Alarm Correlation to improve industrial fault management

M. A. BENATIA [*] A. LOUIS [*] D. BAUDRY [*]

[*] LINEACT-CESI Laboraory, 1, Rue G. Marconi 76130
Mont-Saint-Aignan, France (e-mail: mbenatia, alouis,
dbaudry@cesi.fr)

Abstract: Actual alarm systems used in manufacturing applications lacks explanation and indication of the root causes, which results in a poor decision making. In addition, manufacturing systems are more and more complex, so relaying on human operators for alarm information management becomes impossible. For this, a computerized tool to support human operators (i.E., decision support for information management system) is needed and would increase analytical capability for alarm analysis. To this effect, we introduce in this paper an autonomous data mining method to search historical alarm logs for the correlations that can represent causal relationships, which can support alarm management and system improvement. We investigate the use of Frequent Pattern Mining algorithm, an enumeration-tree based approach, for extracting relationships and automatically detect correlation between industrial alarms. Due to the time indexation of the alarm events, we adapt the algorithm in order to take into account the duration between alarms when extracting the itemsets. Filtered rules where evaluated according to the *Minimmum support & Confidence* framework. Obtained results show that FPM algorithm can derive very useful knowledge on system behaviour allowing the identification of alarm subsequence with the corresponding root cause.

Keywords: Alarm floods, causality inference, alert correlation, Association Rule mining.

1. INTRODUCTION

Industry 4.0 (or industry of the future) can be seen as a convergence point between traditional operational and information technologies (ERP, MES, etc.) and data-driven approaches such as Machine Learning (ML), Big Data Analytics, IoT and cloud computing. Nowadays, impressive progress & level of integration has been made in monitoring technologies and industrial data analysis (i.e., Industrial Artificial Intelligence (AI)) thus allowing new condition-based maintenance (CBM) capabilities. The principal goals/utilities for such CBM systems are to assess product quality while decreasing its respective production cost by automating the maintenance process.

Implementation of this Service-Oriented Architecture (SOA) allows embedded components to invoke actions remotely, using Web Services for example (e.g., REST API) O'Donovan et al. (2016). These new highly connected manufacturing industries represent a data-intensive production environment (ubiquitous sensors, embedded analytical capabilities, etc.) whose objective is to improve production processes (better product quality), increase equipment availability (better maintenance management) and reduce energy consumption and carbon footprint Jin et al. (2016).

In this work, we focus on the second objective which is the predictive maintenance strategies. In this scope, data-driven or empirical approaches have been largely applied. This family of approaches refers to models using historical equipment data to determine, in a statistical or probabilistic manner, a deviation (or degradation) from an expected normal operating conditions Schwabacher (2005). Data Mining & Machine Learning are the two most important scientific domains interested on predictive maintenance. Generally, the goal of DM and ML approaches is to identify and learn a representation of system behavior from equipment historical data in order to allow prediction and prognosis on future behavior/status.

Despite the fact that data-driven CBM algorithms can extract knowledge about an asset status, they are generally hard to interpret making it difficult to integrate them in a real industrial system. More recent works are considering Frequent Pattern Mining (FPM) as an innovative solution to failure prediction. Its advantages are: the simplicity of the model and its interpretation. In addition, they can be trained in both supervised and unsupervised learning processes. This make them useful for both classification and rule identification tasks. Due to their simplicity and ability to use raw transaction databases, they were widely applied to marketing and finance. A dedicated analytics domain has emerged known as *Market-Basket Analysis* Aggarwal et al. (2002). More recently, this kind of models were applied to quality inspection and condition-based maintenance. For example, Zongchang Liu et al. Liu (2018) consider the alarm events of a maintenance system as a labelled sequence to be classified (failure or not). The sequence is composed of multiple transactions representing the sensor states in time. Once the sequence collected, the authors use a FPM algorithm in order to derive association rules between itemsets of the sequence. The used algorithm was the well known *A-priori algorithm* which is widely

[⋆] This work is supported by the French Governments Program Investments for the Future through the project Défi&Co.

used in *Market-Basket Analysis*.

Industrial systems generally include several conveying equipments that helps in package/product routing. This asset is usually equipped with condition monitoring systems that provides rich information about system health and status. Due to the system's complexity (ie. several sensors and parts, changing working regimes, etc.), maintenance experts have predefined hundreds of alarm events and monitoring variables related to the conveying equipment and its auxiliary system. However, analysing and extracting useful patterns (ie. accurate trouble shooting) from this data remains a very difficult task.

Usually, alarm events happen together as bursts and the main/root cause behind it is difficult to identify. Also, changes in work regimes and signal noise may cause several false alarms, thus involving errors when computing the most frequent itemsets. In this case study, a decision maintenance management system for industrial conveying assets is proposed in order to assist operators in identifying the alarm nature (true/false alarm). The proposed approach takes as input bursts of alarm events and identify the most frequent alarm episodes (i.e., the most common and frequent alarm events sequences). Due to time indexation of the alarm events, we adapt the algorithm in order to take the duration between alarm events when constructing the itemsets. Extracted rules where evaluated according to the *Minimmum support* framework Aggarwal and Han (2014) that integrates two evaluation metrics : confidence and lift of the rule. These rules are then used in order to classify alarm sequences to false/real alarm.

The paper is organised as follows. Section II presents a brief literature review on condition based maintenance and their impact on helping maintenance operators and fault management. Section III explains the proposed approach to extract useful rules from raw alarms log. Section IV presents the obtained results and discussion. Finally, we conclude our work by giving some future perspectives in section V.

2. RELATED WORKS

Nowadays, the majority of industrial systems are monitored by operators using SCADA (Supervisory Control And Data Acquisition) systems. This systems can ensure operator safety and improve productivity by notifying, in real-time, on the health status of industrial assets. They are genrally formed by several layers which are as follows: a field layer, a control layer, and a supervision layer. In case of abnormal situation an alarm is triggered consisting of a binary variable displayed in the GUI to inform operator on the occurred situation. The abnormal situation is generally caused by a process key variable crossing a pre-fixed threshold or by an equipment failure. Due to high sensor integration in manufacturing assets, the number of alarms has tremendously increased complicating the supervision task. In general, operators are often faced with a situation where tens or hundreds of alarms are raised in a short time known as an *alarm flood*. The main goal of researchers is to reduce the number of displayed alarms and try to identify the root cause behind them. The literature works can be organised in two research axes into (1) works that tend to detect unnecessary alarms, and (2) axe of research to reduce the operators' overload. The first research axe refers to approaches which aim to identify and discard irrelevant alarms, thus improving the relevance of an alarm system. Several research works were proposed in this scope including (Vogel-Heuser et al., 2015; Wang et al., 2015; Hu et al., 2018). The second research axe include all the approaches that aim to reduce the number of displayed alarms by making an automatic diagnosis of the process. The principal goal is to support the operator during an alarm flood by proposing a cause to the alarm flood. Some researches done in that way include (Ahmed et al., 2013; Wang et al., 2015; Charbonnier et al., 2016). For more details about alert correlation we recommend to the reader the paper published by Salah et al. (2013) that present a very complete survey, describing the different methods used in industrial process monitoring to detect related events.

Although, efforts have been dedicated to control alert flood for several years, industrial operators at various facilities are still unsure how to manage and streamline them. For example, Hu et al. (2018) proposes an approach to study frequent alarm patterns in alert floods. The main goal behind it is to simplify the dynamic alert identification and suppression, which is a common technique to temporarily suppress predefined sets of alerts. In this scope, similarity index was proposed in (Ahmed et al., 2013), to analyze alarm data during a flood by identifying subsequent alert patterns. Another work presented by Charbonnier et al. (2016) use similarity measures to extract fault patterns from a set of alarms and identify root causes, thus helping industrial practitioners in the diagnostics phase. In this scope, correlation methods can be organized into three main families: (1) similarity based approaches, (2) sequential based approaches, and (3) case-based approaches. The first family of approaches refers to methods that use a similarity index or measure to reduce the total number of alerts by clustering them using their similarities. Approaches can be grouped according to the nature of the used parameter into: attribute-based and temporal based methods. In attribute based methods, the correlation technique uses similarities between attributes or features of the existing alerts to correlate alerts, such in(Valdes and Skinner, 2001; Siraj and Vaughn, 2005; Julisch and Dacier, 2002). In the other hand, temporal based techniques uses temporal time constraints to identify underlying relationships between alerts, such in(Jakobson and Weissman, 1995; Alserhani, 2016). Second family of approaches uses causality relationships among alerts to detect frequent episodes. In this case, a pre-condition parameters are to be predefined and used to represent by a logical formulae using combination of predicates and operators (i.e., AND/OR). The most used algorithms are: Markov models (Zan et al., 2009), Bayesian Networks (Liang et al., 2019), and Neural Networks (Liu and Zhu, 2019). Finally, the third family of approaches refers to those that rely on the existence of a knowledge-base system to represent well-defined scenarios (Salah et al., 2013). In this paper we propose an unsupervised clustering algorithm using the frequent pattern mining algorithm to extract usefulness rules that can support industrial practitioners to identify root causes of failures. The proposed approach is explained in the next section.

3. PROPOSED APPROACH

The alert correlation problem in industrial process monitoring can be seen as a temporal classification or temporal sequence-aware labelling problem. The purpose of sequence labelling is to assign a sequence of labels (anomaly types), derived from a fixed and finite alphabet, to a sequence of data received as input (data-stream). The entries/input represent x sequences of fixed size (i.e. real value vectors) and the targets are discrete label sequences z, derived from a finite alphabet $\mathcal{L}$.

Thus, the sequence labelling problem can be mathematically represented as follows: Let S be the set of learning examples independently drawn from a known and fixed a priori, distribution $\Omega_{X\times Z}$. The input space $\chi = (R^M)^*$ is the set of all sequences of size M (real value vectors). The target space $Z = \mathcal{L}^*$ is the set of all sequences on the finished label alphabet $\mathcal{L}$.

Each element of S can thus be represented as a sequence pair (x,z). If the input/output sequence is indexed over time, the values of x and z are called timesteps. The classification task is then:

"use S to learn a temporal-sequence labelling algorithm: $h : \chi \to Z$ to label the sequences of a test set $S' \subset \Omega_{\chi\times Z}$ disjoint from set S, so as to minimize an error metric, which is independent from the identified task"

The proposed model must take into account the problems (challenges) mentioned above, namely: intra-channel variation, intra-channel temporal variation, cross-channel temporal variation, noise and segmentation. To do this, it is necessary to study the system in order to define:

- The indicators that must be taken into account in set X and that do not depend on time or variation in system load
- The most appropriate algorithm (model)
- The task-independent error measurement (Euclidean distance, Levenshtein, Mahlanobis) most appropriate for the type of sequence studied

In addition, generally in most industrial cases, maintenance operators rely on sensor or alarm events to detect equipment health degradation/status. According to Zongchang Liu PhD thesis Liu (2018), the main challenge of condition-based monitoring comes mainly from two aspects:

(1) *Spatio-temporal clustering*: in most cases, alarms happen in groups, and a group of alarm events that are close in time or space may represent multiple failure modes.
(2) *False alarms and failure propagation*: usually one alarm will represent the true incident and the other ones are just triggered by the main alarm. Also, we can observe several false alarms caused by operational regime change on uncertain data collected from sensors.

In order to solve the aforementioned challenges, a clustering approach is required. Based on works done in Liang et al. (2006) which deals with failure detection from network logs, we propose the following steps:

(1) *Failure events extraction and categorization*: in this step, we identify *failure-events* which are all the events with a great severity (*FATAL*) that causes production system crashes. This part was based on maintenance operators knowledge, thus leading to a *human in the loop* approach.
(2) *Temporal clustering (at a single location)*: Failure events from the same location often occur in bursts, referred to as *clusters*. Identifying such clusters needs the definition of a *Time-window* in order to group events into itemsets. In this paper, two events belong to the same cluster if the gaps between them are less than a predifined threshold T_{th}
(3) *Spatial clustering (across multiple locations)*: In order to capture the failure propagation in a spatially distributed manufacturing system, we regroup alarm events based on their locations. This task is generally referred to as *spatial filtering* and removes failures that are close to each other (by defining a threshold S_{th})

Based on the aforementioned steps, we can say that the problem of clustering is equivalent to identify and extract frequent itemsets in the given events sequence. To this end, we choose to implement a frequent pattern mining algorithm with time index consideration, in order to extract very useful rules (sequence of events). We hope that the extracted rules can be used in order to identify root causes and predict alarm events. Once rules are extracted they can also be stored as a knowledge graph which can be useful for false alarm detection. The proposed architecture for false alarm identification and isolation is represented in Figure 1.

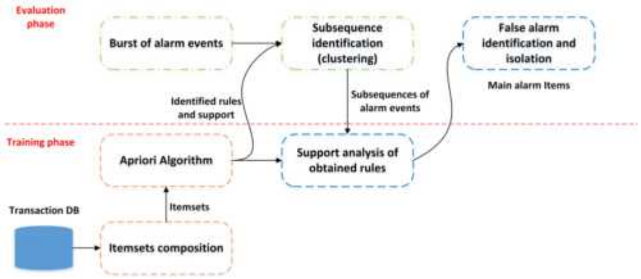
Evaluation phase
Burst of alarm events
Subsequence identification (clustering)
False alarm identification and isolation
Main alarm Items
Identified rules and support
Subsequences of alarm events
Training phase
Apriori Algorithm
Support analysis of obtained rules
Transaction DB
Itemsets
Itemsets composition

Fig. 1. Proposed false alarm identification model

Since the publication of Agrawal, Imielinski, and Swami article on association rule mining Agrawal et al. (1993), FPM has known an increasing interest in several scientific domains (eg.: recommendation systems, PHM, Market analysis, etc.). Despite of its shorter history, it was largely adopted by a variety of research communities leading to a large number of papers that dominate the earlier data mining conferences. According to C. C. Aggarwal Aggarwal and Han (2014), the problem of FPM can be stated as follows:*"Given a database $\mathcal{D}$ with transactions $T_1, \ldots, T_N$, determine all patterns P that are present in at least a fraction s of the transactions."*, where the fraction s is referred to as the *minimum support*. However, in FPM we generally focus on identifying frequently co-occuring events, regardless of their appearance order. for this purpose, Sequential pattern mining (SPM) was introduced, where co-occurence order is considered.

3.1 Itemsets generation phase

For extracting frequent patterns and association rules, FPM and SPM are generally based on the *support framework*. This framework was designed in order to extract

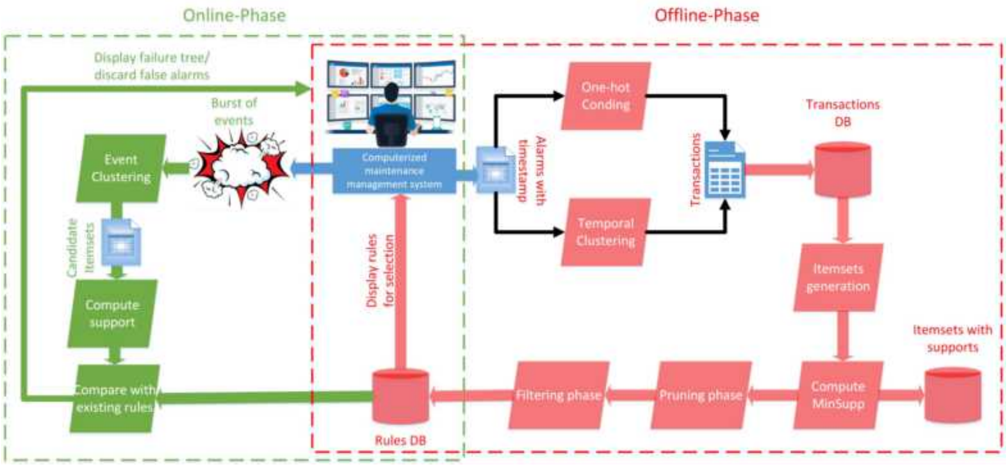

Fig. 2. Global architecture of the proposed model

patterns for which the occurrence frequency is greater (or equal to) than a predefined threshold called *minimum support*. This framework has a convenient property referred to as the *level-wise* property which enables the design of bottom-up models for space exploration. The first proposed algorithms for resolving FPM problems were referred to as *Apriori*-like join-based methods. It generally composed of two phases : itemset generation (*level-wize exploration*) and constraint checking (*rules identification/filtering*). The first phase consist on generating itemset with an increasing itemset-size based on the following property: *"a (k+1)-pattern may not be frequent when any of its subsets is not frequent"*. Afterwards, rules are extracted by testing the genrated itemset against the transaction database. The most frequent itemset satisfying the *minSupp* constraint are retained and stored. Using this strategy, FPM can be viewed as *enumeration trees* that provides different exploration strategies as: depth first, breadth first, or a hybridization between them.

In this paper, we investigate the use of FPM/SPM methods & algorithms for predicting equipment failure events. We implement two different approaches: *Apriori* & *FP-growth*. The first one is considered as the pioneer algorithm for association rule mining and is considered as a *level-wize & breadth-first* algorithm. This algorithm can be also implemented with a *depth-first* strategy where, first the nodes in the root (depth = 0) are constructed, next the algorithm tries to generate nodes at depth 1, and so on Kosters and Pijls (2003). This process of generating candidate itemsets generally implies repeatedly scan the database to count the support of each pattern.

The second approach, called *FP-Growth*, uses a *depth-first* search strategy that limits the repeated scanning of the transaction database. Also, the *FP-growth* algorithm uses a *TreeProjection* method in order to reduce the counting work. This algorithm is based on an important observation is that if a transaction is not relevant for counting at a given node in the enumeration tree, then it will not be relevant for counting in any descendant of that node. Fur further explications on these two algorithm, we recomend to the reader the two following papers: Quadrana et al. (2018) & Aggarwal and Han (2014).

3.2 Filtering process

As described in the previous section, frequent pattern identification in an event sequence (with co-occurrence constraints) consist on finding rules which enables the prediction of specific items occurrence based on the occurrences of the other items in a given transaction. Given a set of transactions $\mathcal{T} = \{T_1, T_2, \ldots, T_n\}$, and let X and Y be two distinct itemsets of a given sequence s. Association rule mining basics can be summarized as follows:

(1) *The support*: it defines the fraction of total sequences that contain (support) a specific itemset. The support of the itemset Y, denoted $sup(Y)$ indicates the frequency (occurrence) of the itemset Y in the transaction database.
(2) *Rule support, Confidence, Lift & conviction*: if we consider a rule $X \rightarrow Y$, its support, confidence, lift and conviction are given by the following equations

$$sup(X \rightarrow Y) = \frac{sup(X \cup Y)}{Card(T)} \tag{1}$$

$$conf(X \rightarrow Y) = \frac{sup(X \cup Y)}{sup(Y)} \tag{2}$$

$$lift(X \rightarrow Y) = \frac{conf(X \rightarrow Y)}{sup(Y)} \tag{3}$$

$$conv(X \rightarrow Y) = \frac{1 - sup(Y)}{1 - conf(X \rightarrow Y)} \tag{4}$$

The first equation is the basic function that defines the support framework. It defines the marginal probability of an event occurring, meaning the proportion of transactions that contains the itemset of interest. The support of a rule is computed as the ratio between the frequency of this rule ($X \rightarrow Y$) and the length of the itemsets database (i.e., number of itemsets) and is generally expressed as *sup(I), Supp(I) or support(I)*. As mentioned in Aggarwal (2015), *"... Clearly, items that are correlated will frequently occur together in transactions. Such itemsets will have high support..."*. To this end, a *minimum support* threshold must be defined before the algorithm execution.

In the other hand the *Confidence* value can explained as the conditional probability of event Y knowing that X is $TRUE$ (i.e., $p(Y/X)$). It is computed as the ratio between the *sup(I)* (i.e., support of itemset I) and the probability of $(X = TRUE)$. Knowing this, higher is the *Conf(I)* value better is the quality of the rule *(I)*.

These two measures are then used as observers when searching for rules of interests. As stated by Fjällström (2016): *"... One of the problems with association mining is that it is very computationally heavy to find and calculate the support and confidence for all rules in a dataset..."*. In fact, the number of possible rules that can be extracted from a d item dataset is assumed to be equal to: $R = 3^d - 2^{d+1} + 1$ Tan (2018). Using the *minimum support framework* makes the rule identification problem computationally feasible by first computing the support of each rule, then calculate the confidence measure only for the rules that have support higher than the user defined threshold.

More generally, this similarity measures belongs to two groups, deending on the range of the obtained value: normalized (i.e., ranging from *0* to *1*) and non-normalized (i.e., great or equal to zero ≥ 0). The first family of measures has the advantage of meaningfully compare a pair of two objects, but they generally need a predefined threshold value that defines if two objects are connected or not. As stated in, *"... choosing this threshold value is problematic, as any number is bound to be somewhat arbitrary....*The second famﬁlly of measures (i.e., non-normalized) can return any positive value, thus resulting in a natural threshold: *1*. In effect, if the non-normalized measure is greater than *1*, this means that the two considered events (i.e., or itemsets) have more in common than they have different. Thus, a value equal to *0* mean that this two events, respectively itemsets, are disconnected.

Two non-normalized measures are investigated in this paper which are: *Lift* and *Conviction*. The *Lift* measure can be defined as the difference degree between the conditional probability $p(Y/X)$ and the marginal probability $p(Y)$. This measure is represented by the equation (3). In the other hand, *Conviction* (i.e., equation (4)) can be interpreted as a comparaison measure between the probability that a failure event X appears without the failure event Y, if they were dependent, and the actual frequency of incorrect predictions (i.e., frequency of the appearance of X without Y).

4. RESULTS & DISCUSSION

4.1 Data description & Feature Engineering

Usually, alarm events happen together as bursts and the main/root cause behind it is difficult to identify. Also, changes in work regimes and signal noise may cause several false alarms, thus involving errors when computing the most frequent itemsets. In this case study, a decision maintenance management system for industrial conveying assets is proposed in order to assist operators in identifying the alarm nature (true/false alarm). The proposed approach takes as input bursts of alarm events and identify the most frequent alarm episodes (i.e., the most common and frequent alarm events sequences).

The data investigated in this study was collected from a tilt-tray sorting condition monitoring system in one of the biggest sorting center /hub in France. The sorting system can be viewed as a sophisticated and highest-throughput automated material handling system. The system has several embedded sensors and automated alarm system based on expert's predefined thresholds. The sorting center has processed more than 650 millions packages in the last 10 years, with a rate ranging from $340K \sim 500K$ packet per day. It deserves more than 90 different destinations with a processing mean-time of 4min/packet. Howver, this center suffers from false alarm events that count for 20% of the total maintenance time per day (i.e., 3/15h). The goal of this study is to reduce this wasted time in verifying the authenticity of the alarm event, thus increasing the processing capacity of the center (i.e., more packets processed).

The whole dataset contains more than 27 millions records collected in a one year period (6 months in 2016 & 6 months in 2018) considering different working regimes and different packet processing rate. Due to system complexity, the considered Proof of Concept (PoC) was limited to a specific supply line (coming packets), a sorting asset (route packets) and four distribution lines (output). The restricted dataset (i.e., PoC area) contains several duty cycles of real operation data. We have identified 59 unique alarm events occurring 23772 times. For confidentiality aspects, alarms names where replaced by abbreviations (eg., BCB3 corresponds to a *jamming* event in distribution line B3). Table 1 summarizes some abbreviations of the most common alarm events.

In order to extract useful patterns from this subset dataset, a feature engineering phase is needed. In this phase, we are principally interested in: removing perturbation (i.e., noise on data) and constructing the itemsets from the sequence of alarm events. The first step consist on a simple data cleansing algorithm that was designed with the maintenance operators. For example, an alarm event ("*Die-Back Activated*") was suppressed due to its non-informative nature (i.e., economic mode). This alarm happens frequently and is always ignored by maintenance technicians.

To be aligned with the concepts in frequent pattern mining, we assume that after each occurrence of a target failure a maintenance action treats the failure in such way that it becomes independent of the previous one. Although this might not be always true, it is a reasonable assumption in most cases. Furthermore, throughout the rest of the paper we consider the time as a discrete variable that is measured in days in order to avoid including possible time periods of non-utilization of an asset during night and second.

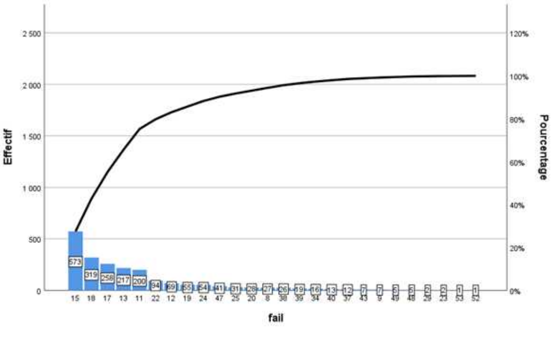

Fig. 3. Frequency distribution of important alerts (10 min)

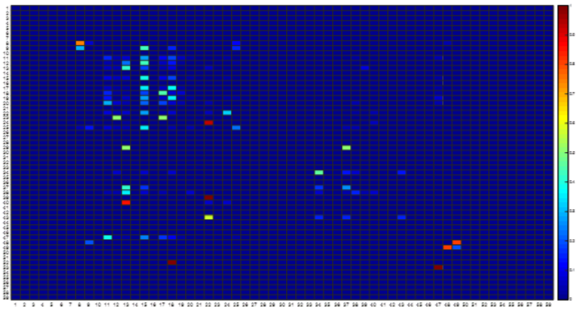

Fig. 4. Correlation heatmap between alerts

Table 1. Example of used abbreviations

Abreviation	Event
MLS	oversized length measurement
AU	Emergency Stop
BCB#	Cell Jam
DDMT	motor circuit breaker failure

Table 2. Example of failure events frequencies

Event	Frequency
MLS	1867
AU	1267
BCB1	1637
DDMT	640

Table 3. Values of mean, min and max of the duration between two consecutive events

Variable	Value (seconds)
Minimum	1
Maximum	180
Median	8
Mean ($\mu_{Duration}$)	9
std ($\sigma_{Duration}$)	3

In a typical FPM experiment, one starts collecting transaction data (i.e., set of transactions done by individuals) and organize a them as an itemsets database that consists of several transactions done by the population of monitored individuals. In order to structure our failure data accordingly, we must cluster our continuous observation (i.e., sequence of failure events) into several itemsets that belongs to the same incidents. The assumption to make on top of this section is that main alarm happens first followed by serial alarms. As mentioned in the previous section, two types of clustering can be considered: *Temporal & Spatial clustering*. Du to the restriction of the PoC (Proof of Concept) only the temporal clustering was considered. Therefore, we have computed the mean duration ($\mu_{Duration}$) and standard deviation ($\sigma_{Duration}$) between two consequent failure events. Obtained results are shown in Table 3. Using this result, we can generate duration threshold for which two alarm events will be considered belonging to the same itemset by using a normal probability distribution function with $mean = \mu_{Duration}$ and $std = \sigma_{Duration}$ (i.e., Table 3). The number of itemsets after compression is 919 transactions, and a summary list of some identified alarm itemsets are shown in Table 4.

Table 4. Summary of the Itemsets database

ID	First Item	Nbr of items
1	*"motor circuit breaker failure"*	5
2	*"oversized length measurement"*	6
3	*"Cell-Jam B1"*	8
4	*"Cell-Jam B3"*	4
5	*"Power failure"*	3

Once the itemsets were identified using the temporal clustering approach proposed in the previous section, we transform the obtained transaction dataset to a one-hot coding matrix. The matrix rows represents observed transactions/itemsets and the columns represents the failure events ID. It is then used in both *Python* and *TANAGRA* in order to extract frequent itemsets. Figure, shows an example list of the obtained frequency of each generated itemset from the Transaction DB. We will define in the next subsection the rule extraction methodology and results.

4.2 Rule identification & Evaluation

In order to extract useful rules from the transaction DB, we use two algorithms with different exploration methodologies as explained in the previous section, which are: *A Priori* & *FP-Growth.* In addition, a variant version of the *A Priori* algorithm that implements a test & validation framework is used in order to assess rules and select only the rules that passes the test phase. To implement this models (i.e., APriori & FP-Growth), we use both Python language with the *MLXTEND* library & R-Studio tool for the visualization of the extracted rules. In order to integrate a train & test framework, as in generic Machine Learning models, we implement a *Python* script that splits the transaction DB to train & test datasets. The train dataset is used in order to exract rules, where the test dataset is used to assess this rules. This is done by comparing the model metrics (i.e., MinSupp, MinConf, Lift and Conviction) between rules extracted from the train dataset to those obtained in the test phase. However, due to the nature of the problem (i.e., association rule discovery & extraction) we cannot compute any ROC Curve or confusion matrix.

Before running the models, we have to set parameters (i.e., support, confidence, max items, etc.). In order to get a prediction rule, we have set the maximum length of the consequent event as 1. Doing so, all the discovered rules will have a one-event consequence, leading to a prediction rule (if $e_1 = TRUE$ & $e_2 = TRUE$ Then e_n will be equal to $TRUE$ also with a certain probability). We also set *MinSupp* to 0.01 and *MinConf* to 0.95 giving a confidence interval of 95%.

Tab 5, Tab 6 and Tab7 summarizes the used parameters, sample & item characteristics, and the obtained results (i.e., rules with metrics) respectively. The *Apriori* approach identified 726 rules with computing the Lift and conviction parameters of each rule. From this extracted rules, we were interested by those leading to an emergency stop of the system. In total, 38 rules were identified for the emergency stop issue. This subset rules are presented in Tab 7. In the other hand, and due to its pruning strategy (i.e., depth-first), FP-Growth algorithm discovers 445 rules with using the same parameters. The difference

Table 5. A-PRIORI parameters

Parameter	Value
Minimum Support	0.02
Minimum confidence	0.95
Maximum Rule Length	6
Lift (for filtering) ($\mu_{Duration}$)	1.1
Learning set (A-Priori MR)	50%
number of repetitions (A-PRIORI MR)	100

Table 6. Sample & Items characteristics

Sample	Nbr of Itemsets
Training Set	459
Test Set	460
Itemset Characteristic	**Value**
All Items	17
Filtered Items (MinSupp)	11
Cardinality(Itemset) = 2	22
Cardinality(Itemset)=3	15
Cardinality(Itemset)=4	5
Extracted Rules (APRIORI)	25
Extracted Rules (APRIORI MR)	57
Extracted Rules (FP-Growth)	56

Table 7. Some samples of extracted rules

Antecedent(s)	Consequent(s)	Supp	Conf	Lift	Conv.
{P-DP-Err, DA-F2.24}	{Emergency stop}	0.039	1	21.9	99.99
{DA-F2.24.2, BCB2}	{Emergency stop}	0.0119	0.785	22.564	4.504
{MLS, BCB2}	{Emergency stop}	0.0206	0.95	27.282	19.303
{MLS, BCB2, DA-F2.24, BCB1}	{Emergency stop}	0.0195	1	28.718	45.684
{CBC, P-DPErr, BCB1}	{Emergency stop}	0.0141	1	28.718	22.536

is the usage of the *MinLift* parameters, with FP-Growth we can select only rules that respects MinSupp, MinConf and MinLift parameters. However, the same 38 rules were discovered for the emergency stop issue. Figure 5 and Figure 6 summarize the discovered rules with their respective parameters values.

Figure 5 shows the visualization of the 445 association rules found from the analysis of the system obtained by the two algorithms according to the 3 statistical indicators (support that reflects the occurrence of the rule, confidence that reflects the accuracy and the lift that indicates the reliability of the rule). The selected rules should have at least one high value in the indicator value scale, which will give a more reliable, and recurring rule. Once obtained, this rules were presented to the stakeolders that gives us some feedbacks in order to filter the 445 rules. After the meeting we were interested in exploring 10 rules, that are presented in Figure 6 . This rules will be implemented in an online manner in order to detect root causes and identify potential false alarms. In this phase the occurring events are organized as itemsets according to a predefined time window (i.e., bursts). Our idea is to extract all the subsets of these itemset, using a clustering approach, that have high support from the obtained rules set. We will compute the support of these rules using the transaction Data Base and visualize the most frequent ones (i.e., that have $sup(I) > MinSup$).

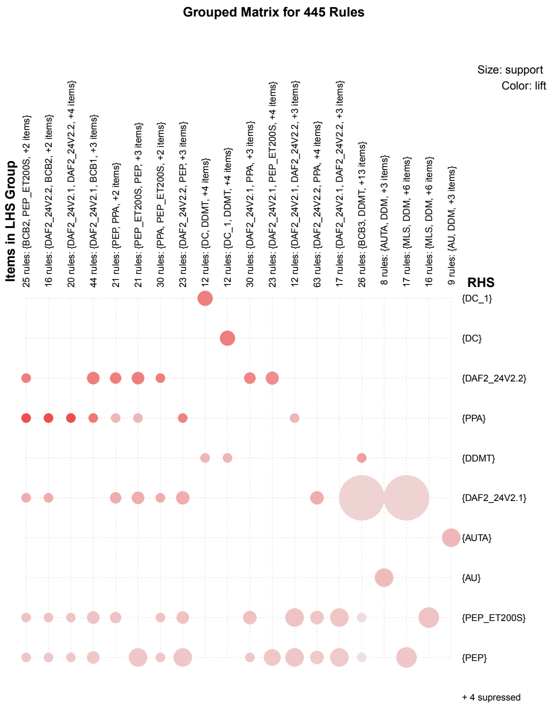

Fig. 5. Grouped matrix of the association rules (LHS: left hand side, RHS: Right hand side)

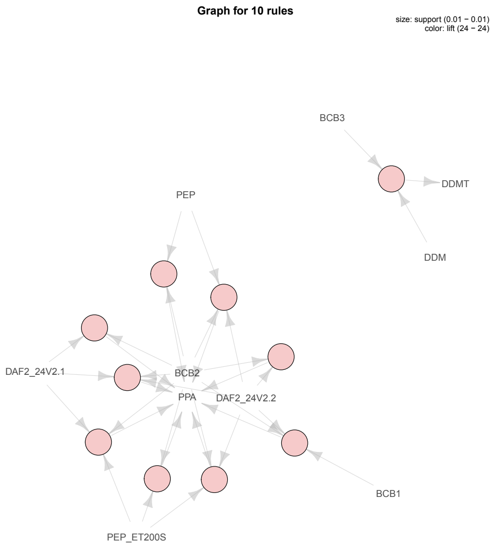

Fig. 6. Graph representation of the 10 most frequent rules: circle represents the *support* & colour of circle represents the *lift* value of each rule

5. CONCLUSION

In this paper, we have discussed the potential of Frequent Pattern Mining models in assessing predictive maintenance issues. As presented in this paper, FPM models can be used as unsupervised machine learning models in order to discover new patterns and relationships between maintenance alarms/events. The output of such methods is given as association rules between alarm events. Once created, the model can be used to solve several predictive maintenance issues such as: predict failures, detect false alarms of identify root causes of a specific failure.

The obtained results shows that FPM can be an innovative option to implement predictive maintenance in manufacturing enterprises without any need of external sensors implementation. The data sources used in this paper are largely available in most enterprises. For example, using such approach can resolve false alarm detection thus saving precious time to maintenance operators.

ACKNOWLEDGMENT

This work is supported by the French Governments Program Investments for the Future through the project Défi&Co & by the CISCO-VINCI Energies Industrial Chair on the Factory of the Future.

REFERENCES

Aggarwal, C.C. (2015). *Data mining: the textbook*. Springer.

Aggarwal, C.C. and Han, J. (2014). *Frequent pattern mining*. Springer.

Aggarwal, C.C., Procopiuc, C., and Yu, P.S. (2002). Finding localized associations in market basket data. *IEEE Transactions on Knowledge and Data Engineering*, 14(1), 51–62.

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, 207–216. ACM.

Ahmed, K., Izadi, I., Chen, T., Joe, D., and Burton, T. (2013). Similarity analysis of industrial alarm flood data. *IEEE Transactions on Automation Science and Engineering*, 10(2), 452–457.

Alserhani, F.M. (2016). Alert correlation and aggregation techniques for reduction of security alerts and detection of multistage attack. *International Journal of Advanced Studies in Computers, Science and Engineering*, 5(2), 1.

Charbonnier, S., Bouchair, N., and Gayet, P. (2016). Fault template extraction to assist operators during industrial alarm floods. *Engineering Applications of Artificial Intelligence*, 50, 32–44.

Fjällström, P. (2016). A way to compare measures in association rule mining.

Hu, W., Chen, T., and Shah, S.L. (2018). Detection of frequent alarm patterns in industrial alarm floods using itemset mining methods. *IEEE Transactions on Industrial Electronics*, 65(9), 7290–7300.

Jakobson, G. and Weissman, M. (1995). Real-time telecommunication network management: extending event correlation with temporal constraints. In *International Symposium on Integrated Network Management*, 290–301. Springer.

Jin, X., Weiss, B.A., Siegel, D., and Lee, J. (2016). Present status and future growth of advanced maintenance technology and strategy in us manufacturing. *International journal of prognostics and health management*, 7(Spec Iss on Smart Manufacturing PHM).

Julisch, K. and Dacier, M. (2002). Mining intrusion detection alarms for actionable knowledge. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 366–375.

Kosters, W.A. and Pijls, W. (2003). Apriori, a depth first implementation. In *FIMI*, volume 3, 63. Citeseer.

Liang, R., Liu, F., Qu, J., and Zhang, Z. (2019). A bayesian-based self-diagnosis approach for alarm prognosis in communication networks. In *2019 8th International Symposium on Next Generation Electronics (ISNE)*, 1–3. IEEE.

Liang, Y., Zhang, Y., Sivasubramaniam, A., Jette, M., and Sahoo, R. (2006). Bluegene/l failure analysis and prediction models. In *Dependable Systems and Networks, 2006. DSN 2006. International Conference on*, 425–434. IEEE.

Liu, Y. and Zhu, L. (2019). A new intrusion detection and alarm correlation technology based on neural network. *EURASIP Journal on Wireless Communications and Networking*, 2019(1), 109.

Liu, Z. (2018). *Cyber-Physical System Augmented Prognostics and Health Management for Fleet-Based Systems*. Ph.D. thesis, University of Cincinnati.

O'Donovan, P., Bruton, K., and O'Sullivan, D.T. (2016). Case study: the implementation of a data-driven industrial analytics methodology and platform for smart manufacturing.

Quadrana, M., Cremonesi, P., and Jannach, D. (2018). Sequence-aware recommender systems. *arXiv preprint arXiv:1802.08452*.

Salah, S., Maciá-Fernández, G., and Díaz-Verdejo, J.E. (2013). A model-based survey of alert correlation techniques. *Computer Networks*, 57(5), 1289–1317.

Schwabacher, M. (2005). A survey of data-driven prognostics. In *Infotech@ Aerospace*, 7002.

Siraj, A. and Vaughn, R.B. (2005). Multi-level alert clustering for intrusion detection sensor data. In *NAFIPS 2005-2005 Annual Meeting of the North American Fuzzy Information Processing Society*, 748–753. IEEE.

Tan, P.N. (2018). *Introduction to data mining*. Pearson Education India.

Valdes, A. and Skinner, K. (2001). Probabilistic alert correlation. In *International Workshop on Recent Advances in Intrusion Detection*, 54–68. Springer.

Vogel-Heuser, B., Schütz, D., and Folmer, J. (2015). Criteria-based alarm flood pattern recognition using historical data from automated production systems (aps). *Mechatronics*, 31, 89–100.

Wang, J., Li, H., Huang, J., and Su, C. (2015). A data similarity based analysis to consequential alarms of industrial processes. *Journal of Loss Prevention in the Process Industries*, 35, 29–34.

Zan, X., Gao, F., Han, J., and Sun, Y. (2009). A hidden markov model based framework for tracking and predicting of attack intention. In *2009 International Conference on Multimedia Information Networking and Security*, volume 2, 498–501. IEEE.