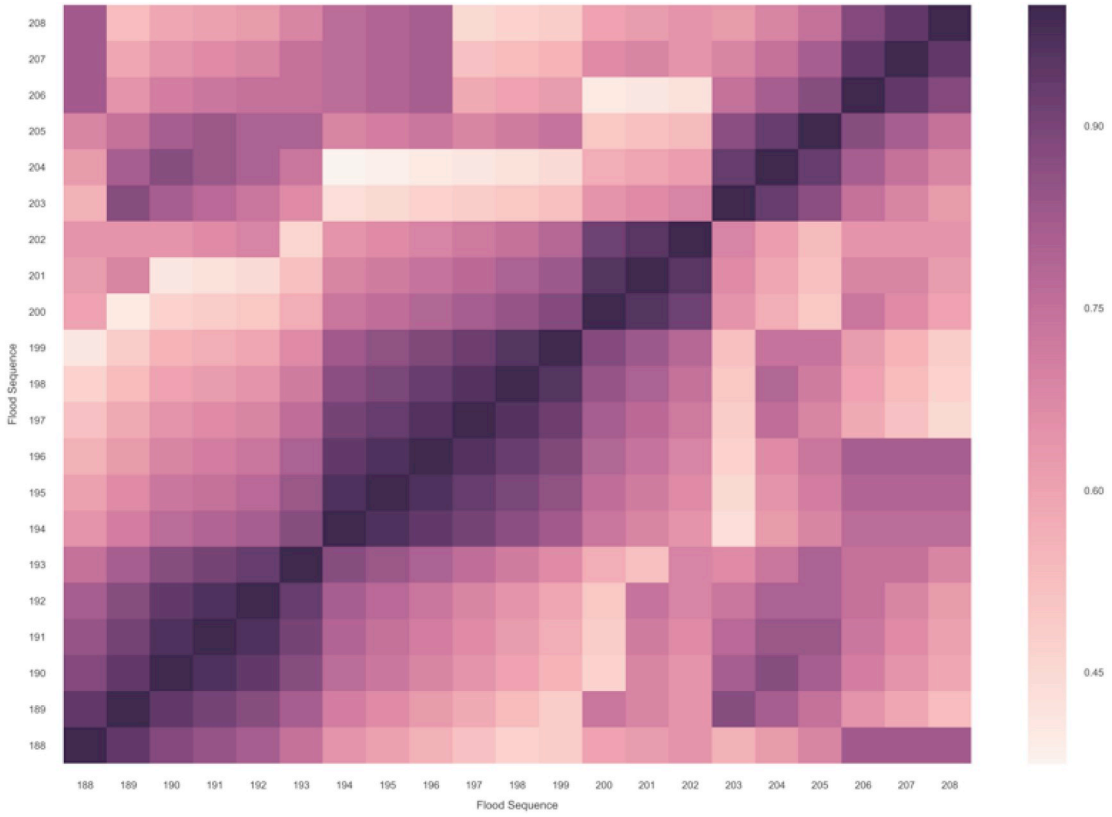


Fig. 17. Similarity analysis for alarm flood clusters (part III).

including data acquisition, pre-processing, KPI evaluation, and visualization is shown in Fig. 7.

4. Industrial case study and results

This section demonstrates the implementation of the developed method. An industrial Alarm and Event log is used to demonstrate the functionality. The sample data-set details are given in Fig. 8. The Alarm and Event log has various attributes such as event stamp (date and time of activation), tag Name (provide the instrument tag details), description, alarm type, priority and msgtype. The data for three days of a plant is used for analysis with 8793 alarms and events in three days for 192 unique instrument tags. The processed data-set is reduced to 4513 after removal of the system alarms and other undesired events as shown in Table 1.

The pre-processed items were analyzed to find the KPIs such as average alarms per day, peak alarms/10 min, average alarms/10 min and priority distribution. These values were compared against the values prescribed in EEMUA 191 guideline (EEMUA-191, 2013) and ISA 18.2 standard (ANSI/ISA-18.2, 2016). The results are summarized in Table 2. The following major observations can be made from the results:

- There are a significant number of average alarms on day 2 and day 3 operations (almost more than 3 times as specified in the standard).
- The number of peak alarms/10 min is considerably high on these days suggests alarm floods during these days.
- The average alarm/10 min are also high, signifying a poor alarm management strategy being followed.
- The prioritization captured in this case is observed as 63/30/5 for low/medium/high priority settings.

After the pre-processing and KPI calculation, the visualization plots are generated for the data-set. The visualization provides user with an opportunity to understand the system performance at a glance and address the issues related to alarm management. The key visualization methods used for this study are:

- **Alarm rate plot:** In order to understand the metrics, alarm rate plots are generated as depicted in Fig. 9. These plots are line graphs which display the trends of alarm rate $R(t)$ by aggregating the number of alarms during a period of 10 min. The shaded dark area under the line represents the KPI alarm flood limit of 10 alarms per 10 min. Each day in the plot is colored with a different color for easy interpretation. When an user hovers over a particular period, the details such as date, time and the number of average alarms at that time can be seen in the top-right corner section of the tool screen. The selection bar at the bottom of the screen provides user with the option to zoom-in and zoom-out of the alarm rate plot and have a better understanding about the system at a time. An interactive plot is shown in Fig. 18.
- **Tree map:** The tree maps (depicted in Figs. 10 and 11) are used to show the bad actors (alarm tags appearing multiple times) during an operation. The darker color and bigger size of the block indicates the presence of the alarm tag multiple times. By hovering over a block, the user can see the number of times the alarm tags have appeared. This is very useful for the user to get a summary of all the tags on one screen with relevant tag activation count information while performing the analysis. An interactive plot is shown in Fig. 19.
- **Heat map:** The heat maps for top 45 alarms tags with the highest count appearing during 24-h time period are plotted in Figs. 12–14. The data values in the heat map are shown as colors; color bar on the right side of the heat map provides the number of times the alarm occurred on each day. The rectangular block shows the count of alarms per 10 min for each tag. Some conclusions that can be drawn from Figs. 12–14 are as follows:
  – For Day 1 that there are some bad actors which appear as alarms

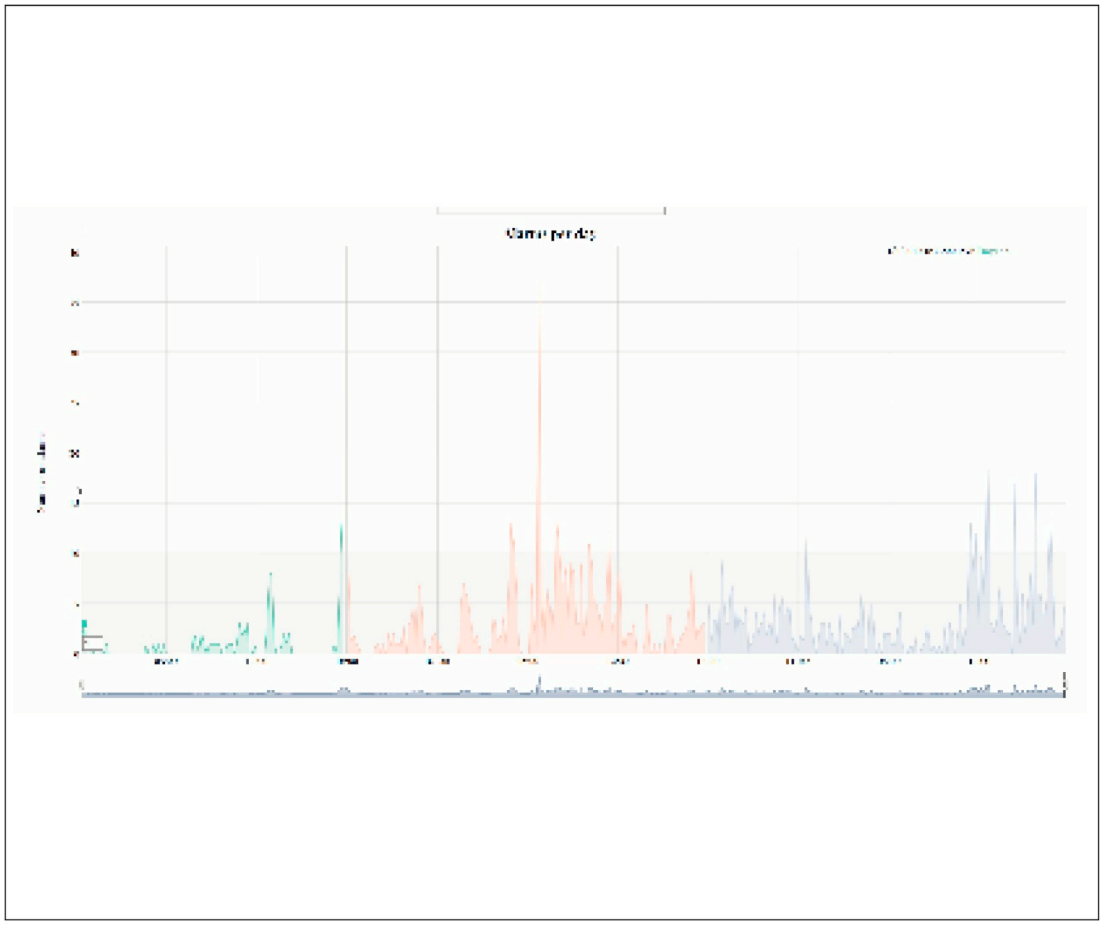


Fig. 18. Interactive alarm/day plot showing alarm information for three days (day1 - green, day2 - orange, day3 - blue) with date time and alarm values on top right corner, scroll option for zoom selection at bottom, and a gray colored line highlighting number of alarms=10 depicting alarm floods in case the value exceeds. **Note: This figure contains an integrated 3d content file and requires flash player access for the pdf reader. Use appropriate pdf reader to open the figure. Follow prompts in the pdf reader and click on the figure to see the media file**. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

several times during a 10-min *e.g.* tag 142 appeared during 1040–1050 min 6 times alone.

– During the same time frame, there are few other tags which appeared multiple times resulting in a total number of alarms per 10-min as >10 alarms. Hence, such maps can be used to performed tag-wise detail analysis and observed the patterns and bad actors during an alarm flood or normal sequences. These tag values can be cross-checked with the tree map plots generated earlier.

- **Similarity plots:** A similar analysis is illustrated which shows a significant number of flood patterns due to several tags activating alarms in a 10 min time frame. The similar pattern was observed during the KPI calculation for Day 2 and Day 3. To understand the relationship between the alarm flood sequence; the correlation plots are developed as shown in Figs. 15–18. A correlation test is used to evaluate the association between two or more variables. In this case, we are using correlation test to find an association between the sequences of alarm floods as described in equation (9). As observed from the figure, sequence 203 & 189 and 204 & 190 are highly correlated. On further investigation, it is observed that these sequences have similar tags which appeared in these sequences. Similar, analysis is performed on different alarm flood sequences to find the similarity between the sequences.

5. Conclusions and future work

Industrial alarm system management has improved over past few years. However, there are still some critical challenges related to alarm management that need to be addressed. The advancement in automation technology and the increase in connected devices has resulted in a higher number of alarms, poor system performance, additional workload on operators, and in some cases has led to abnormal situations. To provide a solution to these challenges, we showed an alarm management framework with four distinct levels - design, rationalize, advance, and intelligent. In addition, this paper proposed a method to reduce alarm flooding by the use of data mining methods on Alarm and Event logs from an industrial control system which can be integrated to ANSI/ISA 18.2 alarm management life-cycle process. A real industrial data set is used to demonstrate the proposed method. As the data-set size increases it is more challenging to calculate the metrics for alarm management manually. The metrics for alarm management also known as KPIs were calculated with the proposed method and bench-marked against the available guidelines and standards. The KPIs were used to understand the alarm system performance and identify gaps at a glance. The visualization tools in the form of alarm rate plots, tree maps, heat map and similarity maps plotted for the data set to infer the data with ease and provide meaningful information related to the bad actors and assist in the overall decision making. The information generated from the proposed method can be used during the different stages of the alarm management life-cycle process for an operating plant with a well established alarm philosophy document and a plan in place. These can be used in either step of the proposed framework *e.g.,* to re-design or re-rationalize the alarm system by revisiting the master alarm database

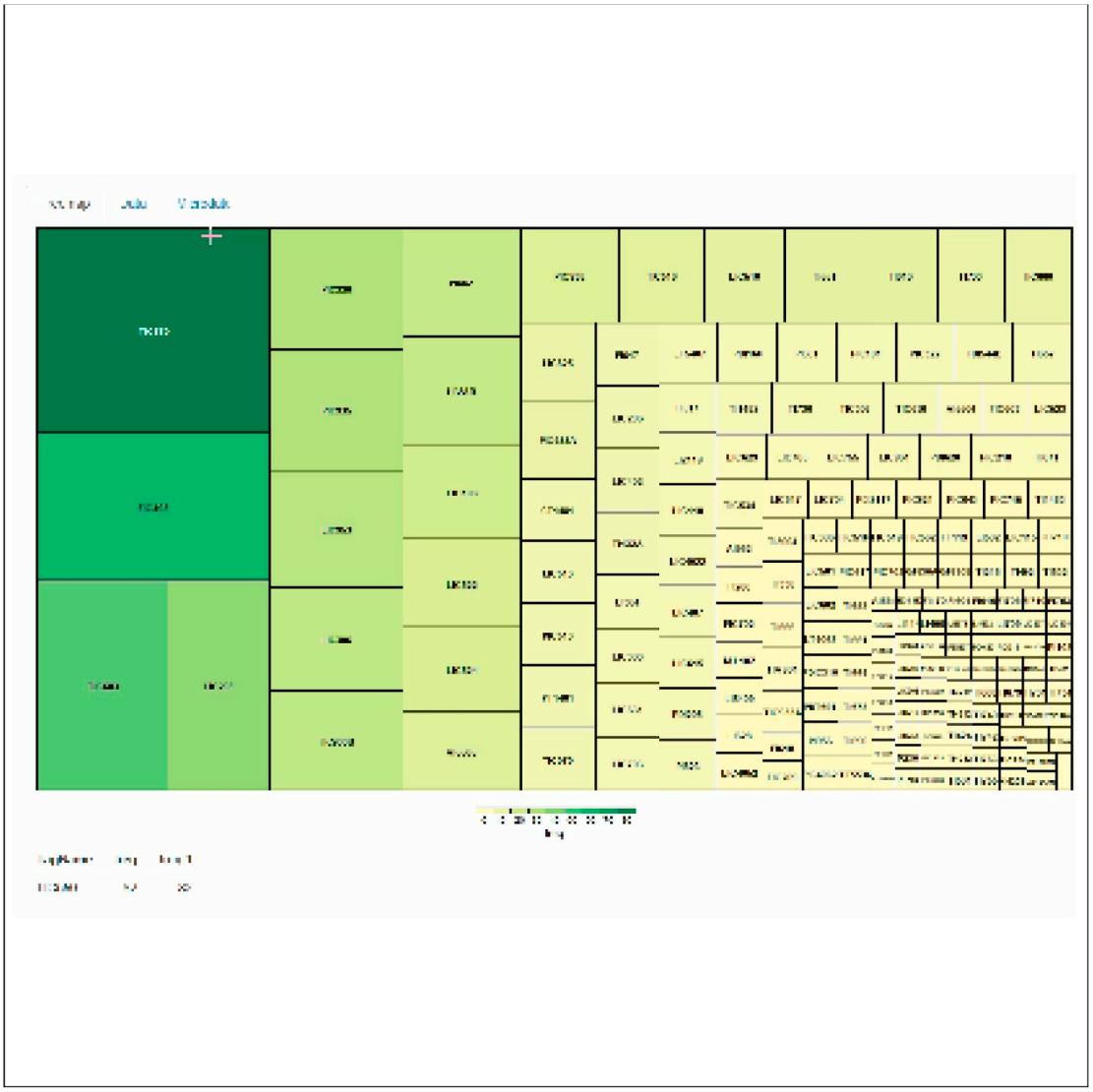


Fig. 19. Interactive plots showing alarm tag information with the number of appearances for each tag number in the bottom left corner, color and size of the box depicts the frequency of tag appearance. **Note: This figure contains an integrated 3d content file and requires flash player access for the pdf reader. Use appropriate pdf reader to open the figure. Follow prompts in the pdf reader and click on the figure to see the media file**. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

and address the requirement of each tag identified as a bad actor; design advanced alarm suppression rules based on the results obtained as similar tag sequences and thereby address issues related to the bad actors and alarm flooding.

We have demonstrated some of the functions of the proposed method. Each DCS system has a different format of alarm and event log report (location of columns, tag specifications and other relevant details). The proposed method requires careful data extraction, processing and integration of expert knowledge while designing the solution. This is required to ensure that no important information is missing and the application provides the correct desired results. In future, additional visualization options can be added to enhance the dissemination of information and the overall user experience. Additionally, process operation data can be used in future to enhance the classification of the alarm and event messages and finding the correlations based on the data between the alarm messages. The proposed method can be run with the help of cloud computing tools and can be made accessible to more users based on the remote functionality. This can also be used as a head start and further developed to design tools to address the requirements of real-time OT services for alarm management as described in OPA framework. The proposed solutions need further research and development and can be subject of future studies.

Acknowledgements

The authors would like to thank the anonymous reviewers for their feedback and suggestions. The authors would like to express their deepest gratitude, appreciation and respect for the late Professor M. Sam Mannan, for his guidance and support. The authors acknowledge funding from the Mary Kay O'Connor Process Safety Center and partial support from Texas A&M Energy Institute.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jlp.2019.103959.

References

Ahmed, K., Izadi, I., Chen, T., Joe, D., Burton, T., 2013. Similarity analysis of industrial alarm flood data. IEEE Trans. Autom. Sci. Eng. 10, 452–457.

ANSI/ISA-18.2, 2016. Management of Alarm Systems for the Process Industries. International Society of Automation (ISA).
Burnell, E., Dicken, C., 1997. Handling of repeating alarms. In: Stemming the Alarm Flood (Digest No: 1997/136), IEE Colloquium on. IET 12–1.
Carrera, J.P., Easter, J.R., 1991. Advanced alarm management in the aware system. In: Nuclear Science Symposium and Medical Imaging Conference, 1991., Conference Record of the 1991 IEEE. IEEE, pp. 1389–1393.
Charbonnier, S., Bouchair, N., Gayet, P., 2016. Fault template extraction to assist operators during industrial alarm floods. Eng. Appl. Artif. Intell. 50, 32–44.
Dorgo, G., Pigler, P., Abonyi, J., 2018. Understanding the importance of process alarms based on the analysis of deep recurrent neural networks trained for fault isolation. J. Chemom. 32, e3006.
EEMUA-191, 2013. Alarm Systems: a Guide to Design, Management and Procurement, vol. 3 EEMUA Edition.
Folmer, J., Vogel-Heuser, B., 2012. Computing dependent industrial alarms for alarm flood reduction. In: Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on IEEE, pp. 1–6.
Fullen, M., Schüller, P., Niggemann, O., 2017. Defining and validating similarity measures for industrial alarm flood analysis. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN) IEEE, pp. 781–786.
Goel, P., Datta, A., Mannan, M.S., 2017a. Application of big data analytics in process safety and risk management. In: 2017 IEEE International Conference on Big Data (Big Data) IEEE, pp. 1143–1152.
Goel, P., Datta, A., Mannan, M.S., 2017b. Industrial alarm systems: challenges and opportunities. J. Loss Prev. Process. Ind. 50, 23–36.
Grosdidier, P., Connor, P., Hollifield, B., Kulkarni, S., 2003. A path forward for dcs alarm management. Hydrocarb. Process. 82, 59–68.
Higuchi, F., Yamamoto, I., Takai, T., Noda, M., Nishitani, H., 2009. Use of event correlation analysis to reduce number of alarms. In: Computer Aided Chemical Engineering, vol. 27. Elsevier, pp. 1521–1526.
Hu, W., Chen, T., Shah, S.L., 2018a. Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods. IEEE Trans. Ind. Electron. 65, 7290–7300.
Hu, W., Shah, S.L., Chen, T., 2018b. Framework for a smart data analytics platform towards process monitoring and alarm management. Comput. Chem. Eng. 114, 225–244.
Jain, P., Mentzer, R., Mannan, M.S., 2018a. Resilience metrics for improved process-risk decision making: survey, analysis and application. Saf. Sci. 108, 13–28.
Jain, P., Pasman, H.J., Waldram, S., Pistikopoulos, E., Mannan, M.S., 2018b. Process resilience analysis framework (praf): a systems approach for improved risk and safety management. J. Loss Prev. Process. Ind. 53, 61–73.
Jain, P., Pasman, H.J., Waldram, S.P., Rogers, W.J., Mannan, M.S., 2017. Did we learn about risk control since seveso? yes, we surely did, but is it enough? an historical brief and problem analysis. J. Loss Prev. Process. Ind. 49, 5–17.
Kondaveeti, S.R., Izadi, I., Shah, S.L., Black, T., Chen, T., 2012. Graphical tools for routine assessment of industrial alarm systems. Comput. Chem. Eng. 46, 39–47.
Lai, S., Yang, F., Chen, T., 2017. Online pattern matching and prediction of incoming alarm floods. J. Process Control 56, 69–78.
Lucke, M., Chioua, M., Grimholt, C., Hollender, M., Thornhill, N.F., 2018. Online Alarm Flood Classification using Alarm Coactivations, vol. 51. IFAC-PapersOnLine, pp. 345–350.
Lucke, M., Chioua, M., Grimholt, C., Hollender, M., Thornhill, N.F., 2019. Advances in alarm data analysis with a practical application to online alarm flood classification. J. Process Control 79, 56–71.
Mannani, Z., Izadi, I., Ghadiri, N., 2019. Preprocessing of Alarm Data for Data Mining. Industrial & Engineering Chemistry Research.
Mehta, B.R., Reddy, Y.J., 2014. Industrial Process Automation Systems: Design and Implementation. Butterworth-Heinemann.
Niyazmand, T., Izadi, I., 2019. Pattern mining in alarm flood sequences using a modified prefixspan algorithm. ISA Trans. 90, 287–293.
Pariyani, A., Seider, W.D., Oktem, U.G., Soroush, M., 2012. Dynamic risk analysis using alarm databases to improve process safety and product quality: Part i–data compaction. AIChE J. 58, 812–825.
R Core Team, 2018. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.
Rossum, G., 1995. Python tutorial. Technical Report CS-R9526. In: Centrum voor Wiskunde en Informatica (CWI). Amsterdam.
Rothenberg, D.H., 2009. Alarm Management for Process Control: a Best-Practice Guide for Design, Implementation, and Use of Industrial Alarm Systems. Momentum Press.
Wang, J., Li, H., Huang, J., Su, C., 2015. A data similarity based analysis to consequential alarms of industrial processes. J. Loss Prev. Process. Ind. 35, 29–34.
Yang, F., Shah, S.L., Xiao, D., Chen, T., 2012. Improved correlation analysis and visualization of industrial alarm data. ISA Trans. 51, 499–506.
Yang, Z., Wang, J., Chen, T., 2013. Detection of correlated alarms based on similarity coefficients of binary data. IEEE Trans. Autom. Sci. Eng. 10, 1014–1025.