Association Rules Extraction for the Identification of Functional Dependencies in Complex Technical Infrastructures

Federico Antonello[1], Piero Baraldi[1], Ahmed Shokry[1,5], Enrico Zio[1,2,3,] U. Gentile[4], L. Serio[4]

[1]*Energy Department, Politecnico di Milano, Via Lambruschini 4, 20156, Milan, Italy.*
E-mail: piero.baraldi@polimi.it

[2]*MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France.*
[3]*Eminent Scholar, Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Republic of Korea*
[4] CERN, 1211 Geneva 23, Switzerland
[5] Center for Applied Mathematics, Ecole Polytechnique, Route de Saclay, 91120 Palaiseau, France

Abstract:

This work proposes a method for identifying functional dependencies among components of complex technical infrastructures using databases of alarm messages. The developed method is based on the representation of the alarm database by a binary matrix, the use of the Apriori algorithm for mining association rules and a new algorithm for identifying groups of functionally dependent components. The effectiveness of the proposed method is shown by means of its application to an artificial case study and a real large-scale database of alarms generated by different supervision systems of the complex technical infrastructure of CERN (European Organization for Nuclear Research).

Notation

CTI Complex Technical Infrastructure

N_c number of CTI components

c_j generic *j-th* component

a_j^k alarm associated to the *k-th* malfunction of the *j-th* component

M_j^{al} number of types of alarm messages triggered by the *j-th* component

M^{al} total number of types of alarm messages

$A = \{a_j^k\}$ set of all possible alarm types

N^{al} total number of alarm messages collected in the database

$[t_0, t_f]$ time domain during which the N^{al} alarm messages of the database have been collected

Z number of time intervals in which the time domain $[t_0, t_f]$ is subdivided

Δt time interval length

$s_j^k(z)$ Boolean variable associated to the occurrence of the alarm a_j^k in the *z-th* time interval

$\vec{c}_j(z)$ vector of size M_j^{al} indicating the state of the *j-th* component in the *z-th* time interval

$\vec{\mathcal{T}}(z)$ vector of size M^{al} indicating the state of the CTI in the *z-th* time interval

T matrix of size $[\mathrm{Z} \times M^{al}]$ representing the evolution of the CTI state in the time domain $[t_0, t_f]$

X pattern of alarms

n(*X*) number of time intervals in which at least all the alarms of *X* occur

S(X) support of the pattern of alarms *X*

X^{fp} frequent pattern of alarms

s% minimum support

c% minimum confidence

$r_l = \{x_l^a \Rightarrow x_l^a\}$ generic association rule

$C(x_l^a \Rightarrow x_l^a)$ confidence of the *l-th* association rule

x^a antecedent of the *l-th* association rule

y^a consequent of the *l-th* association rule

$N^{\boldsymbol{p}}{}_{rule}$ number of $r_l, l = 1, .., N^{\boldsymbol{p}}{}_{rule}$ association rules

AR set of the obtained association rules

C_{fd}^d *d-th* group of functionally dependent components

n^{group} number of groups of functionally dependent components

λ_j^k transition rate of component *j* out of state *k*

1. INTRODUCTION

Complex Technical Infrastructures (CTI) are made by thousands of interconnected components which perform diverse functions, utilize technologies belonging to various domains (i.e., mechanics, hydraulics, electronics, information and communication technologies) and are organized in complex hierarchical architectures (Filip, 2008). CTIs are typically distributed over vast geographic areas and their systems are designed and built independently, and assembled considering only their direct physical interfaces. Furthermore, CTI systems are continuously modified with respect to their initial design, e.g. they grow in size, include new components, and old components are updated as a result of technology advancements, consolidations and/or operational needs, and they are operated, managed and maintained by different teams of operators, who autonomously control and develop the system and the processes. All these factors modify the physical interconnections among the systems and create functional dependencies among different sets of components. Therefore, due to the intricated dependencies among the components of the CTI and its evolutionary behaviour, it is impossible to model the CTIs using series and parallel structures (Billinton and Allan, 1992).

A local malfunction or perturbation may propagate into a CTI through groups of dependent components, originating unexpected cascades of failures across systems, which can lead to large-scale consequences and CTI unavailability. Vulnerability and resilience to failures of CTIs is an issue of great concern (Johansson and Hassel 2010; Eusgeld et. al. 2011; Kröger and Zio, 2011) strongly related to functional dependencies assessment and control (Zio, E. 2016). The identification of functional dependencies relying on classical methods of system decomposition and logical analysis cannot be applied since they require deep knowledge of the systems, such as their logic and structure function, which is not easy to retrieve for complex and evolving CTIs (Billinton and Allan, 1992).

In this context, the objective of the present work is the identification of functional dependencies among components of CTIs using the large amount of alarm messages, which are collected thanks to the recent advancement in the sensors, data acquisition, data storage and monitoring technologies.

Alarm sequences are currently used for purposes different from those of the present work, such as root causes analyses, CTI management and malfunctions and failures identification by plant personnel (Mika Klemettinen, Mannila, and Toivonen 1999; Amani, Fathi, and Dehghan 2005; Lozonavu, Vlachou-Konchylaki, and Huang 2016; Oborski 2014; Wang et al. 2017). However, the large amount of alarm messages collected in short periods of time, which is referred to as “alarms flood” (Karoly and Abonyi 2017; Dorgo and Abonyi 2018), makes the direct use of alarm messages unfeasible. For example, the operation center of a telecommunication network receives approximately 1 million alarms every day (Wang et al. 2017), whereas the CTI of the particle accelerator of CERN has produced more than 10 million alarms during 2016 (L. Serio et. al. 2018). Since most of these alarms do not require any intervention of the operators unless they occur within a well-defined critical sequence, it is fundamental to develop methods to process the alarm flood to identify the combination of alarms which are critical for the system availability and vulnerability. Alarm management systems have been developed for supporting plant personnel in their tasks (Sage and Cuppan 2001; Amani, Fathi, and Dehghan 2005; Lozonavu, Vlachou-Konchylaki, and Huang 2016; Wang et al. 2017). The information content of the available alarm message databases has been used for mining knowledge on the monitored system in (Filip 2008; Hatonen et al. 1996; M. Klemettinen, Mannila, and Toivonen 1999; Singh et al. 2011; Li and Li 2011). An alarm management system based on the filtering of the alarms and their hierarchical grouping has been proposed in (Zarri, 1991). Similarly, dependencies among data are discovered in (Priss, 2006; Wille, 2009) by transforming them into a lattice through the application of formal concept analysis. Notice that the approaches in (Zarri, 1991), (Priss, 2006; Wille, 2009) assume that the structure function of the system is known and they are not applicable to systems characterized by thousands of components such as a CTI.

In this context, the most promising methods are based on the identification of frequent patterns of alarms (Witten and Frank 2016), from which information on the correlation and conditional occurrences between groups of events is obtained in the form of "*if* (*condition*) *then* (*consequence*)" rules (Hu, et al., 1999). For example, an Association Rules Mining (ARM) technique has been applied to alarm messages collected from telecommunication networks, with the objective of identifying faults involving a large number of correlated components (Klemettinen, Mannila, and Toivonen, 1999). A pattern-growth algorithm has been applied in (Lozonavu, et al., 2016) for mining sequences of failure alarms causing telecommunication networks faults. The obtained results have been used for fault isolation and failure root cause investigation. Also, case-based reasoning has been used in (Amani, et al., 2005) for fault isolation in telecommunication network.

More recently, an Apriori-based algorithm has been developed for identifying alarm suppression rules (Karoly and Abonyi 2017) in the chemical industry. The method has been further extended to allow anticipating the alarm suppression by predicting the next alarms of the series (Dorgo and Abonyi 2018). The applicability of these two methods is restricted to laboratory pilot plants characterized by a relatively small number of possible alarms. The approaches proposed in (Amani, Fathi, and Dehghan 2005; M. Klemettinen, Mannila, and Toivonen 1999; Lozonavu, Vlachou-Konchylaki, and Huang 2016), based on frequent pattern and association rule mining techniques, are more suitable for the analysis of alarm messages generated by systems composed of thousands of components. All these studies aim at identifying temporal and/or spatial patterns of alarms for faults isolation and root cause analysis, without addressing the issue of identifying functional dependencies among components. Other issues that have not been yet addressed and limit the applicability of the proposed methods to CTIs are: *i*) the presence of various supervision systems collecting different types of alarm messages with different fields of information; *ii*) the possible desynchronization of the times at which the alarms are recorded by various supervision systems; *iii*) the repetition of the same alarm messages at small time intervals after its first occurrence.

This work proposes a method for the identification of functional dependencies among sets of components of a CTI, relying on a large database of alarm messages collected by various supervision systems. The method addresses the issues *i*), *ii*) and *iii*) above and involves three main steps:

1) Alarm database representation;
2) Extraction of association rules;
3) Identification of groups of functionally dependent components.

The main original contributions of the proposed work are:

a) the use of databases of alarm messages as a source of information for the identification of functional dependencies in complex systems;
b) the development of a method based on the Apriori algorithm for the extraction of association rules involving alarm messages;
c) the development of a novel algorithm for the identification of groups of functionally dependent components from the identified association rules.

The effectiveness of the proposed method is shown by means of its application to *i*) an artificial case study, which mimics the complexity of a real CTI and *ii*) a real large-scale database of alarms generated by different supervision systems of the CTI of CERN, where a particle accelerator composed by thousands of components is located along a 27 km circumference ring.

The remainder of the paper is organized as follows: Section 2 describes the problem setting and the concept of functional dependencies among components of a CTI. In Section 3, the proposed method is described. Section 4 introduces the case studies and discusses the obtained results. Finally, Section 5 draws some conclusions and recommends potential future lines of work.

2. Problem setting

The aim of the present work is to develop a method for the identification of functional dependencies in CTI by the analysis of large databases of alarm messages. The method is required to be robust with respect to delays between the time of occurrence of the malfunction and the recording of the affected physical measurement in the alarm database, and to possible occurrences of false alarms.

We consider a CTI composed by a large number of components, $N_c \gg 1$, and we assume to have available a database containing a large number of alarm messages, $N^{al} >> 1$, generated by a CTI during a long period of time $[t_0, t_f]$. The generic *i-th* alarm message is associated to the pair (t_i, m_i), defined by the time t_i at which the alarm occurs and a label m_i identifying the type of alarm. Assuming that there are M_j^{al} different types of alarms associated to the generic *j-th* component, c_j, we use the label a_j^k to refer to the *k-th* type of alarm message associated to component c_j.The set containing all the possible types of alarm messages in the database is:

$$A = \left\{a_1^1, \dots, a_1^{M_1^{al}}, \dots, a_j^1, \dots a_j^{M_j^{al}}, \dots, a_{N_c}^1, \dots, a_{N_c}^{M_{N_c}^{al}}\right\} \quad (1)$$

and the total number of alarm message types:

$$M^{al} = \sum_{j=1}^{N_c} M_j^{al} \quad (2)$$

Since the aim of the work is to identify functional dependencies that typically involve several CTI components, whose malfunction alone can be not critical, all CTI alarms are individually considered, without any apriori judgment about the criticality of the process they monitor. Also, approaches based on the grouping of the alarms considering their type (e.g., the group of the high pressure alarms) (Zarri, 1991) or the involved component (e.g., the group of the alarms involving a given compressor) (Zarri, 1991; Amani, et al., 2005) are not considered in this work, since a given type of alarm can be triggered by various components when the monitored quantity (e.g., the pression) exceeds a, possibly different, pre-set threshold and the same component can be involved in different functional dependencies with different types of malfunctions detected by different alarms.

According to (Estesami et al., 2016), two components of a system are functionally dependent if the operation of one is influenced by the operation of the other. Notice that this definition does not consider that there is an intrinsic causal unidirectional relationship among functional dependencies, e.g., a compressor requires the functioning of the electrical motor to operate, whereas the electrical motor does not require the functioning of the compressor to operate, and, therefore, malfunctions and failures triggered by functional dependencies are temporally ordered, e.g. the failure of the electrical motor occurs before the failure of the compressor. This causal relation, which is not represented by several methods for treating dependent failures, such as the beta factor and the binomial failure rate models (Mosleh, 1991; Zio, E. 2009; O'Connor and Mosleh, 2016), is not taken into account also in this work. For the identification of causal chains of malfunctions, post-processing of the identified groups of functionally dependent components is performed by operators and experts of the CTI or with algorithms developed ad-hoc. For example, in (Antonello et al., 2020) the temporal sequences of the alarms and the possible propagation delays between the events are used by an ad-hoc algorithm to automatically reconstruct the causal sequence of the alarms triggered by the functional dependencies.

Considering alarm messages triggered when components have abnormal behaviors or malfunctions, two generic components of a CTI, c_{j_1} and c_{j_2}, are assumed to be functionally dependent if a malfunction of component c_{j_1}, revealed by an alarm, causes a malfunction of components c_{j_2}, revealed by another alarm, or vicevesa. Figure 2 shows an example of alarm propagation in a case in which component c_2 is functionally dependent from component c_1. The malfunction of component c_1 causes the anomalous behavior of three physical quantities (q_1^1, q_1^2 and q_1^3), which exceed their predefined alarm thresholds and, therefore, produce the sequence of alarms a_1^1, a_1^2 and a_1^3. The functional dependency between components c_1 and c_2 determines the malfunction of component c_2 with the associated variation of the physical quantity q_2^2, which triggers alarm a_2^2 when it exceeds the corresponding threshold. Notice that different types of functional dependencies, such as cascades of malfunctions and common cause failures, satisfy this definition of functional dependency and can potentially be identified considering databases of alarms. Also, malfunctions propagations stopped by safety systems (e.g., UPS systems, which are electrical safety components providing emergency power in case of main failure to the electrical system) trigger alarm messages, of whose analysis can potentially reveal the corresponding functional dependencies.

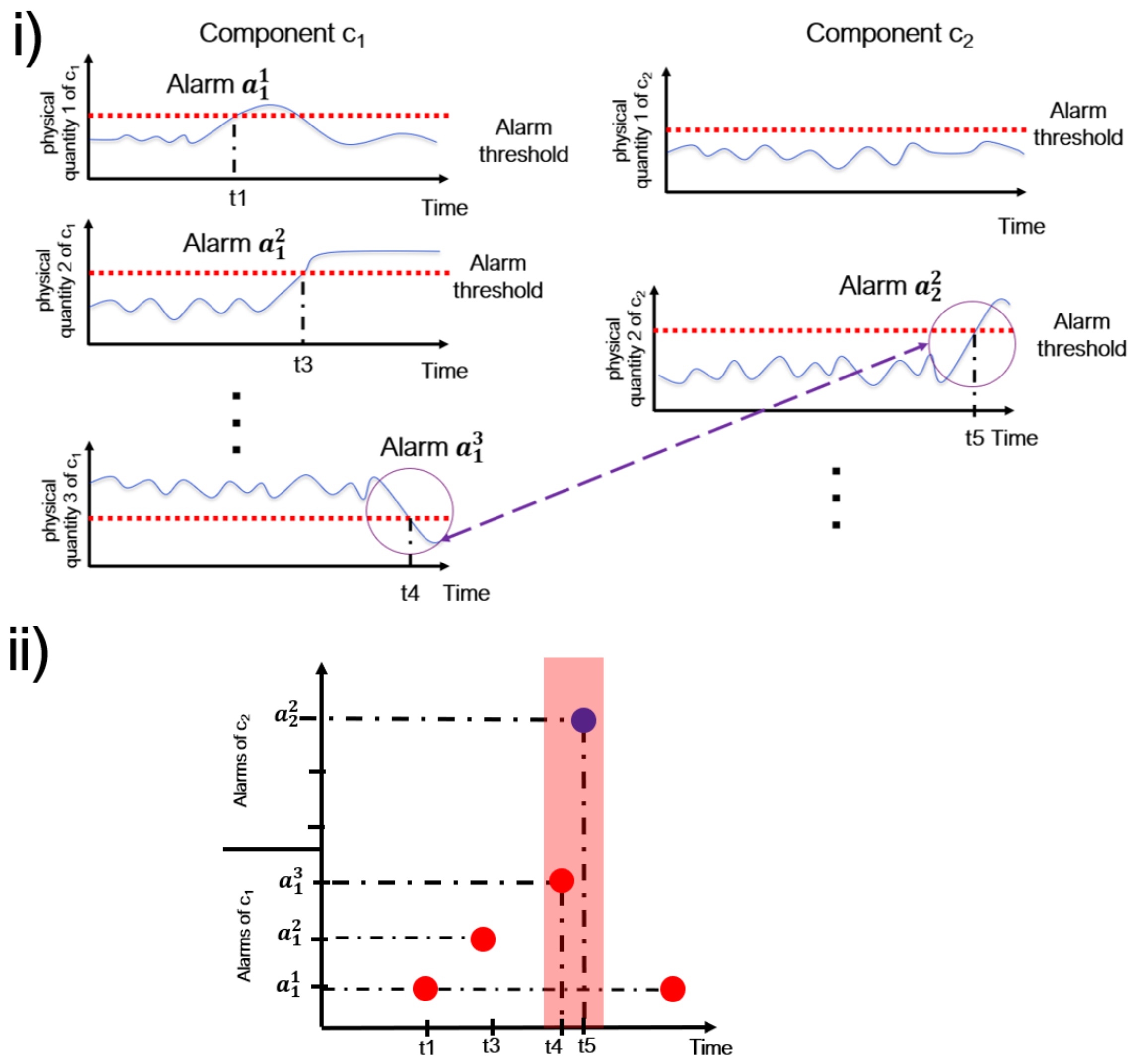


Figure 1. Example of alarm propagation in case of functional dependency between components c_1 and c_2: (i) signals behaviors; (ii) generated alarms.

The proposed method is based on:

1) The identification of association rules $r = \{x^a \Rightarrow y^a\}$ with x^a and $y^a \subset A$ and $x^a \cap y^a = \emptyset$ describing the conditional co-occurrence between the two subsets of disjoint alarm messages x^a and y^a, i.e., if the set of alarms x^a occurs, then the set of alarms y^a is also expected to occur.
2) The identification of groups of functionally dependent components, $C_{fd}^d = \{c_1^d, \dots, c_{N_d}^d\}$.

3. Method

The proposed method is based on the three phases of alarm database representation, association rules extraction and identification of disjoint groups of functionally dependent components, which will be described in Sections 4.1, 4.2 and 4.3, respectively. It resorts to the off-line analysis of large-scale databases of alarms messages, which are collected over long periods of operation (months/years) to guarantee that they include enough information about the CTI behaviour. Since it is expected that every few months of operation new databases of alarms containing useful information become available and new dependencies can emerge in the CTI, the application of the proposed method should be periodically repeated.

3.1. Alarm database representation

The time interval $[t_0, t_f]$ during which the N^{al} alarm messages of the database have been collected is subdivided into Z consecutive small time intervals of the same length $\Delta t = \frac{t_f - t_0}{Z}$. A Boolean variable, $s_j^k(z)$, is associated to the occurrence of the alarm of type a_j^k in the z-th time interval:

$$s_j^k(z) \in [0,1] \text{ with } \begin{cases} s_j^k(z) = 1 \text{ if alarm } a_j^k \text{ occurs at least once in } [t_0 + (z-1)\cdot\Delta t, t_0 + z\cdot\Delta t] \\ s_j^k(z) = 0 \text{ otherwise} \end{cases} \quad (3)$$

The state of the generic component c_j in the generic *z-th* time interval is represented by the Boolean vector:

$$\vec{c}_j(z) = \left[s_j^1(z), s_j^2(z), \dots, s_j^{N_j^{al}}(z)\right] \in [0,1]^{M_j^{al}} \quad (4)$$

and that of the CTI by the concatenation of the component state vectors $\vec{c}_j(z)$:

$$\vec{\mathcal{T}}(z) = \left[\vec{c}_1(z), \dots, \vec{c}_{N_c}(z)\right] \in [0,1]^{M^{al}} \quad (5)$$

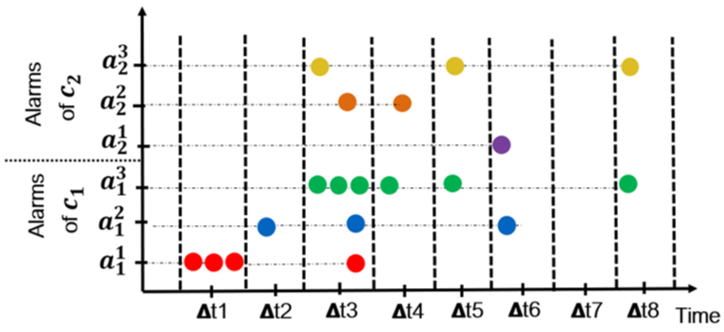
Since the discretization of the time in intervals of lengths Δt does not consider the sequence with which the alarms are recorded in the time interval, $\vec{\mathcal{T}}(z)$ is a robust representation of the CTI state with respect to possible desynchronizations among the alarms. Furthermore, the use of Boolean vectors allows handling possible problems of alarm repetition in the same time interval.

Finally, the database of alarms $(t_i, m_i), i = 1, \dots, N^{al}$, is transformed into the Boolean matrix:

$$T = \begin{bmatrix} \vec{\mathcal{T}}(1) \\ \dots \\ \vec{\mathcal{T}}(Z) \end{bmatrix} \in [0,1]^{Z \cdot \times M^{al}} \quad (6)$$

whose generic *z-th* row represents the state of the CTI during the *z-th* time interval. Therefore, *T* provides a dynamic representation of the CTI state evolution in the time interval $[t_0, t_f]$. The overall process of obtaining the Boolean matrix *T* from the alarm database is shown in Figure 2.

Notice that the time interval length Δt is a critical parameter for the database processing phase. Since the propagation of a malfunction from a component to another is not instantaneous, Δt should be large enough to guarantee that the alarm propagation occurs in a single time interval. Furthermore, the larger the Δt, the more efficient the handling of alarm desynchronization and repetition issues. On the other hand, the use of a too large Δt can unnecessary limit the information content of the dataset by reducing the number of extracted vectors $\vec{\mathcal{T}}(z)$ and can create correlations among independent alarms that occur far away from each other. A proper setting of Δt requires considering several aspects such as the frequency of occurrence of spurious alarms, the delays in the alarm acquisition, and the alarms acknowledgment, i.e., the set of actions made by the operators before dealing with the malfunction associated to the alarm (e.g. the alarm repetition can be stopped by the operators who alert the supervision system that they are taking responsibility for the related malfunction). Therefore, Δt should be large enough to guarantee that the cascade of alarms involved in the functional dependency is included in the same time interval, but, at the same time, small enough to guarantee that spurious alarms, which are not triggered by functional dependencies, do not frequently occur in the same time interval causing the identification of false functional dependencies.



Δt	Boolean Vector					
	s_1^1	s_1^2	s_1^3	s_2^1	s_2^2	s_2^3
Δt1	1	0	0	0	0	0
Δt2	0	1	0	0	0	0
Δt3	1	1	1	0	1	1
Δt4	0	0	1	0	1	0
Δt5	0	0	1	0	0	1
Δt6	0	1	0	1	0	0
Δt7	0	0	0	0	0	0
Δt8	0	0	1	0	0	1

Figure 2. Example of representation of an alarm database in a binary matrix. We consider a case with $N_c = 2$ components and M_1^{al}=$M_2^{al} = 3$ alarm types for component.

3.2. Association rules mining

Considering a set of alarms $X \subseteq A$, an association rule is a probabilistic logical expression of the form $x^{\boldsymbol{a}} \Rightarrow y^{\boldsymbol{a}}, x^{\boldsymbol{a}} \subset X,\ y^{\boldsymbol{a}} = X - x^{\boldsymbol{a}}$, representing the conditional co-occurrence of the two subsets, $x^{\boldsymbol{a}}$ and $y^{\boldsymbol{a}}$, of the set $X \subseteq A$, where $x^{\boldsymbol{a}}$ and $y^{\boldsymbol{a}}$ are referred to as "antecedent" and "consequent" of the rule, respectively (Hui et al. 2005; Srikant and Agrawal 1996).

We introduce the counter *n*(*X*) of the number of vectors $\vec{\mathcal{T}}(z)$ of the databases $T = \begin{bmatrix} \vec{\mathcal{T}}(1) \\ \dots \\ \vec{\mathcal{T}}(Z) \end{bmatrix}$ characterized by the occurrences of at least all the alarms of *X* (i.e., $\forall\ a_j^k \subset X,\ s_j^k(z) = 1$). The probabilistic logic expression $x^{\boldsymbol{a}} \Rightarrow y^{\boldsymbol{a}}$ is an association rule if the two following conditions are verified:

i) the *support* of *X* (Srikant and Agrawal 1996):

$$S(X) = S(x^{\boldsymbol{a}} \ \Rightarrow \ y^{\boldsymbol{a}}) = \frac{n(x^{\boldsymbol{a}} \cup y^{\boldsymbol{a}})}{Z} \tag{7}$$

is larger than a *minimum support* $s\%$. It means that all the alarms in the two subsets $x^{\boldsymbol{a}}$ and $y^{\boldsymbol{a}}$ occur in the same time interval Δt in at least $s\%$ of the *Z* time intervals.

ii) The *confidence* of the rule $x^{\boldsymbol{a}} \ \Rightarrow \ y^{\boldsymbol{a}}$:

$$C(x^{\boldsymbol{a}} \ \Rightarrow \ y^{\boldsymbol{a}}) = \frac{n(x^{\boldsymbol{a}} \cup y^{\boldsymbol{a}})}{n(x^{\boldsymbol{a}})} \tag{8}$$

is larger than a *minimum confidence* $c\%$. It means that the alarms in $y^{\boldsymbol{a}}$ occur in at least $c\%$ of the time intervals in which the alarms in $x^{\boldsymbol{a}}$ occur. The parameter $c\%$ represents the strength of conditionality between the two subsets, $x^{\boldsymbol{a}}, y^{\boldsymbol{a}}$, i.e. if the set of alarms in $x^{\boldsymbol{a}}$ occurs in a Δt, then the set of alarms in $y^{\boldsymbol{a}}$ is expected to be present in the same Δt with probability of at least $c\%$ (Singh et. al. 2011).

Association rules mining (ARM) algorithms have been originally developed in the market basket analysis area to analyze the customers purchase information, recorded at points-of-sales in the form of databases of transactions (Agrawal et. al. 1993; Srikant and Agrawal 1996; Zaki 2000; Witten and Frank 2016). The purpose is to identify hidden customers behaviours, i.e. frequently sold combination of items (e.g. {Milk, cheese, bread}, {Laptop, Laptop-bag, web-cam}, etc.), and the association among them in the form of probabilistic conditional relationship *"if-then"*, e.g. *if* {Milk and bread} are sold, *then* {bread} is sold. Nowadays, ARM techniques have been applied in various domains, such as telecommunication networks, medical and banking systems, web- and bio-mining, to extract knowledge from different sources of information, such as log events, maintenance reports and signal data (Fayyad et al. 1996; Jain et al. 2000; Hui et al. 2005; Witten and Frank 2016; Reder et. al. 2018; Bevilacqua and Ciarapica 2018).

The following paragraph reviews the applications of ARM techniques to fault detection, predictive maintenance and quality control. Martínez-de-Pisón, et al. (2012) proposed a method to analyse sequences of log events in industrial processes foe identifying sets of events (e.g., faults, malfunctions and maintenance actions) occurring simultaneously or within short periods of time. The method is shown able to improve the process efficiency and the product quality. Bastos, et al. (2012) applied an ensemble of data mining techniques, such as neural networks, decision trees and ARM algorithms, to maintenance data with the objective of extracting rules useful for root cause analyses and failure anticipation. Similarly, Kamsu-Foguem, et al. (2013) applied an ARM technique to maintenance data collected from a drill production process with the objective of extracting knowledge on operations and information management. Djatna, et al. (2015) applied an ARM technique to data collected in a wooden door manufacturing industry. The method investigates the relationships among operational settings and equipment efficiency. Antomarioni, et al. (2019) proposed a method based on ARM to investigate the relationship between maintenance operation and unexpected plant shutdowns. The method has been applied to an oil refinery plant and the extracted association rules are used to the anticipate possible occurrence of components breakdown during maintenance.

In this work, we treat the vector $\vec{\mathcal{T}}(z), z = 1, \dots, Z$, describing the CTI state in the *z-th* time interval as a transaction in the market basket analysis framework and the alarm $a_j^{\mathrm{k}}, j = 1, \dots N_c, k = 1, \dots, M_j^{al}$ as an item.

ARMs are typically based on the two steps of:

1) Identification of frequent patterns of alarms, $X^{fp} \subseteq A$ characterized by a support larger than $s\%$, i.e., $S(X^{fp}) > s\%$;
2) Extraction of association rules $x^{\boldsymbol{a}} \Rightarrow y^{\boldsymbol{a}}, x^{\boldsymbol{a}} \subset X^{fp},\ y^{\boldsymbol{a}} = X^{fp} - x^{\boldsymbol{a}}$ from the frequent patterns of alarms identified in 1). An association rule should satisfy the confidence condition $C(x^{\boldsymbol{a}} \Rightarrow y^{\boldsymbol{a}}) > c\%$.

With respect to 1), a wide spectrum of frequent pattern mining algorithms, such as Apriori (Srikant and Agrawal 1996), Frequent Pattern Growth Tree (FP-GT) (Han, Pei, and Yan 2000), Hash-based algorithm (Park, J. S. et. al. 1995) and H-Mine (Pei, J. et. al. 2007), have been developed. Among them, the Apriori algorithm has been considered in this work since it has shown to require smaller computational effort than the other algorithms, when dealing with large-scale datasets made by thousands of items (Witten and Frank 2016) (Mannila, Toivonen, and Verkamo 1994; Oswaldo and Monroy 2006).

Apriori mines the frequent patterns performing a level-wise iterative scanning of the database to identify those alarm sets $X \subseteq A$ with a support larger than $s\%$. To reduce the enormous number of candidate frequent patterns, the Apriori search exploits the "anti-monotone property of the support", which states that the support of a set of alarms never exceeds the support of its subsets:

$$\forall\, X, X' \colon (X \subset X') \Rightarrow S(X) \geq S(X') \qquad (9)$$

Therefore, if a set of alarms *X* has a support lower than the minimum support, $S(X) < s\%$, than, all of its supersets, $X' \supset X$, have also a support lower than $s\%$, $S(X') < s\%$.

Once the frequent patterns of alarms X^{fp} have been identified, all the possible subdivisions of each frequent pattern X^{fp} into non-empty pairs of subsets $(x^{\boldsymbol{a}}, y^{\boldsymbol{a}})$, such that $x^{\boldsymbol{a}} \subset X^{fp}$, $y^{\boldsymbol{a}} = X^{fp} - x^{a}$ and:

$$C(x^{\boldsymbol{a}} \Rightarrow y^{\boldsymbol{a}}) = \frac{n(X^{fp})}{n(x^{\boldsymbol{a}})} > c\%$$

are identified (Srikant and Agrawal 1996).

Although the identification of association rules through the Apriori algorithm is based on the two measures of support and confidence, the effectiveness of the generated rules is typically evaluated considering also the *lift* measure, which quantifies the mutual dependency among the alarms in the rule antecedent and consequent parts:

$$Lift(x^{\boldsymbol{a}} \Rightarrow y^{\boldsymbol{a}}) = \frac{S(x^{\boldsymbol{a}} \cup y^{\boldsymbol{a}})}{S(x^{\boldsymbol{a}}) * S(y^{\boldsymbol{a}})} \qquad (9)$$

Larger the lift, stronger the mutual dependency among $x^{\boldsymbol{a}}$ and y^{a}. In particular, values of lift equal to (lower than) 1, indicate that the rule antecedent and consequent parts are (negatively correlated) uncorrelated.

Since several frequent patterns are typically identified in large-scale alarm databases and a single frequent pattern formed by *k* alarms can potentially generate $2^{k} - 2$ rules, thousands of association rules can be generated (Zaki 2000; Pasquier et al. 2005). Therefore, we apply a postprocessing phase whose objective is to eliminate redundant rules without reducing the overall information content to facilitate the analysis of the results. According to (Pasquier et al. 2005), a generic rule $r_l: \{x^{\boldsymbol{a}}{}_l \Longrightarrow y^{\boldsymbol{a}}{}_l\}$ is pruned if there is another rule $r_m: \{x^{\boldsymbol{a}}{}_m \Longrightarrow y^{\boldsymbol{a}}{}_m\}$, characterized by the same values of support and confidence, i.e. $s(r_m) = s(r_l)$, $c(r_l) = c(r_m)$, that involves all the components of r_l, i.e. $(x^{\boldsymbol{a}}{}_l \cup y^{\boldsymbol{a}}{}_l) \subseteq (x^{\boldsymbol{a}}{}_m \cup y^{\boldsymbol{a}}{}_m)$, and whose antecedent $x^{\boldsymbol{a}}{}_m$ is a subset of (or equal to) the antecedent $x^{\boldsymbol{a}}{}_l$ of r_l, i.e. $x^{\boldsymbol{a}}{}_m \subseteq x^{\boldsymbol{a}}{}_l$. It is important to notice that all the alarms involved in the initial list of rules are still involved in the pruned list and all pruned rules can be retrieved from the non-redundant rules.

The final set of the $N^{\boldsymbol{p}}{}_{rule}$ non-redundant rules obtained by applying the pruning procedure will be referred to as $AR = \{x_l^a \Rightarrow y_l^a,\ \ l = 1, \dots, N^{\boldsymbol{p}}{}_{rule}\}$, with $N^{\boldsymbol{p}}{}_{rule}$ indicating the obtained number of rules

3.3. Identification of groups of functionally dependent components

The objective of this module is the identification of n^{group} groups of functionally dependent components. The generic *d-th* group $C_f^d = \{c_1^d, \dots, c_{N_d}^d\}$ is formed by N_d components functionally dependent with at least another component of the same group, i.e., $\forall\ c_{j'}^d \in C_f^d$, it exists another component $c_{j''}^d \in C_f^d$ such that there is a pair of alarms $(a_{j'}^{k'}, a_{j''}^{k''})$ generated by the two components $c_{j'}^d$ and $c_{j''}^d$, which belong to the same rule

$\{x_l^a \Rightarrow y_l^a\} \in AR$.

In this work, for simplicity, we assume that a component of the CTI can have only one type of malfunction which is caused by or causes a malfunction of another component. Therefore, the groups of functionally dependent components are disjoint, i.e. $C_f^{d_1} \cap C_f^{d_2} = \emptyset$ for any $d_1 \neq d_2 = 1, \dots, n^{group}$. Notice that the case in which a component c_j has two or more types of malfunctions, which are caused by or cause malfunctions, in different components, can be treated by fictitiously duplicating the component.

To identify the groups of functionally dependent components, the association rules $r_l = \{x_l^a \Rightarrow y_l^a\}$, $l = 1, \dots, N^{\boldsymbol{p}}{}_{rule}$, involving the alarms are transformed into the association rules $r_l^c = \{x_l^c \Rightarrow y_l^c\}$ involving the corresponding components, by transforming the generic alarm a_j^k into the component c_j. For example, the association rule $\{a_3^4, a_{35}^{20} \Rightarrow a_{25}^{31}, a_{15}^{26}\}$ involving the alarm is transformed into the rule $\{c_3, c_{35} \Rightarrow c_{25}, c_{15}\}$ involving the components. Then, the subsets $H_l^c = (x_l^c \cup y_l^c)$, $l = 1, \dots, N^{\boldsymbol{p}}{}_{rule}$, formed by the union of the components in the antecedent and consequent of the rule, are identified and the sets C_f^d are found by merging the sets H_l^c with non-empty intersections, according to the algorithm of Figure 3.

```
INPUT: H_1^c, ..., H^c_{N^p_rule}
C_f^1 = {H_1^c}
N^group = 1
  For l=2, ..., N^p_rule
        Group_l= ∅ % sets of groups containing at least one component of H_l^c
        For d=1, ..., N^group
              If H_l^c ∩ C_f^d ≠ ∅
                    Group_l=[Group_l, d]
              End
        End
        If Group_l=[∅]:
              N^group = N^group + 1
              C_f^{N^group} = {H_l^c}
        Else
              C_f^{Group_l(1)} = C_f^{Group_l(1)} ∪ H_l^c
              If size(Group_l)>1:
                    For d*=2, ..., size(Group_l):
                          C_f^{Group_l(1)} = C_f^{Group_l(1)} ∪ C_f^{Group_l(d*)}
                          C_f^{Group_l(d*)} = ∅
                    End
              End
        End
  End
OUTPUT: C_f^1,...., C_f^{N^group}
```

Figure 3. Algorithm developed for the identification of groups of functionally dependent components.

The obtained set of functionally dependent components can be represented by employing a graph-oriented visualization technique, where each vertex represents a component and the edges connect all the pairs of components in the same rule. Figure 4 shows an example of graphical visualization of groups of functionally dependent components.

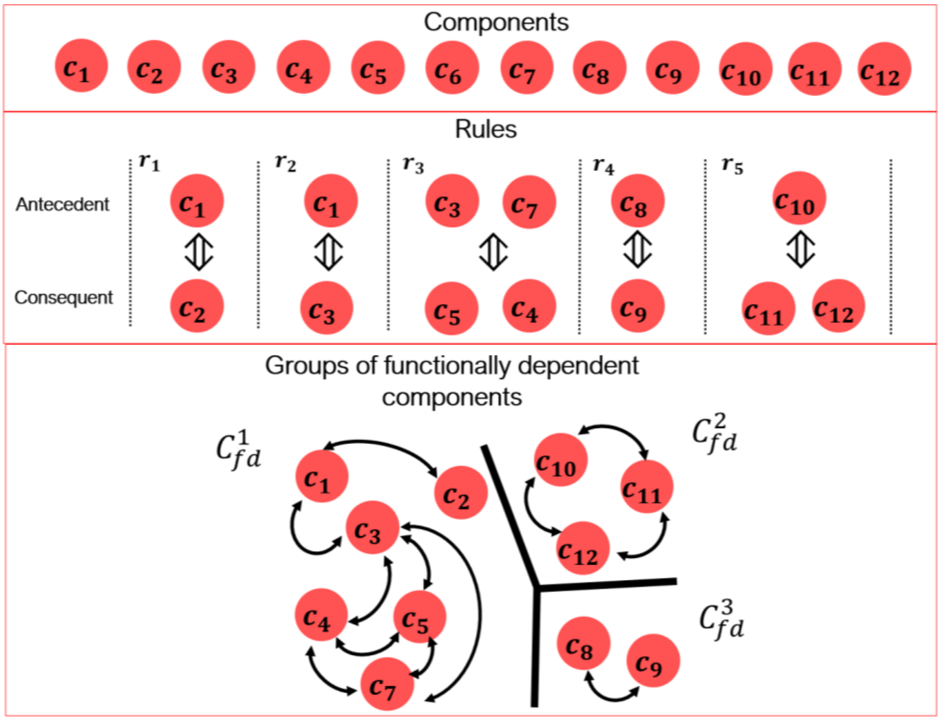


Figure 4. Examples of visualization of groups of functionally dependent components: list of components (upper), extracted rules (middle), groups of functionally dependent components (bottom).

4. Case studies

The proposed method is applied to a synthetic alarm database, which mimics occurrences of malfunctions in a CTI for which the true functional dependencies are known, and to a real large-scale alarm database collected at the CERN's CTI during 2016. The first application shows that the functional dependencies are correctly identified using as input only the database of alarms, without any information on the system structure function. The second application, which considers a real CTI for which the functional dependencies are not apriori known, investigates the potentiality of the method for the analysis of a real CTI and its robustness against spurious alarms and propagation delays.

4.1. Application to a synthetic alarm database

We consider a CTI formed by N_c=300 components. We assume that each CTI component, c_j, $j = 1, \dots, N_c$, can be in the healthy, $k = 1$, partially degraded, $k = 2$ and very degraded, $k = 3$, state and performs transitions among the states at random times. Figure 5 shows the possible stochastic state transitions representing component degradation (from state 1 to state 2 and from state 2 to state 3) and repair (from state 3 to state 1). Table 1 reports the transition rates, λ_j^k, of component, $j = 1, \dots, N_c$, from the state k to the state $k+1$ if k=1,2 and from state 3 to state 1 if $k=3$. We assume that the transition rates are constant and that the alarm a_j^k is triggered each time component c_j performs a state transition out of state k.

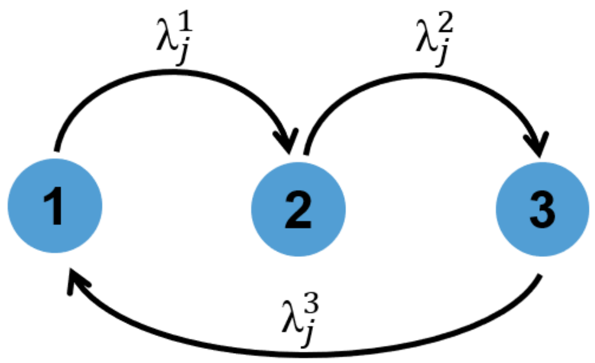


Figure 5. State transitions of a CTI component.

Table 1. Transition rates in day^{-1}.

Component c_j	Transition rates		
$j = 1, \dots, 100$	λ_j^1 =0.05	λ_j^2 =0.1	λ_j^3 =0.02
$j = 101, \dots, 200$	λ_j^1 =0.15	λ_j^2 =0.2	λ_j^3 =0.02
$j = 201, \dots, 300$	λ_j^1 =0.015	λ_j^2 =0.1	λ_j^3 =0.02

We further model twelve different functional dependencies among CTI components, which are divided into the three following types:

1) 10 functional dependencies between pairs of components. They are originated by the transition from state 2 ('degraded') to state 3 ('very degraded') of a component $c_{j'}$ with $j' \leq 100$, which can cause the transition from state 2 to state 3 of a component $c_{j''}$ with 100< $j'' \leq 200$. The probability of the malfunction propagation, i.e., the probability that the transition of component $c_{j'}$ from state 2 to state 3 causes the same state transition of component $c_{j''}$, is set to 0.9. The time necessary for the

propagation of the malfunction is randomly sampled from a uniform continuous distribution in the range [1-30] minutes. Table 2 reports the list of functional dependencies of category 1.

Table 2. Functional dependencies of category 1.

Components involved	**Functional Dependency**
c_1, c_{101}	$a_1^2 \rightarrow a_{101}^2$
c_2, c_{102}	$a_2^2 \rightarrow a_{102}^2$
c_3, c_{103}	$a_3^2 \rightarrow a_{103}^2$
c_4, c_{104}	$a_4^2 \rightarrow a_{104}^2$
c_5, c_{105}	$a_5^2 \rightarrow a_{105}^2$
c_6, c_{106}	$a_6^2 \rightarrow a_{106}^2$
c_7, c_{107}	$a_7^2 \rightarrow a_{107}^2$
c_8, c_{108}	$a_8^2 \rightarrow a_{108}^2$
c_9, c_{109}	$a_9^2 \rightarrow a_{109}^2$
c_1, c_{110}	$a_{10}^2 \rightarrow a_{110}^2$

2) A functional dependency involving four components. It originates when component c_{111} performs a transition from state 2 to state 3, which can cause an ordered sequence of events leading to the transitions of components c_{112}, c_{211} and c_{212} from state 2 to state 3 (Table 3). The probability of malfunction propagation between two components of the sequence is set to 0.9 and the time necessary for the malfunction propagation is randomly sampled from a uniform continuous distribution in the interval [1-5] minutes.

Table 3. Functional dependency of category 2.

Components involved	**Functional Dependency**
c_{111}, c_{112}, c_{211}, c_{212}	$a_{111}^2 \rightarrow a_{112}^2 \rightarrow a_{211}^2 \rightarrow a_{212}^2$

3) A functional dependency involving six components. It originates when component c_{11} performs a transition from state 2 to state 3 which can cause an ordered sequence of events leading to the transitions of components c_{12}, c_{113} c_{114}, $c_{213}$and c_{214} from state 2 to state 3 (Table 4). The probability of malfunction propagation between any two components of the sequence is set to 0.9 and the time necessary for the malfunction propagation is randomly generated from a uniform distribution in the interval [1-20] minutes.

Table 4. Details of the functional dependency of category 3.

Components involved	**Functional Dependency**
c_{11}, c_{12}, c_{113}, c_{114}, c_{213}, c_{214}	$a_{11}^2 \rightarrow a_{12}^2 \rightarrow a_{113}^2 \rightarrow a_{114}^2 \rightarrow a_{213}^2 \rightarrow a_{214}^2$

The CTI behaviour is simulated by sampling for each component of the system the sequence of times $[t_1, \dots \dots .., t_{i-1}, t_i, t_{i+1}, \dots]$ at which it performs state transitions. The length of the time interval between two consecutive transitions, $t_i - t_{i-1}$ is randomly sampled from the corresponding exponential distribution, with the constant failure rate values reported in Table 1. Once the *i*-th transition is performed, the corresponding alarm is generated at time $t_i + \nu_i$, where ν_i is a truncated gaussian noise with positive values and standard deviation of 5 minutes, which represents possible delays or desynchronizations in the alarm supervision system. The repetition of the alarm is simulated by sampling the number of times, r_i, that the same alarm is triggered from a uniform discrete probability distribution in the range [1,10]. The time intervals, δt_1^i, δt_2^i..., $\delta t_{r_i}^i$ between two consecutive occurrences of the *i*-th alarm is sampled from a uniform continuous distribution in the range [1,15] minutes.

Since there is the possibility that the failure propagation induced by a functional dependency be stopped, for example by the intervention of an operator or a safety system, the probability of malfunction propagation between any two components of the sequence is set to 0.9. This is in accordance with the modelling of cascading failures, where the failure of a component is typically assumed to cause an increase of the probability of failure of other components, which, for example, share common loads or functionalities with it (Zio. E. 2009; David A. E., et. al., 2020). Furthermore, the use of a probability of propagation lower than 1 allows simulating the occurrence of spurious alarms, i.e. alarms triggered without the presence of an actual malfunction and that are not causing cascading failures.

The CTI behaviour is simulated for a period of time $[t_0, t_f]$=[0, 720 days] obtaining N^{al}=71807 couples (t_i, a_j^k) that include the alarm message a_j^k and its occurrence time t_i. Then, the entire time domain of the analysis (720 day) is discretized considering *Z*=17280 time intervals of length $\Delta t = 60$ min and the Boolean matrix, *T*, representing the CTI state in each time interval is computed by applying the procedure of Section 3.1. Notice that the matrix *T* does not distinguish whether an alarm occurs once or several times in the same time interval, i.e. the generic element of the matrix corresponding to the *z*-th time interval and the a_j^k alarm type will be 1, independently on when the alarm of type a_j^k has occurred and how many times it has been repeated within the period of time $[t_0 + (z-1) \cdot \Delta t, t_0 + z \cdot \Delta t]$. This modelling choice allows efficiently managing the fact that the alarm of type a_j^k can be repeated and that there is not any synchronization among the beginning of the time interval, which is prefixed, and the occurrence of the alarms, which is random.

With respect to the setting of the Apriori parameters *minimum support,* $s\%$, and minimum confidence, *c*%, a small value of *s%*=0.0005 is used since the occurrence of alarms caused by functional dependencies is rare. On the other side, *c%* is set to a relatively large value to counterbalance the fact that small *s%* values can lead to the identification of frequent patterns formed by alarms occurring in the same time intervals by chance.

The application of the method leads to the identification of ${N^{p}}_{rule}$ =235 non-redundant association rules in a computational time of 63 seconds on an Intel core (TM) i7-4790 CPU@ 3.6 GHz, 16 GB RAM.

Table 5 reports the rules involving the alarms associated to each functional dependency, which provide a complete description of all the 12 simulated functional dependencies. Notice that all the functional dependencies of type 1 are described by two rules containing alarms triggered by the two functionally dependent components $c_{j'}$ and $c_{j''}$: $\left\{a_{j'}^2 \implies a_{j''}^2\right\}$ and $\left\{a_{j''}^2 \implies a_{j'}^2\right\}$. This is consistent with the fact that an association rule $r = \{x^a \Rightarrow y^a\}$ is a logical probabilistic expression and the rule direction, ⇒ , does not imply causality among the components in the antecedent and consequent part of the rule. Similarly, the functional dependencies of types 2 and 3 are described by rules, whose antecedent and consequent are combinations of the alarms triggered by the malfunctions involved in the functional dependency. In these latter cases, some of the identified rules do not involve all the dependent components but only a subset of them. It is also worth underlining that all the generated rules are characterized by large values of lift (i.e., lift>>1), which ensures the mutual dependency among the rule antecedent and consequent parts.

The robustness of the obtained results with respect to *s%* and *c%* parameters setting is discussed in Figures 6 and 7. Figure 7 shows that *s%*=0.0005 allows identifying a remarkable larger number of frequent patterns than those found with a *s%*>0.0007, whereas a further reduction of *s%* does not significantly increase the number of frequent patterns. Figure 8 shows the number of non-redundant rules generated considering different values of *c%*, when *s%* is set equal to 0.0005. Notice that *c%*=0.6 allows obtaining a compromise solution among the objectives of obtaining a set of reliable rules (i.e., rules with large confidence) and a complete set of rules, which includes all the CTI functional dependencies.

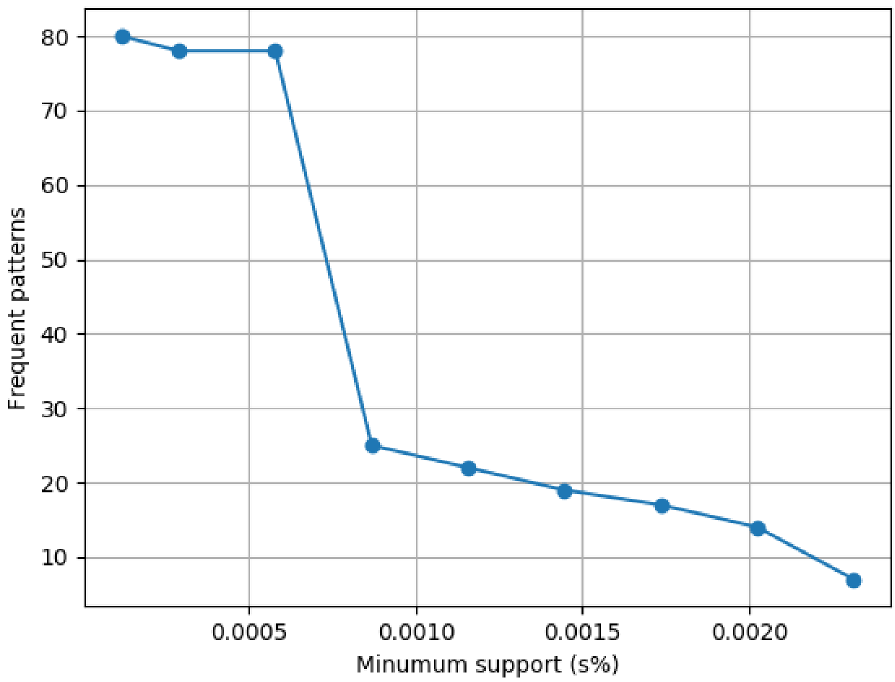


Figure 6. Number of frequent patterns identified considering different values of s% when the confidence is set equal to 0.6.

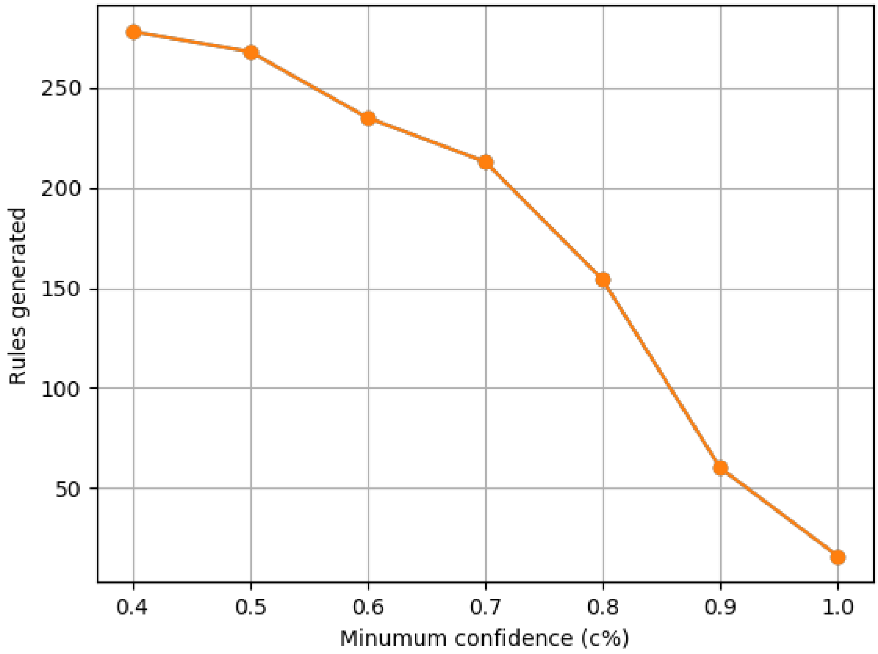


Figure 7. Number of rules generated considering different values of c% when s% is set equal to 0.0005.

Table 5. Rules extracted by setting s% =0.0005 and c%=0.6 and corresponding threshold, confidence and lift.

Simulated Functional Dependency	**Extracted Rules**				
	Antecedent **{alarm Identifier}** ⇒	Consequent **{alarm Identifier}**	Support	Confidence	Lift
$a_1^2 \rightarrow a_{101}^2$	$\{a_{101}^2\} \Rightarrow$	$\{a_1^2\}$	0.00139	0.90	568
	$\{a_1^2\} \Rightarrow$	$\{a_{101}^2\}$	0.00139	0.855	568
$a_2^2 \rightarrow a_{102}^2$	$\{a_{102}^2\} \Rightarrow$	$\{a_2^2\}$	0.00130	0.86	560
	$\{a_2^2\} \Rightarrow$	$\{a_{102}^2\}$	0.00130	0.81	560
$a_3^2 \rightarrow a_{103}^2$	$\{a_{103}^2\} \Rightarrow$	$\{a_3^2\}$	0.00118	0.85	670
	$\{a_3^2\} \Rightarrow$	$\{a_{103}^2\}$	0.00118	0.93	670
$a_4^2 \rightarrow a_{104}^2$	$\{a_{104}^2\} \Rightarrow$	$\{a_4^2\}$	0.00115	0.86	682
	$\{a_4^2\} \Rightarrow$	$\{a_{104}^2\}$	0.00115	0.91	682
$a_5^2 \rightarrow a_{105}^2$	$\{a_{105}^2\} \Rightarrow$	$\{a_5^2\}$	0.00109	0.90	636
	$\{a_5^2\} \Rightarrow$	$\{a_{105}^2\}$	0.00109	0.75	636
$a_6^2 \rightarrow a_{106}^2$	$\{a_{106}^2\} \Rightarrow$	$\{a_6^2\}$	0.00101	0.80	590
	$\{a_6^2\} \Rightarrow$	$\{a_{106}^2\}$	0.00101	0.701	590
$a_7^2 \rightarrow a_{107}^2$	$\{a_{107}^2\} \Rightarrow$	$\{a_7^2\}$	0.000983	0.81	756
	$\{a_7^2\} \Rightarrow$	$\{a_{107}^2\}$	0.000983	0.92	756
$a_8^2 \rightarrow a_{108}^2$	$\{a_{108}^2\} \Rightarrow$	$\{a_8^2\}$	0.000954	0.75	682
	$\{a_8^2\} \Rightarrow$	$\{a_{108}^2\}$	0.000954	0.87	682
$a_9^2 \rightarrow a_{109}^2$	$\{a_{109}^2\} \Rightarrow$	$\{a_9^2\}$	0.000925	0.74	750
	$\{a_9^2\} \Rightarrow$	$\{a_{109}^2\}$	0.000925	0.91	750
$a_{10}^2 \rightarrow a_{110}^2$	$\{a_{110}^2\} \Rightarrow$	$\{a_{10}^2\}$	0.000549	0.63	994
	$\{a_{10}^2\} \Rightarrow$	$\{a_{110}^2\}$	0.000549	0.86	994

$a^2_{111} \to a^2_{112} \to a^2_{211} \to a^2_{212}$	$\begin{Bmatrix} a^2_{111} \\ a^2_{112} \end{Bmatrix} \Rightarrow$	$\begin{Bmatrix} a^2_{211} \\ a^2_{212} \end{Bmatrix}$	0.00127	*1*	734
	$\{a^2_{112}\} \Rightarrow$	$\begin{Bmatrix} a^2_{211} \\ a^2_{212} \end{Bmatrix}$	0.00130	*1*	652
	…				
	$\{a^2_{112}\} \Rightarrow$	$\{a^2_{212}\}$	0.00133	*0.6*	474
$a^2_{11} \to a^2_{12} \to a^2_{113} \to a^2_{114} \to a^2_{213} \to a^2_{214}$	$\begin{Bmatrix} a^2_{213} \\ a^2_{214} \end{Bmatrix} \Rightarrow$	$\begin{Bmatrix} a^2_{11} \\ a^2_{12} \\ a^2_{113} \\ a^2_{114} \end{Bmatrix}$	0.000636	*1*	1234
	$\begin{Bmatrix} a^2_{11} \\ a^2_{113} \\ a^2_{114} \end{Bmatrix} \Rightarrow$	$\begin{Bmatrix} a^2_{213} \\ a^2_{214} \end{Bmatrix}$	0.000752	*0.89*	1186
	….				
	$\{a^2_{213}\} \Rightarrow$	$\begin{Bmatrix} a^2_{214} \\ a^2_{12} \end{Bmatrix}$	0.000752	0.76	608

The n^{fd}=12 groups of functionally dependent components reported in Table 6 have been identified by applying the algorithm of Section 3.3 to the obtained set of non-redundant association rules. Notice that they completely correspond to the simulated functional dependencies.

Table 6. Identified groups of functionally dependent components

Set	**Components**
1	C_1, C_{101}
2	C_2, C_{102}
3	C_3, C_{103}
4	C_4, C_{104}
5	C_5, C_{105}
6	C_6, C_{106}
7	C_7, C_{107}
8	C_8, C_{108}
9	C_9, C_{109}
10	C_{10}, C_{110}
11	C_{111}, C_{112}, C_{211}, C_{212}
12	C_{11}, C_{12}, C_{113}, C_{114}, C_{213}, C_{214}

Figures 8 shows the number of groups of functionally dependent components identified using different values of the *minimum support*, $s\%$, when the *minimum confidence*, $c\%$, is set equal to 0.6. As expected, the smaller the values of *s%,* the larger the number of groups of functionally dependent components. Similarly, to the considerations made with respect to the number of rules, it appears that small values of minimum support (s%<0.0005) allow identifying all the functional dependencies.

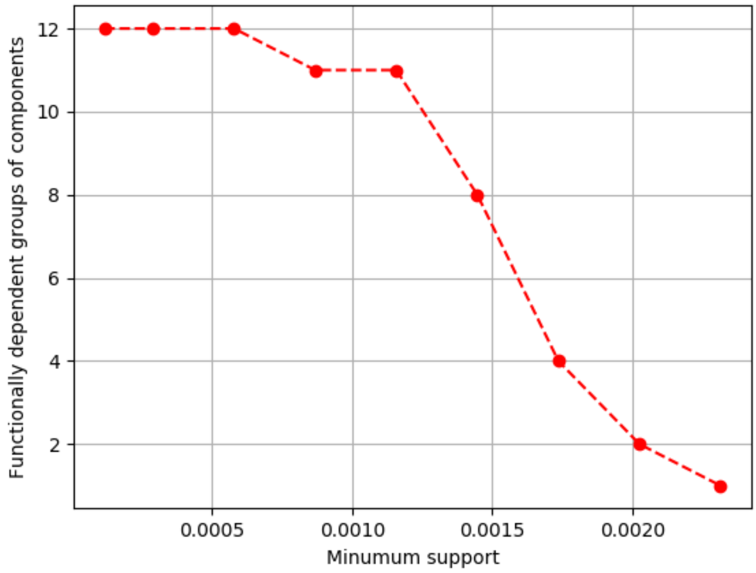


Figure 8. Number of groups of functionally dependent components identified considering different values of s% with c% set to 0.6.

Notice that a further reduction of the *minimum support* threshold*, s%,* increases the probability of identifying spurious rules, i.e. rules including alarms that do not belong to a real functional dependency. Table 7 reports some examples of spurious rules extracted by considering s%, and c%, equal to 0.00005 and 0.01, respectively. *Rules 1* and *2,* which are spurious rules since they involve two alarms occurring by chance in a small portion of the same time intervals, have associated remarkably smaller values of *support*, *confidence* and *lift* in comparison to the rules in Table 5, which include only alarms involved in real functional dependencies. This result shows that spurious rules can be excluded by applying larger confidence thresholds. It is also interesting to consider *Rule 3*, which includes two alarms that are involved in a functional dependency (a^2_{101}, a^2_1) and another alarm (a^1_{252}) that occurs with the other two just by chance in a small portion of the same time intervals. This is confirmed by the fact that the rule including only a^2_{101} and a^2_1 is characterized by a support of 1.39e-03, whereas the rule including a^2_{101}, a^2_1 and a^1_{252} has a smaller support of 8.68e-05 and is characterized by *confidence* and *lift* values of the same order of magnitude of the rules in Table 6. This is because, given the dependency among alarms a^2_{101} and a^2_1, in the time intervals in which a^2_{101} occur by chance with a^1_{252}, the alarm a^2_1 is always present. Therefore, this rule can be excluded only considering a large value of minimum support.

Table 7. Examples of spurious rules extracted by setting the Apriori parameters, s%, and c%, equal to 0.00005 and 0.01, respectively

Spurious Rules

	Antecedent **{alarm Identifier}** ⇒	Consequent **{alarm Identifier}**	Support	Confidence	Lift
Rule 1	$\{a_{141}^{1}\} \Rightarrow$	$\{a_{61}^{1}\}$	0.000115	0.10	82
Rule 2	$\{a_{39}^{1}\} \Rightarrow$	$\{a_{77}^{1}\}$	0.0000868	0.16	74
Rule 3	$\{a_{101}^{2}, a_{252}^{1}\} \Rightarrow$	$\{a_{1}^{2}\}$	0.0000868	1.0	1152
Rule 4	$\{a_{1}^{2}\} \Rightarrow$	$\{a_{101}^{2}, a_{252}^{1}\}$	0.0000868	0.1	1152

4.2. CERN complex technical infrastructure

The CTI of CERN, which is the largest existing particle accelerator, is composed by several systems working together for the functioning of the Large Hadron Collider (LHC) (J. Nielsen and L. Serio 2016). It consists of a 27 Km ring of superconducting magnets and infrastructures, extending over the Swiss and French borders and located about 100 m underground. Since in 2016 the CTI component faults contributed to more than 1/5 of the overall faults downtime (B. Todd et al, 2016), the identification of unknown functional dependencies among the CTI components is of paramount importance for improving the accelerators performance and availability for physics. The availability of advanced sensor networks and monitoring facilities allows the real-time collection of information about the operational states of the various CTI assets and the generation of alarm messages when key physical quantities exceed prefixed thresholds.

In this Section, we consider the alarms databases generated during the period $[t_0, t_f]$=[January 1st, 2016; December 31st, 2016] by three supervision systems of the *LHC point 8*, which is a part of the infrastructure representative of the overall CTI complexity. The alarms are generated by the cryogenic, the electric and the cooling and ventilation systems. N_{al} =253591 alarms reporting M^{al}=6800 different types of malfunctions caused by $N_c = 2895$ components have been collected during the considered period.

Based on CERN operators and expert knowledge about the malfunction propagation among the components and systems, the time interval length is set to $\Delta t = 30$ min. Therefore, the one-year period [January 1st, 2016, December 31st, 2016] has been divided into $Z = 17500$ time intervals. Setting the minimum support, $s\%$, equal to 0.001 and the minimum confidence, $c\%$, equal to 0.8, the computational time required to apply the proposed method on an Intel core (TM) i7-4790 CPU@ 3.6 GHz, 16 GB RAM) has been of 3*63* seconds. The use of $s\%$ values smaller than 0.02, which is expected to allow identifying more association rules, and, consequently, more functional dependencies, is not feasible from a computational point of view, given the large number of alarms (N_{al} =253591) and alarms types (M^{al}=6800) to be processed by the Apriori algorithm. Notice that the fraction of spurious rules identified by the method cannot be evaluated since the real dependencies of the CTI are not known. It can, however, be highlighted that the analysis in Section 4.1 has shown that the larger the value of parameters, $s\%$ and $c\%$, the lower the risk of identifying spurious rules, and that in the real case study here considered $s\%$ and $c\%$ have been set equal to 0.01 and 0.8, respectively, which are values larger than those used in the synthetic case study ($s\%$ =0.0005 and $c\%$= 0.6).

Table 8 reports the numbers of extracted rules, whereas Table 9 reports the distribution of the 202 rules among the systems which the components included in the rules belong to. The majority of the rules associate components belonging to the same system, whereas 21 rules associate components of two different systems and no rules involve components of all the three systems. According to the experts of CERN, the large number of rules involving the components of the electric system (163) is caused by the fact that its components are intrinsically and hierarchical correlated (e.g., a malfunction of a high voltage electric component typically leads to malfunctions of low voltage components).

Table 8. number of extracted rules and identified groups of functionally dependent components.

Number of rules extracted	***Before pruning***	**1112**
	After pruning	**202**

Table 9. Distribution of the extracted 202 rules in terms of the systems which the components belong to.

Involved systems	Number of rules
Electric	163
Cryogenic	8
Cooling and Ventilation	10
Electric and Cryogenic	2
Electric and Cooling and Ventilation	2
Cryogenic and Cooling and Ventilation	17

The analysis of the results will focus on the association rules most interesting for the plant experts, i.e. those most difficult to identify using traditional system analysis techniques, such as those involving components of different systems and those characterized by values of the lift measure remarkably larger than 1, which indicate that the rule antecedent and consequent parts are strongly correlated.

Table 10 reports some examples of the identified rules. Since the subdivision of the alarms in rule antecedent and consequent parts does not imply causality among them, the authors are currently investigating the development of a method for the automatic identification of the casual chains of the alarms by considering their temporal sequences and the possible delays between the events and the corresponding alarms. In this work, the CTI experts have been asked to infer the causal chains of the involved alarms for some of the identified association rules. For example, *Rule 1,* which is characterized by a strong correlation among the alarms in the rule antecedent and consequent parts (lift equal to 266), describes the propagation of a malfunction triggered by a problem in a cooling tower of the CV system (revealed by the alarms a_{1123}^{2}, a_{1124}^{2})

to the low pressure compressors of the Cryogenic system (revealed by the alarms a^3_{100}, a^3_{101}, a^3_{102}), which, finally, cause faults in the high pressure compressors of the Cryogenic system (revealed by the alarms a^4_{123}, a^4_{124}). The analysis of the major failure events occurred in the past has shown that this chain of malfunctions has been responsible of a CTI shutdown occurred in 2016. The identification of this rule confirms the capability of the method of identifying real functional dependencies among the CTI components.

Table 10. Example of the generated association rules. Table 12 reports a description of the involved alarms.

	Antecedent **[System] {Alarm Identifier} ⇒**	Consequent **{Alarm Identifier} [System]**	Support	Confidence	Lift
Rule 1	$\begin{bmatrix} Cryogenic \\ Cryogenic \\ CV \end{bmatrix} \begin{Bmatrix} a^4_{100} \\ a^4_{101} \\ a^2_{1123} \end{Bmatrix} \Rightarrow$	$\begin{Bmatrix} a^3_{123} \\ a^3_{124} \\ a^2_{102} \\ a^2_{1124} \end{Bmatrix} \begin{bmatrix} Cryogenic \\ Cryogenic \\ Cryogenic \\ CV \end{bmatrix}$	0.0019	1.0	266
Rule 2	$\begin{bmatrix} Cryogenic \\ Cryogenic \end{bmatrix} \begin{Bmatrix} a^4_{123} \\ a^4_{124} \end{Bmatrix} \Rightarrow$	$\begin{Bmatrix} a^4_{100} \\ a^4_{102} \\ a^2_{101} \\ a^2_{4223} \end{Bmatrix} \begin{bmatrix} Cryogenic \\ Cryogenic \\ Cryogenic \\ Electric \end{bmatrix}$	0.0025	0.978	23
Rule 3	$\begin{bmatrix} CV \\ CV \end{bmatrix} \begin{Bmatrix} a^2_{1324} \\ a^2_{1374} \end{Bmatrix} \Rightarrow$	$\{a^2_{1424}\}[CV]$	0.0020	0.92	80
Rule 4	$[CV]\{a^2_{1724}\} \Rightarrow$	$\{a^2_{1094}\}[CV]$	0.0017	0.939	532
Rule 5	$\begin{bmatrix} Electric \\ Electric \\ Electric \\ Electric \\ Electric \\ Electric \end{bmatrix} \begin{Bmatrix} a^2_{3223} \\ a^2_{3613} \\ a^2_{3569} \\ a^2_{4445} \\ a^2_{4789} \\ a^2_{2991} \\ a^2_{3367} \end{Bmatrix} \Rightarrow$	$\begin{Bmatrix} a^2_{4523} \\ a^2_{2451} \\ a^2_{2937} \end{Bmatrix} \begin{bmatrix} Electric \\ Electric \\ Electric \end{bmatrix}$	0.0019	1.0	462
Rule 6	$\begin{bmatrix} Cryogenic \\ Cryogenic \end{bmatrix} \begin{Bmatrix} a^4_{723} \\ a^4_{793} \end{Bmatrix} \Rightarrow$	$\begin{Bmatrix} a^3_{192} \\ a^3_{561} \\ a^2_{245} \end{Bmatrix} \begin{bmatrix} Cryogenic \\ Cryogenic \\ Cryogenic \end{bmatrix}$	0.0418	0.697	16

Once the rules have been found, the procedure of Section 3.3 for the identification of groups of functional dependent components has been applied. Table 11 reports a group of components of all three systems, describing a cascade of dependent failures which originates from a malfunction in the electric system, propagates through the cooling and ventilation system and, finally, affects the cryogenic system. Table 12 provides the description of the alarms involved in Rule 1 of Table 10 and in the group of functionally dependent components of Table 11. The complete list of alarms involved in the dataset cannot be provided for confidentiality reasons.

According to the CTI experts and engineers, the analysis of the groups of functional dependent components provides practical indications useful to:

i) modify the planning of the periodic maintenance interventions and the spare parts management. For example, increase the frequency of the inspections of the components triggering the chains of malfunctions, with the objective of reducing the probability of their occurrences;

ii) develop plans to upgrade the most critical components of the identified groups and the system architecture, e.g. by implementation of new redundancies;

iii) identify emerging functional dependencies caused by uncontrolled changes and ageing of equipment, that result in new operating conditions and performances;

iv) facilitate root cause analysis by focusing the investigation on the identified groups of functionally dependent components. In particular, the traditional a posteriori analysis techniques, such as those based on oscilloperturbography and on the analysis of the trends of the key signals, can benefit from a preselection of the groups of functionally dependent components and the corresponding alarms;

Also, notice that:

i) although the proposed method has not been designed to support control room operators in real time, it can be useful to off-line develop operational strategies to be performed when the first alarms of the identified chains are triggered to stop the cascade of failures and mitigate its consequences. In particular, additional alerts (automatic or semi-automatic) can be implemented to warn the operators to anticipate preventive interventions. This is particularly important for the cryogenic system, which is a downtime amplifier due to the large inertia and thermal capacity of its components and cooling fluids. Therefore, any preventive alert which allows anticipating the recovery can significantly contribute to reduce the downtime of the system of the CTI;

ii) This will significantly reduce the down time of the system by stopping the chain of events and/or anticipating the recovery mode.

iii) the use of a time interval length (Δt) of 30 minutes allows identifying both groups of components which frequently fail together in a short period of time due to common cause failures and cascading failures characterized by delay in the propagation of the malfunctioning. The proper setting of the time interval length should consider the characteristics of the chains of events of interest. In this case study, since the LHC main components are large and heavy metal based superconducting magnets with significant thermal inertia coupled with large volume of cooling fluids, the time interval to be used has to be long enough to allow the propagation of the malfunctioning.

Table 11. Example of a group of functionally dependent components identified by the method. The description of the involved alarm is reported in Table 12

Component	**System**
c_{4728}	Electric
c_{100}	Cryogenic
c_{101}	Cryogenic

c_{102}	Cryogenic
c_{123}	Cryogenic
c_{124}	Cryogenic
c_{1123}	Cooling and Ventilation
c_{1124}	Cooling and Ventilation
c_{1161}	Cooling and Ventilation
c_{1163}	Cooling and Ventilation
c_{1165}	Cooling and Ventilation
c_{1553}	Cooling and Ventilation
c_{1554}	Cooling and Ventilation
c_{1555}	Cooling and Ventilation

Table 11. Description of the alarm involved in the rule of Table 9 and in the group of functionally dependent components of Table 10.

System	Alarm ID	Component ID	Component type	Alarm Description
Electric	a_{4728}^{6}	c_{4728}	UPS	Uninterruptible Power Supply Fault
Cryogenic	a_{100}^{3}	c_{100}	Low pressure compressor	Water flow not ok
Cryogenic	a_{101}^{3}	c_{101}	Low pressure compressor	Water flow not ok
Cryogenic	a_{102}^{3}	c_{102}	Low pressure compressor	Water flow not ok
Cryogenic	a_{123}^{4}	c_{123}	High pressure compressor	Water flow not ok
Cryogenic	a_{124}^{4}	c_{124}	High pressure compressor	Water flow not ok
Cooling and Ventilation	a_{1123}^{2}	c_{1123}	Cooling Tower	Malfunction
Cooling and Ventilation	a_{1124}^{2}	c_{1124}	Cooling Tower	Malfunction
Cooling and Ventilation	a_{1161}^{1}	c_{1161}	Ventilator	Short circuit ventilation cycle

Cooling and Ventilation	a^1_{1163}	c_{1163}	Ventilator	Short circuit ventilation cycle
Cooling and Ventilation	a^1_{1165}	c_{1165}	Ventilator	Short circuit ventilation cycle
Cooling and Ventilation	a^3_{1553}	c_{1553}	Pump	Short circuit pump
Cooling and Ventilation	a^3_{1554}	c_{1554}	Pump	Short circuit pump
Cooling and Ventilation	a^3_{1555}	c_{1555}	Pump	Short circuit pump

5. Conclusions and future work

CTIs are complex systems vulnerable to failures cascading through the dependencies among their sub-systems and components. The identification of functional dependencies among malfunctions of components is, however, very challenging considering traditional approaches of system decomposition and functional logic analysis. In this work, a method for identifying functional dependencies among components of a CTI relying on large scale alarm databases has been proposed. It is based on a novel procedure for the representation of the alarm database using a binary matrix, the application of the Apriori algorithm for rule mining and a novel algorithm for the identification of groups of functionally dependent components.

The application of the proposed method to a synthetic case study and a large scale database collected at CERN CTI has shown: *i*) its capability of identifying the functional dependencies among the components; *ii*) the need of using small values of the minimum support parameter of the Apriori algorithm for the identification of all the functional dependencies; *iii*) the fact that the association rules represent probabilistic relations among the occurrence of the alarms, which cannot be interpreted as causal relations among the alarms in the rule antecedent and consequent; *iv)* the finding that the most interesting groups of functionally dependent components are those involving components of different systems of the CTI, which are often unknown to plant experts.

The identified functional dependencies can be exploited for enhancing the CTIs safety. In particular, when considering vulnerability and resilience analyses, the identified functional dependencies are useful to: *i)* analyse the propagation of cascades of malfunctions and the recovery from them; *ii)* identify unknown combinations of failures, which can lead to the shutdown of the CTI; and *iii)* capture the dynamics of the CTI during accidental scenarios.

For future work, one direction lies in improving the capability of the method in identifying very rare functional dependencies. This requires the use of very small minimum support values, which can be unfeasible for the Apriori algorithm due to the computational effort required. Finally, methods based on formal concept analysis to transform data in lattices will be considered to discover relationships and dependencies in alarm databases.

It is worth highlighting that CTIs are typically equipped with monitoring systems measuring the time evolution of the signals related to their operation. Given the high information content of these numerical data, a natural continuation of this work is their use for the identification of functional dependencies in CTIs. This is expected to allow the identification of functional dependencies from data collected in normal operating conditions, without requiring occurrences of failures and malfunctions, which are typically rare events in CTIs. Also, the identification of functional dependencies among physical quantities can be very useful for anticipating and diagnosing the occurrence of chains of malfunctions.

The use of signal data to this purpose is challenged by the following technical and methodological issues:

1) large CTIs are made by tens of thousands of components, each of which is constantly monitored at high frequency by sensors measuring various signals such as temperature, pressure, vibrations etc. This requires to acquire and store very large quantities of data for a long period of time. This demands the use of complex and expensive Information Technology (IT) architectures (Hu et al., 2018; Kabir et al., 2019).
2) The identification of functional dependencies and the reconstruction of the resulting chains of malfunctions from hundreds of thousands of signals requires the development of complex algorithms able to explore their correlations.

Given the fast development of IT technologies for data acquisition and storage, and the growing capability of Artificial Intelligence (AI) algorithms in the extraction of information from increasingly large datasets, the authors will explore this research line.

References

Amani, Nahid, Mahmood Fathi, and Mehdi Dehghan. 2005. "A Case-Based Reasoning Method for Alarm Filtering and Correlation in Telecommunication Networks." *Canadian Conference on Electrical and Computer Engineering* 2005(May): 2182–86.

Antonello, F., Baraldi, P, A.,Gentile, U., Serio, L , Shokry A., Zio, E. 2020, "A Method for Inferring Causal Dependencies Among Abnormal Behaviours of Components in Complex Technical Infrastructures", Proceedings of the 30th International European Safety and Reliability Conference, ESREL-PSAM 2020

Antomarioni, S., Bevilacqua, M., Potena, D., & Diamantini, C. (2019). Defining a data-driven maintenance policy: an application to an oil refinery plant. International Journal of Quality & Reliability Management, 36(1), 77-97.

Agrawal, R., Imieliński, T., 1993. "Mining Association Rules Between Sets of Items in Large Databases", ACM SIGMOD Record, Volume 22, Issue 2, 6 January 1993, Pages 207-216.

Bastos, P., Lopes, I. and Pires, L.C.M. (2012), "A maintenance prediction system using data mining techniques", World Congress on Engineering, Vol. 3, pp. 1448-1453.

Billinton R., Allan R.N. 1992. Network modelling and evaluation of complex systems. In: Reliability Evaluation of Engineering Systems. Springer, Boston, MA

Billinton R., Allan R.N. 1996. Reliability Evaluation of Power Systems. Springer, Boston, MA.

Maurizio Bevilacqua, Filippo Emanuele Ciarapica, "Human factor risk management in the process industry: A case study", *Reliability Engineering & System Safety*, Volume 169, 2018, Pages 149-159

Djatna, T., Alitu, I.M., 2015. An Application of Association Rule Mining in Total Productive Maintenance Strategy: An Analysis and Modelling in Wooden Door Manufacturing Industry. Procedia Manuf. 4, 336–343.

Dorgo, Gyula, and Janos Abonyi. 2018. "Sequence Mining Based Alarm Suppression." *IEEE Access* 6: 15365–79.

Etesami, J., Kiyavash, N. 2017, "Measuring causal relationships in dynamical systems through recovery of functional dependencies" (2017) *IEEE Transactions on Signal and Information Processing over Networks*, 3 (4), art. no. 7782866, pp. 650-659.

Fayyad, Usama, Gregory Piatetsky-shapiro, and Padhraic Smyth. 1996. "From Data Mining to Knowledge Discovery in Databases." 17(3): 37–54.

Filip, F. G. 2008. "Decision Support and Control for Large-Scale Complex Systems." *Annual Reviews in Control* 32(1): 61–70.

Han, Jiawei, J Pei, and Y Yan. 2000. "Mining Frequent Patterns without Candidate Generation." *SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*: 1–12.

Hatonen, K. et al. 1996. "Knowledge Discovery from Telecommunication Network Alarm Databases." *Proceedings of the Twelfth International Conference on Data Engineering* (March).

Hu, Keyun, Yuchang Lu, Lizhu Zhou, and Chunyi Shi. 1999. "Integrating Classification and Association Rule Mining: A Concept Lattice Framework." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 1711(1): 443–47.

L. Hu, W. Sheng, K. Liu and Z. Lin, "A Bad Data Identification Method for Multiple Spatio-temporal Data in Power Distribution Network," *2018 International Conference on Power System Technology (POWERCON)*, Guangzhou, 2018, pp. 4083-4088

Hui, C Y et al. 2005. "Mining Quantitative Associations in Large Database." *Software Engineering And Middleware* 3399(60373053): 405–16.

Irene Eusgeld, Cen Nan, Sven Dietz, "System-of-systems approach for interdependent critical infrastructures", 2011, *Reliability Engineering & System Safety*, Volume 96, Issue 6, 2011, Pages 679-686.

Jain, Ani K, Robert P W Duin, and Jianchang Mao. 2000. "Statistical Pattern Recognition : A Review." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1): 4–38.

Karoly, R., and J. Abonyi. 2017. "Multi-Temporal Sequential Pattern Mining Based Improvement of Alarm Management Systems." *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*: 3870–75.

Klemettinen, M., H. Mannila, and H. Toivonen. 1999. "Interactive Exploration of Interesting Findings in the Telecommunication Network Alarm Sequence Analyzer TASA." *Information and Software Technology* 41(9): 557–67.

Klemettinen, Mika, Heikki Mannila, and Hannu Toivonen. 1999. “Rule Discovery in Telecommunication Alarm Data.” *Journal of Network and Systems Management* 7(4): 395–423.

W. Kröger, E. Zio, 2011, “Vulnerable Systems” (2011), Springer, London 2011

Kabir, G., Tesfamariam, S., Hemsing, J., & Sadiq, R. (2019). *Handling incomplete and missing data in water network database using imputation methods. Sustainable and Resilient Infrastructure, 1–13.*

Kamsu-Foguem, B., Rigal, F., Mauget, F., 2013. Mining association rules for the quality improvement of the production process. Expert Syst. Appl. 40, 1034–1045.

Li, Tong Yan, and Xing Ming Li. 2011. “Preprocessing Expert System for Mining Association Rules in Telecommunication Networks.” *Expert Systems with Applications* 38(3): 1709–15.

Jonas Johansson, Henrik Hassel, “An approach for modelling interdependent infrastructures in the context of vulnerability analysis”, *Reliability Engineering & System Safety*, Volume 95, Issue 12, 2010, Pages 1335-1344.

Lozonavu, Mihaela, Martha Vlachou-Konchylaki, and Vincent Huang. 2016. “Relation Discovery of Mobile Network Alarms with Sequential Pattern Mining.” *2017 International Conference on Computing, Networking and Communications (Icnc)*: 363–67.

Mannila, Heikki, Hannu Toivonen, and A. Inkeri Verkamo. 1994. “Efficient Algorithms for Discovering Association Rules.”*AAAI Workshop on Knowledge Discovery in Databases KDD94* 118(1–4): 181–92.

Martínez-de-Pisón, F.J., Sanz, A., Martínez-de-Pisón, E., Jiménez, E., Conti, D., 2012a. Mining association rules from time series to explain failures in a hot-dip galvanizing steel line. Comput. Ind. Eng. 63, 22–36.

Mosleh, A. 1991. Common cause failures: an analysis methodology and examples. Reliab Eng Syst Safe 1991; 34: 249–292.

J. Nielsen and L. Serio, "Technical Services: Unavailability Root Causes, Strategy and Limitations", *in Proc. 7th Evian Workshop on LHC beam operation*, Evian Les Bains, France, December 2016.

NUREG/CR 2003, NRC Nuclear Regulatory Commision, Probabilistic Risk Analysis: Procedure Guide. Washington, DC, 1983

Oborski, Przemysław. 2014. “Developments in Integration of Advanced Monitoring Systems.” *International Journal of Advanced Manufacturing Technology* 75(9–12): 1613–32.

O'Connor, A., Mosleh, A., 2016. A general cause based methodology for analysis of common cause and dependent failures in system risk and reliability assessments (2016) Reliability Engineering and System Safety, 145, pp. 341-350.

Oswaldo, Vicente, and Baez Monroy. 2006. “Neural Networks as Artificial Memories for Association Rule Mining.”

Pasquier, Nicolas et al. 2005. “Generating a Condensed Representation for Association Rules.” *Journal of Intelligent Information Systems* 24(1): 29–60.

Park, J.S., Chen, M.-S., Yu, P.S. 1995“*An Effective Hash-Based Algorithm for Mining Association Rules*”. *ACM SIGMOD Record*, 24 (2), pp. 175-186.

Pei, J., Han, J., Lu, H., Nishio, S., Tang, S., Yang, D. 2007, “*H-Mine: Fast and space-preserving frequent pattern mining in a large databases*” *IIE Transactions (Institute of Industrial Engineers)*, 39 (6), pp. 593-605

Priss, U. Formal concept analysis in information science (2006) *Annual Review of Information Science and Technology*, 40, pp. 521-543.

Sage, Andrew P, and Christopher D Cuppan. 2001. “On the Systems Engineering and Management of Systems of Systems and Federations of Systems.” *Inf. Knowl. Syst. Manag.* 2(4): 325–45.

Serio, L., Antonello, F., Baraldi, P., Castellano, A.,Gentile, U., Zio, E. 2018, “*Smart framework for the availability and reliability assessment and management of accelerators technical facilities*” 9th International Particle Accelerator Conference, IPAC 2018

Sinda Rebello, Hongyang Yu, Lin Ma, “An integrated approach for system functional reliability assessment using Dynamic Bayesian Network and Hidden Markov Model”, *Reliability Engineering & System Safety*, Volume 180, 2018, Pages 124-135.

Singh, Archana, Megha Chaudhary, Ajay Rana, and Gaurav Dubey. 2011. “Online Mining of Data to Generate Association Rule Mining in Large Databases.” *2011 International Conference on Recent Trends in Information Systems*: 126–31.

Srikant, Ramakrishnan, and Rakesh Agrawal. 1996. “Mining Quantitative Association Rules in Large Relational Tables.” *ACM SIGMOD Record* 25(2): 1–12.

B. Todd *et al*, "LHC Availability 2016: Standard Proton Physics", CERN, Geneva, Switzerland, Rep. CERN-ACC.NOTE-2016-0067, December 2016.

Wang, Jiantao et al. 2017. “Efficient Alarm Behavior Analytics for Telecom Networks.” *Information Sciences* 402: 1–14.

Maik Reder, Nurseda Y. Yürüşen, Julio J. Melero, “Data-driven learning framework for associating weather conditions and wind turbine failures”, *Reliability Engineering & System Safety*, Volume 169, 2018, Pages 554-569

Wille, R. Restructuring lattice theory: An approach based on hierarchies of concepts (2009) *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5548 LNAI, pp. 314-339.

Witten, I H, and E Frank. 2016. *Data Mining: Practical Machine Learning Tool and Techniques*. Morgan Publishers.

Zaki, Mohammed J. 2000. “Generating Non-Redundant Association Rules.” *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’00*: 34–43.

Zio, E. 2009. Computational Methods for Reliability and Risk Analysis, World Scientific, 2009.

Zio, E. 2016. “Some Challenges and Opportunities in Reliability Engineering.” *IEEE Transactions on Reliability* PP(99): 1769–82.