



Identifying Dependency Relationships Between Events in Production Systems

Pedro Nunes¹, Isabel Lopes^{1,2(✉)}, Luís Basto², and Cláudia Pires³

¹ Department of Production and Systems, School of Engineering, University of Minho, 4800-058 Guimarães, Portugal
ilopes@dps.uminho.pt

² ALGORITMI Research Centre/LASI, School of Engineering, University of Minho, 4800-058 Guimarães, Portugal

³ Technological Center for the Metal Working Industry (CATIM), Rua dos Plátanos 197, 4100-414 Porto, Portugal

Abstract. Industry 4.0 is directly related to the digital transformation of organizations. One of the main characteristics of this transformation is the ability to analyze and manage data to obtain relevant and previously unknown information from it. Specifically, in the context of operational risk management, an efficient analysis of the data that typically come from occurrence records, can support decision-making. This paper presents a new algorithm to identify dependencies between events in productive environments, such as equipment failures, defects, or safety and environmental incidents, supporting the decision-making process in the context of analyzing and managing operational risks. The algorithm was developed based on association rule mining algorithms, then tested with data from a Portuguese metalworking company, and finally validated. The test results show that the developed algorithm is capable of identifying dependencies, which have to be confirmed by specialized staff to conclude if they are causal relationships.

Keywords: Association Rule Mining · Data Analysis · Frequent Pattern Mining · Risk Management

1 Introduction

The fourth industrial revolution or Industry 4.0 is characterized by the ability of organizations to take advantage of technological development for industrial applications [1]. The integration of information and communication technologies with the various components of production systems results in a huge amount of data. Therefore, new technologies need to be used in the best way to process data efficiently and thereby improve the management of processes, including the process of operational risk analysis and management [2].

Operational risk analysis and management focus on the uncertainty inherent to the execution of activities that an organization needs to carry out to achieve its goals [3]. In most companies, operational risk is the most common form of risk and should be

seen as the most dangerous and the most difficult to understand because it results from losses related to failed or inadequate internal processes [4]. Thus, inefficient operational risk analysis and management can lead to the loss of customers, limitation of productive efficiency, and inefficient resource allocation [5].

One way to improve operational risk analysis and management is to discover previously unknown information to improve decision-making. For this purpose, the application of association algorithms can be a useful tool, because they are based on a technique called Frequent Pattern Mining, which is characterized by the ability to locate the repeating relationships between different items in a data set and represent them in the form of association rules. As an example of its usefulness, Wang *et al.* [6] presented a data-focused approach based on association algorithms and frequent pattern mining that allows the extraction of relevant information from historical records, such as association rules between keywords of each record and their conditional probability. As shown in a literature review carried out by Chee *et al.* [7], several algorithms have been developed over time to find frequent patterns, however, they assume that the data set must be explicitly in the form of transactions, restricting its application variability and increasing the possibility of losing potentially relevant relationships in certain scenarios. Antomarioni *et al.* [8] propose an approach to support the analysis of failures in production processes, using an association rule mining algorithm, but which presents the limitation of not considering the temporal dimension to extract association rules.

This paper presents a new algorithm that aims to relate disruptive events in production systems related to different areas such as quality (i.e., defects), safety (i.e., work accident), environment (i.e., gas leak), production (i.e., production stoppage) or maintenance (i.e., machine failure). This algorithm was developed based on existing association rule mining algorithms to achieve the intended results and overcome the aforementioned limitations, calculating similar metrics. The methodology used was CRISP-DM, explained in detail by Huber *et al.* [10]. The need to relate events from different areas arises since sometimes these events have a common root cause and therefore must be treated collaboratively by the various departments [9]. For instance, if the occurrence of a non-compliant product ends up being related to work accidents may indicate that there is a common and potentially unknown root cause for these events. Thus, realizing that two events are related makes it possible to approach them more effectively.

This paper is organized as follows: Sect. 2 presents the proposed algorithm to identify relationships between events, Sect. 3 includes the testing and validation of the algorithm using real data from a Portuguese metalworking company and Sect. 4 refers to the conclusion and future work.

2 The Association Rule Mining Algorithm

The proposed algorithm was developed from existing association rule mining algorithms. To overcome the fact that the dataset is not in the form of transactions, time windows are considered to search for possible dependency relationships.

The algorithm is triggered by the introduction of a dataset, containing the records of events and their times of occurrence (t_{oco}), combined with a user input regarding the analysis window (u_{input}) that he considers appropriate for the algorithm to find

possible dependency relationships (e.g., 5 days or 100 000 units). That is, for each instance (event) of a given dataset, the algorithm will try to find “who” (previous events) has the potential to have caused it considering the window defined by the user.

With this purpose, the algorithm was designed to perform a set of steps to identify possible relations between two events, an antecedent and a consequent event, and count their number of occurrences to identify an association rule depending on the value of the calculated metrics, as shown in Fig. 1.

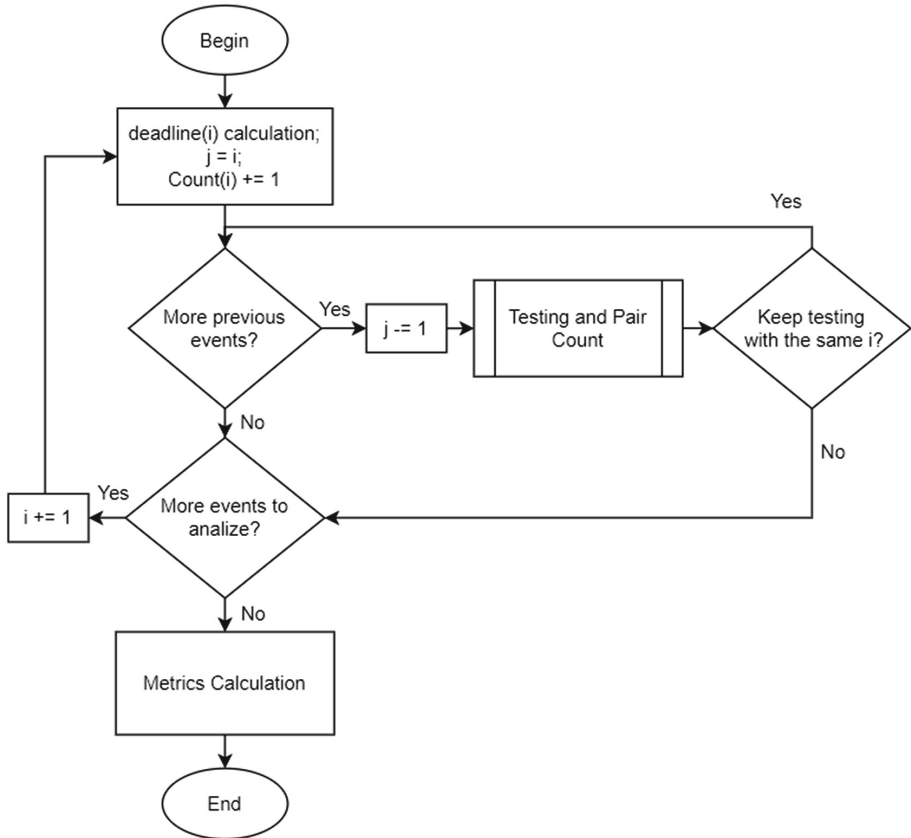


Fig. 1. Algorithm flowchart

For a given row of the dataset (event occurrence), the associated event is considered as the event under analysis (i) and its count is incremented by one ($\text{count}(i) += 1$), representing the number of times the event was registered in the dataset.

The calculation of the deadline (t_{lim}) for the event under analysis is then performed, representing the moment when a given event can no longer produce effects on others.

$$t_{lim} = t_{oco} + u_{input} \quad (1)$$

In the next step, it is checked which of the previous events (j) has the potential to have triggered the event under analysis (i).

To this end, a reverse search is started, testing the previous events based on three main conditions, where all of them must be satisfied for the pair count to be incremented.

1. The previous recorded event must have a deadline (t_lim) greater or equal to the time of occurrence (t_occ) of the event under analysis, ensuring that the former has the potential to have triggered the latter, being potentially considered the antecedent event.

$$t_lim(j) \geq t_oco(i) \quad (2)$$

If the result of this condition is false, the test for the event under analysis (i) is stopped, advancing to a new event ($i += 1$) if it exists.

2. To be considered an antecedent event, the previous event must be different from the event under analysis.

$$event(j) \neq event(i) \quad (3)$$

If this condition is not checked, the test for the event under analysis (i) is stopped, advancing to a new event ($i += 1$) if it exists. Additionally, this previous event record is put into inactive mode by using a binary variable “flag” that by default presents the value of 0 and assumes, in these cases the value of 1 ($Flag(j) = 1$). Its usefulness is explained in condition 3.

3. The previous event record must be active at the time of the test to be potentially considered the antecedent event.

$$Flag(j) = 0 \quad (4)$$

As indicated in the second condition an event record may become inactive, causing it not to be considered thereafter. The goal is to avoid counting the same pair of events more than once based on the same previous event record. If this condition is not checked, the testing proceeds with the same event (i) as the possible consequent event.

With the three main conditions tested and checked the count of the pair ($j \rightarrow i$) is incremented by one, representing the number of times that two given events were recorded in a time interval less than or equal to the analysis window.

However, a fourth condition has been defined indicating that a particular pair will also be counted in the opposite direction ($i \rightarrow j$) in the case when both events occur at the same time since it is impossible to know which of the events is the antecedent.

4. That is, if the previous recorded event has a time of occurrence ($t_occ[j]$) equal to the time of occurrence ($t_occ[i]$) of the event under analysis.

$$t_oco(j) = t_oco(i) \quad (5)$$

In the last step for each of the pairs, metrics are calculated to provide the end user with greater knowledge about them.

- **Confidence:** Two metrics are considered: C1 and C2. Both are conditional probabilities.

$$C1 = \frac{Count(j \rightarrow i)}{Count(j)} \quad (6)$$

$$C2 = \frac{Count(j \rightarrow i)}{Count(i)} \quad (7)$$

C1 means how likely event i occurs after event j (within the time window), given that event j has occurred. C2 means how likely event j occurs before event i (within the time window), given that event i has occurred. They indicate the level of certainty that the pair presents.

- **Average time:** Displays to the end user the average time that normally occurs between the two events of a given pair, indicating the time he may have at his disposal to prevent the consequential event from occurring.

$$\bar{T}_{j \rightarrow i} = \frac{\sum(t_{oco(i)} - t_{oco(j)})}{Count(j \rightarrow i)} \quad (8)$$

3 Testing and Validation

To test and validate the developed algorithm, data from a Portuguese metalworking company were used, specifically the history of corrective maintenance actions related to a machine tool from a progressive stamping process, whose main components are punches and die inserts. The data were collected through forms filled in by machine operators and maintenance technicians, during a period of approximately a year and a half. The records are related to failure occurrences, where each failure is a specific damaged component through wear. Thus, in this case, each event is identified by the name of the damaged component.

Analyzing the data associated with the machine tool, 109 occurrences were recorded during the production of 6 191 466 parts. The data were then prepared to serve as input to the algorithm and an analysis window (user input) of 100 000 production parts was defined. Additionally, to form association rules and increase the probability of a given pair representing a dependency relationship between two events a filtering condition was defined.

$$(Count \geq 10 \cap (C1 > 90\% \cup C2 > 90\%)) \quad (9)$$

Subsequently, the algorithm was executed, and the results presented in Table 1 were obtained.

The evaluation from a computational and algorithmic perspective was positive since the algorithm can identify possible dependency relationships between events of a given data set.

As an example, through the output of the algorithm, the user can obtain the following information about the pair ($4H \rightarrow 20D$):

- 4H can be the cause of 20D.

Table 1. Testing Output

Pair	Count	$\bar{T}_{j \rightarrow i}$	C1	C2
20D \rightarrow 4H	18	5490	40.9	100
4H \rightarrow 20D	17	84	94.4	38.6
20C2 \rightarrow 20C1	10	8647	90.9	66.7
20D \rightarrow 20C2	10	21897	22.7	90.9

- The occurrence of this pair, considering an analysis window of 100 000 production parts, was checked 17 times.
- On average, 20D occurs 84 production parts after 4H has occurred.
- 94.4% of the time 4H was registered, 20D was recorded during the production of the next 100 000 production parts.
- 38.6% of the times 20D was recorded, it was after the 4H registration, based on a 100 000 production parts analysis window.

The results presented were also analyzed and evaluated by the company's technical managers. The objective was to understand whether the pairs of events proposed by the algorithm are somehow representative of what happens daily in production regarding machine wear and failure.

In this sense, the following conclusions were drawn over the pairs suggested by the algorithm:

- **(20D \rightarrow 4H/4H \rightarrow 20D)** – This combination of events has been suggested both ways by the algorithm. This is because they occur or at least are recorded at the same time, leaving the question of which of them will be the trigger of the other. However, from a procedural and mechanical perspective, the pairs presented for this tool should be in the direction of “4 \rightarrow 20”, therefore the pair considered as correct and that makes sense in practice is the (4H \rightarrow 20D), whereas 4 is a die insert and 20 is a punch, and it will make more sense to be a damaged matrix to damage the puncture and not the other way around.
- **(20C2 \rightarrow 20C1)** – They are two components that work in parallel and perform the same function under identical conditions, when one of them fails is common the other also fails. Thus, it was deduced that this pair of events presents an association relationship, but not causality.
- **(20D \rightarrow 20C2)** – Initially, no physical justification was found for the pair, so it seemed to make no sense. Since these are two punctures that perform apparently unrelated operations at different stations. It may eventually be reevaluated in the future regarding a temporal relationship (time between failures) if the algorithm continues to suggest it. However, the low confidence value reinforces the pair's lack of logical sense.

With the test performed it was noticed that the result of the algorithm can be conditioned by the user's choice regarding the time window of analysis, which can be seen as a limitation.

Also, the pairs presented by the algorithm have a suggestive character since they always require final validation by the user and the relationship between two events of a given pair may come from a causality ($4H \rightarrow 20D$), an association based on work conditions ($20C2 \rightarrow 20C1$), or merely temporal ($20D \rightarrow 20C2$).

4 Conclusions

The ability to use data to support decision-making should be seen as the new standard for the future of organizations. In operational risk analysis and management obtaining previously unknown information regarding how events relate to each other can improve resource allocation and increase productivity.

Thus, in this article, an association algorithm was developed capable of suggesting dependency relationships between two events, introducing a new way to apply association algorithms, where it is not necessary to have transactional databases as input thus allowing the inclusion of the temporal dimension in the discovery of frequent patterns. More specifically, with the application of the proposed algorithm, the end-user is presented with a set of pairs of events, which should be analyzed to understand whether they make sense and, based on this conclusion, make it possible to make decisions efficiently.

The test performed with the data from the Portuguese metalworking company allowed an understanding of the limitations and the algorithm validation since it was able to find some relationships between the recorded events. Later, the algorithm is intended to be implemented in collaborative software to be used in the most diverse industrial scenarios.

For future work, the objective is to mitigate some of the limitations of this algorithm as its subjective adjacent to human intervention in the choice of the analysis window and the suggestive character of the pairs presented.

Acknowledgment. The authors would like to express appreciation for the companies involved. This work has been supported by Norte 010247 FEDER 046930 – CORIM I&DT.

References

1. Filz, M.-A., Langner, J.E., Herrmann, C., Thiede, S.: Data-driven failure mode and effect analysis (FMEA) to enhance maintenance planning. *Comput. Ind.* **129**, 103451 (2021). <https://doi.org/10.1016/j.compind.2021.103451>
2. Niesen, T., Houy, C., Fettke, P., Loos, P.: Towards an integrative big data analysis framework for data-driven risk management in industry 4.0. In: 49th Hawaii International Conference on System Sciences (HICSS), pp. 5065–5074 (2016). <https://doi.org/10.1109/hicss.2016.627>
3. Raz, T., Hillson, D.: A comparative review of risk management standards. *Risk Manage* **7**, 53–66 (2005). <https://doi.org/10.1057/palgrave.rm.8240227>
4. Williams, R., et al.: Quality and risk management: what are the key issues? *TQM Mag.* **18**, 67–86 (2006). <https://doi.org/10.1108/09544780610637703>
5. Elahi, E.: Risk management: the next source of competitive advantage. *Foresight* **15**, 117–131 (2013). <https://doi.org/10.1108/14636681311321121>

6. Wang, X., Huang, X., Zhang, Y., Pan, X., Sheng, K.: A data-driven approach based on historical hazard records for supporting risk analysis in complex workplaces. *Math. Probl. Eng.* **2021**, 1–15 (2021). <https://doi.org/10.1155/2021/3628156>
7. Chee, C.-H., Jaafar, J., Aziz, I.A., Hasan, M.H., Yeoh, W.: Algorithms for frequent itemset mining: a literature review. *Artif. Intell. Rev.* **52**(4), 2603–2621 (2018). <https://doi.org/10.1007/s10462-018-9629-z>
8. Antomarioni, S., Ciarapica, F.E., Bevilacqua, M.: Association rules and social network analysis for supporting failure mode effects and criticality analysis: framework development and insights from an onshore platform. *Saf. Sci.* **150**, 105711 (2022). <https://doi.org/10.1016/j.ssci.2022.105711>
9. Basto, L., Lopes, I., Pires, C.: Risk analysis in manufacturing processes: an integrated approach using the FMEA method. In: Kim, D.Y., von Cieminski, G., Romero, D. (eds.) *APMS 2022. IFIP AICT*, vol. 663, pp. 260–266. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-16407-1_31
10. Huber, S., Wiemer, H., Schneider, D., Ihlenfeldt, S.: DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model. *Procedia CIRP* **79**, 403–408 (2019). <https://doi.org/10.1016/j.procir.2019.02.106>