CaseID detection for Process Mining: a heuristic-based methodology

Roberta De Fazio[1][0000−0002−0271−132X], Antonio Balzanella[1][0000−0001−8370−6006], Stefano Marrone[1][0000−0003−1927−6173], Fiammetta Marulli[1][0000−0001−5226−2326], Laura Verde[1][0000−0003−2422−1732], Vincenzo Reccia[2], and Paolo Valletta[2]

[1] Università della Campania "Luigi Vanvitelli" - Dipartimento di Matematica e Fisica, viale Lincoln, 7 — Caserta, Italy
{roberta.defazio,antonio.balzanella,stefano.marrone, fiammetta.marulli,laura.verde}@unicampania.it
[2] Gematica srl, via Diocleziano 107 — Naples, Italy
{v.reccia,p.valletta}@gematica.com

Abstract. Process Mining is getting a growing interest in many contexts where performance bottlenecks are critical for the business. Unfortunately, real cyber-physical systems are usually not implemented to easily address these techniques. One of the most frequent problems to face is transforming acquired data, often heterogeneous and unlabelled, to allow the application of Process Mining technique. In this study, we propose an automatised and unsupervised methodology for extracting *CaseIDs* from an unlabelled event log. The proposed detection of *CaseIDs* is based on the definition of appropriate heuristic metrics, able to highlight the correlation between events that are part of the same process instance, according to temporal and semantic features (e.g., kinds of functionally-related devices, topological distance, etc.). These features constitute the inputs for a clustering technique, which has been used to extract different cases. The performance of the proposed methodology was evaluated on a real diagnostic management system to support the decisions in maintenance operations in railway infrastructures.

Keywords: Unlabelled Event Log · CaseID detection · Event Data · K-Means Clustering · Predictive Maintenance · Railway Infrastructure

1 Introduction

The massive use of information technologies and the development of the Internet of Things (IoT) have produced a huge amount of data in different contexts: from smart cities to Industrial Internet of Things (IIoT) and Supervisory Control And Data Acquisition (SCADA) systems. This change is driving the need to extract valuable insights from this data. Nowadays, Data Mining is a mature body of knowledge and methods, proposing new techniques applicable to real-life problems. Process Mining (PM) is affirming as one of the most valuable set

of techniques able to extract explicit, useful knowledge from real data, bridging computational intelligence and process modelling [20]. PM has found multiple applications throughout the years [9]. Critical infrastructures like transportation, public health, and telecommunication networks can enhance their reliability and security with the support of PM. The extracted models offer monitoring capabilities, detecting anomalies and cyber-attacks. They also enhance understanding of system criticality, boosting resilience, and reducing vulnerabilities [14].
The event logs constitute the starting point for PM: an event log involves a set of cases, stored in a multiple-record table. Each case is a sequence of events executed in a single process instance. Each record has a precise structure in which the key attribute groups are clearly prescribed. First, all the events belonging to the same case are marked by a *CaseID*, which constitutes one of the discriminant features of an event log. Moreover, an event is, usually, characterized by other attributes such as a *timestamp*, a corresponding *activity*, and some *resources* involved in the logged event [17]. However, this well-defined data structure is difficult to meet in real-world applications [5, 15]. One of the most discussed issues in the literature is *CaseID* detection, i.e., the identification of events belonging to the same case: the event logs in which the values of this attribute are not a-priori determined are named *unlabelled event logs* [2]. Erroneous or invalid values of *CaseID* can damage the accuracy and reliability of the model [8].
This work is framed in Cyber-Physical Systems (CPSs) domain. Such systems are often characterized by a high level of heterogeneity and coexistence of people, legacy hardware and software, and natural and external events that affect the performance and availability of the systems themselves. As an example, if we consider a smart building, many of the different interactions between the users and the building itself cannot be accurately tracked. To this aim, having a post-processing tool able to understand which events are related to the same case, is of paramount importance to effectively run PM algorithms.
Several approaches can be adopted to detect appropriate CaseIDs in an unlabeled event log. [4, 11, 15]. In this paper, we introduce a heuristic-based methodology for the determination of *CaseIDs* in unlabelled datasets, necessary to define a PM model. In particular, we propose the application of a specifically addressed clustering technique in an unsupervised approach. The core of this methodology is the definition of metrics, based on heuristics that model guidelines and insights provided by domain experts. In this context, the *CaseID* represents a system failure - i.e. a sequence of faults that are correlated and lead to the delivery of an incorrect service. The metrics, that quantify the distance between couples of events, are designed in order to catch the characteristics of a process instance (for example the time frame, the location of the faults' chain in the infrastructure or the functional dependencies among the components involved) hidden among the events' attributes. A case study considering a monitoring system for transportation and distribution infrastructures is used to demonstrate the potentiality of the proposed methodology. In detail, railway infrastructure monitoring is used as a test bench for the approach.

The rest of the paper is organized as follows. Section 2 describes the main existing approaches of *CaseID* detection for PM processes. Section 3 describes the proposed methodology to determine these cases automatically, while Section 4 describes its application to a concrete monitoring system. Conclusions are drawn in Section 5.

2 Related work

Recently, PM has found several applications in different fields involving real-life contexts from business informatics to critical infrastructures, moving from academia towards the industry. Many scientific papers consider the event logs already structured and labelled. However, in real-life problems, it may be difficult to have event logs properly defined with all the key attributes [17]. Hence, the pre-processing constitutes a fundamental phase to achieve an accurate and reliable analysis [13]. The quality of the input, in fact, affects the performances of the algorithms, the readability of the process model and the interpretation of the results [7, 13].

In [2], an approach called Deduce Case IDs (DCI) has been introduced, which oversees generating labelled events from cyclic processing defined by a relation matrix derived from the process model. In [8], the authors propose an approach for identifying a *CaseID*, knowing only the sequence of the activities. Bayomie et al., propose an approach that infers the *CaseID* by solving a multi-level optimization problem, using the fitness metrics calculated from the unlabelled logs and the process model, and looking for the nearest optimal correlated log [3]. Other authors extract this knowledge using association rules and defining cost functions on the base of these rules [4]. However, all these approaches require information about the performed activities and how they are related in the model to build. All the described inputs could be unavailable in real case studies, especially if considering a complex CPS. In fact, in this class of systems, people and things interact together, letting complex behaviours emerge. In these situations, the mechanism of labelling each action with a specific *CaseID* is often unfeasible.

In the literature, there are also some works providing methodologies that do not require a process model as input: in [5], the authors define a methodology for *CaseID* detection, based on the correlation between different activities that show the same values for that so-called decorative attributes -i.e. that are not strictly required for applying PM technique- of the event logs. Although they start from the assumption that the *CaseID* is a combination of these attributes and their values depend on the type of activity in the respective log entry. Lichtenstein et al., instead, propose an approach where the *CaseID* is identified in an unlabelled event log without any information about the process model. They introduce a strategy based on the division of the event log into several classes of data models defined by an activity-attribute relationship diagram [11].

Most of the studies existing in the literature propose methodologies that, starting from a model or from the known relationship among activities, leverage knowledge about the process itself, building tools for inference using a supervised

approach. We should assume that in a real context, we could not be aware of these specific details as in the case study proposed.

In this study, we propose a bottom-up strategy: starting from every single event, a set of events is clustered in different cases using some metrics defined considering guidelines that enhance the meaning of the domain attributes. In other words, the strategy proposed is formulated without using any supervision by a process model or any kind of control flow, although guided by some knowledge of the domain experts able to catch the peculiarities of the considered domain.

3 Methodology

The proposed methodology aims at identifying an automatised and unsupervised way, based on domain-oriented heuristics, to extract *CaseIDs* from an event log where they are not provided. The heuristics should be defined with the support of domain experts. Before describing such an approach, a general introduction of the application domain is due. Fig. 1 puts at the centre of the study a communication network, composed of Network Agents that are the elements (e.g., a router, a switch, a Base Transceiver Station (BTS) in case of wireless communication) of the network enabling the communication of domain-specific ends. Examples of these ends are trains, trackside controllers, rail switches (railway); smart Heating, Ventilation and Air Conditioning (HVAC), lighting systems, connected healthcare equipment (smart hospitals); infotainment car systems, roadside equipment, Smart Road Centres (smart vehicles).

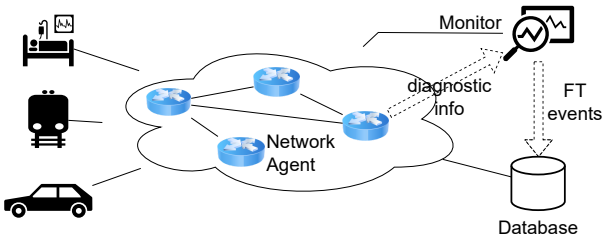

Fig. 1: Monitored system structure.

As these systems are in operation, the interaction between such elements solicits the Network Agents that exchange messages according to the adopted communication protocol. To monitor the correct functioning of the network, a Monitor is responsible for periodically polling Network Agents and for auditing for specific diagnostic messages detecting failures or performance degradations. The Monitor is responsible for populating an On-line Transaction Processing (OLTP)-level Database (DB) (i.e., the *Database*) with these Fault Tolerance (FT)-related events. Among these events, there are faults, system failures, performance degradation, repair actions, and returning to full operation.

Based on the schema, this work proposes the workflow depicted in Fig. 2, which is divided into four phases. Starting from the *Database*, two preparatory

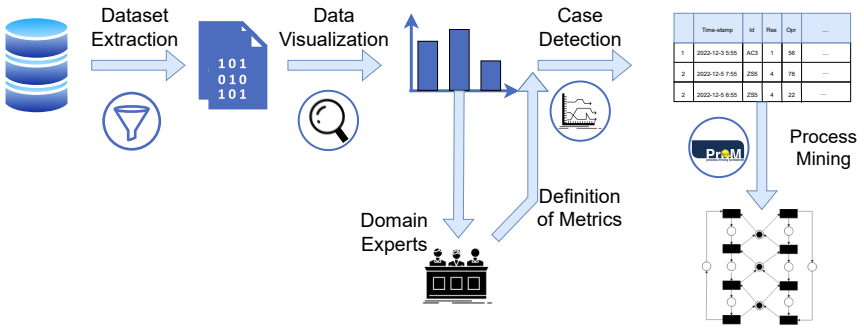

Fig. 2: The proposed workflow.

phases are related to the extraction of valuable features from the Database itself into a dataset on which proper visualisation actions are performed. These phases are respectively *Dataset Extraction* and *Data Visualization*, and they are performed to capture evident relationships between the considered features and could be used to reduce the dimensionality of the problem. Other useful actions could be performed in synergy (e.g., Principal Component Analysis (PCA) and T-distributed Stochastic Neighbour Embedding (t-SNE)).

Another important effect of the Data Visualization phase is to design a concrete playground for the definition of heuristics by the Domain Experts. The reduction of the dimensionality allows the domain expert to address the most relevant aspects on which heuristics metrics could be defined, as it will be soon explained in detail. The *Case Detection* phase, which is the core of the present research, aims at inferring the cases from both the considered dataset and the metrics determined by the Domain Experts. A process instance is meant as a system failure and the metrics are defined to measure similarity between events in terms of process features. The output of such a phase is a labelled event log, in which correlated faults have the same *CaseID*. The last phase, the *Process Mining*, applies common PM algorithms and toolchains to extract readable and explicit models of the Predictive Maintenance (PdM) models.

3.1 The Case Detection Phase

The basic steps constituting the *CaseID Detection* are depicted in Fig.3: first of all, the entities and relations are identified, resulting in the definition of the methodology formal model. In detail, let $\mathcal{D} = \{d_1, d_2, \ldots, d_m\}$ be the set of the monitored devices, and let $E = \{e_1, e_2, \ldots, e_n\}$ be the set of the events of interest. Let us define $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ a set of attributes characterizing the devices. In this way, each device d_i is characterized by a set of values for each attribute in $\mathcal{A}$. The function $\mathcal{R}$, defined according to Eq. 1, is a non-bijective function

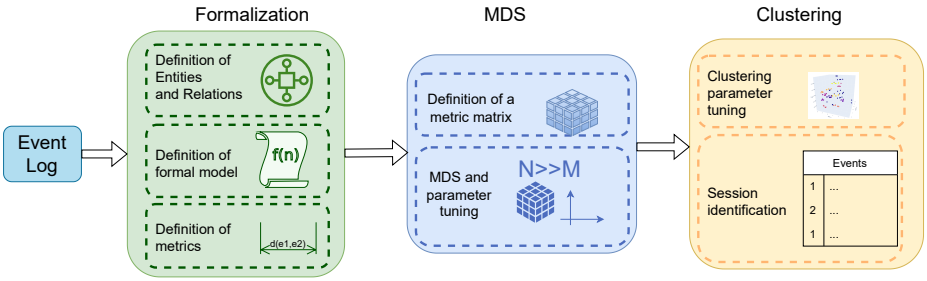

Fig. 3: The Case Detection phase workflow.

that relates events and devices; in fact, some devices may not be involved in any event as well as in many events.

$$\mathcal{R} : E \to \mathcal{D} \tag{1}$$

Eq. 2 defines another function, which assigns a timestamp to every event; this function is not bijective, since more than one event could occur at the same time. The range of this function is the set of real numbers because the timestamp is meant as its conversion to a float number.

$$\mathcal{T} : E \to \mathbb{R} \tag{2}$$

For each $a_s \in \mathcal{A}$, a function φ_{a_s} is defined as in Eq. 3.

$$\varphi_{a_s} : \mathcal{D} \to \vartheta(a_s) \subseteq \mathbb{R} \tag{3}$$

$\vartheta(a_s)$ is the set of all the possible values for the attribute a_s. $\varphi_{a_s}(d_i)$ is then used to assign actual values to the device d_i for the attribute a_s.

Indeed, the core of the entire approach consists of the definition of a domain-aware metric, quantifying the distance between two events in terms of process instance's attributes, such as time frame or location of the failure in the infrastructure. It is possible to define:

$$\mathbf{m} : E \times E \to \mathbb{R} \,|\, (e_i, e_j) \to m_{i,j} \tag{4}$$

where the $\mathbf{m}$ function is used to build the distance matrix M, and $m_{i,j} = \mathbf{m}(e_i, e_j)$ for all $i, j \in \{0, \ldots, n\}$. It is important to underline that despite being very general, this formalization allows the definition of metrics that should quantify the crucial aspect of the specific application domain and more in particular the case study. Indeed, it is possible to model some aspects strictly dependent on domain knowledge, providing more consciousness of the context. One of the metrics that can be generally adopted in all the applications is the *time metric*: given two events e_i and e_j where their timestamps are, respectively, $\mathcal{T}(e_i) = t_i, \mathcal{T}(e_i) = t_j \in \mathbb{R}$ then it is defined as:

$$m_{i,j} = |t_i - t_j| \tag{5}$$

and it is easy to derive the properties of symmetry and the zero diagonal of M. It should be generically adopted starting from the assumption that the events that belong to the same time frame should be part of the same process instance. However, this only assumption could not satisfy all the requirements of the problem; for example, two separated components of a CPS could fail at the same time for completely different reasons, generating events belonging to two different process instances, but this metric is not able to detect this distinction by itself. A possible solution, could be to adopt more than one metric to evaluate different aspects of the cases and quantify their distances under different points of view (i.e. lexical, topological..). Once a problem is formalized, one or more matrices are generated and now every event is described in terms of distances by the others events and identified by a row in every matrix. These will be the input for the following operations. For the sake of simplicity, here we consider only one matrix M, but in the following section, we will present the implementation taking into account two different matrices obtained from two metrics.

The first operation we propose consists of using the proximity relations in M to represent the events as points on q-dimensional Cartesian axis. We utilize the Multidimensional Scaling (MDS) algorithm [6], which takes as input a number of components q, and the matrix of distances M whose dimensions are $n \times n$, with $q \ll n$. The MDS algorithm calculates the coordinates of the points in an $\mathbb{R}^q$ space by minimizing a loss function that ensures the preservation of distances between objects as much as possible. The aim of this operation is twofold: it supports visualization of the events where the distances between points in the scatter plot correspond to the dissimilarities between the original objects, allowing the domain expert to understand the proximity relation among events; it supports the use of centroid-based clustering methods for the partitioning of events into homogeneous groups. It is important to underline that in case of using more than one metric, the MDS has to be performed on each resulting matrix, obtaining q features for every transformed matrix and finally using all these features as input for the following steps, as reported in the following section.

Every event is then identified by q coordinates in a $\mathbb{R}^q$ space:

$$\mathbf{MDS} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times q} \mid M \rightarrow \mathbf{MDS}(M) \tag{6}$$

Let $\mathcal{B} = \mathbf{MDS}(M)$ where $b_{i,j} \in \mathcal{B}$. We can then assign a tuple of q values to each event, as in Eq. 7.

$$\psi : E \rightarrow \mathbb{R}^q \ , \ e_i \rightarrow \psi(e_i) = (b_{i,1}, b_{i,2}, \ldots, b_{i,q}) \tag{7}$$

The dataset is now composed of a set of n tuples $S = \{\psi(e_1), \ldots, \psi(e_n)\}$ where $\psi(e_i) \in \mathbb{R}^q$. This reduced dataset serves as input to a clustering algorithm that partitions the events into homogeneous groups, evaluating the distances between events described by the q attributes provided by the previous MDS step. K-means is chosen as the method for this clustering phase [12] because it provides, as output, the partition of events and a set of centroids. The latter are a useful synthesis of the data as they describe the average behavior of events in each cluster. K-means requires as input the dataset and the number of clusters k we

want to obtain, the output is a n dimensional vector of the label, one for each row of the input data set, and the set of cluster centroids. Since K-means optimizes a within-cluster heterogeneity criterion, the value of k is typically determined using rules that evaluate within-cluster and/or between-cluster variability for various values of k. Among these rules, the most common include the Elbow method [18], the Average Silhouette method [16], and the Gap Statistic method [19]. Eq. 8 defines the function:

$$\mathbf{KM} : \mathbb{R}^{n \times q} \times \mathbb{N} \to \mathbb{R}^n \ , \ (S, k) \to \mathbf{KM}(S, k) = (l_1, l_2, \ldots, l_n) \tag{8}$$

The labels l_i represent for each event the values of the *CaseID*: the dataset can now be processed by PM techniques.

$$\mathcal{C} : E \to \mathbb{R}^{q+1} \ , \ e_i \to \mathcal{C}(e_i) = (\psi(e_i), l_i) \tag{9}$$

4 Case study

The reliability of the proposed methodology for detecting correctly *CaseID* from an unlabelled dataset was evaluated in a case study involving cyber-physical infrastructure activities. In detail, a monitoring infrastructure is analysed with the support of Gematica[3] company, which has developed solid expertise in complex communication systems and Information Technology (IT) infrastructure management solutions in heterogeneous domains — e.g, railway, automotive, building management. Gematica has realized a test bed in their laboratories where the data used in this paper has been collected.

In particular, by modelling a railway infrastructure for a simple plant, data coming from trains, stations, waysides, and a control centre were collected into a monitoring centre. In this simulated test bed, redundant network links interconnect the different devices, which exchange signals with each other and send alerts to the centre by raising events. All the events are classified by a growing severity: from Information (associated to the severity code 5) to High (associated to the severity code 1). These levels are also mapped on three kinds of triggering events: (1) *fault events*, which are associated with a severity code from 1 to 3; (2) *resolving events*, whose severity code is equal to 4, which logs the clearing of an error situation by the end of a maintenance action; and *information events*, associated to severity code 5, which is limited to reporting non-fault-related events worth to be logged. The dataset extracted comprises several rows, every of which represents an event with its details: event ID, timestamp, event description, severity, IP of the device involved, and three attributes that topologically locate the event within the infrastructure, as shown in Table 1.

In our preliminary tests, 56 fault events with high-level severity and 106 fault events with medium-level severity were considered. Each event involves a device that is configured in a specific system topology, hierarchically structured

[3] https://gematica.com/

Table 1: An excerpt of the dataset.

id	timestamp_event	description_event	severity_event	Level1	Level2	Level3	device_id	device_ip_address
320897	2023-05-09 11:47:59+02:00	Link Down FastEthernet0/5	2	1	342	538	3844	192.168.231.253
320898	2023-05-09 11:48:08+02:00	Device Down	2	1	342	538	3846	192.168.231.120
320903	2023-05-09 12:10:13+02:00	Link Down	2	2	345	542	3862	192.168.220.253

under three different levels, that specify where the device is located in the overall infrastructure.

The *CaseID* to infer represents the identifier for all the fault events correlated and generated from the same cause in a way that they could be assigned to the same failure process. Therefore, we have: $n = 162$ events (E); $m = 19$ devices ($\mathcal{D}$); and $k = 21$ attributes ($\mathcal{A}$). In detail, from the set of attributes, we selected those useful in our methodology and PM application, as indicated in Table 1: the topological features *Level1*, *Level2*, *Level3*, *IP* and *time*, i.e. timestamp converted in second, were selected.

Successively, the metrics, necessary to quantify the distance between two events, were calculated. The proposed methodology is a very general approach that allows applying this formalism to different case studies in multiple domains, giving high flexibility to the technique. The definition and calculation of these metrics were suggested by the guidelines of domain experts. In our context, we defined the metric following the approach proposed in [10]: we started from the assumption that faults caused by the same root event should be "near" in terms of time and semantics. The temporal distance was calculated as stated in Eq. 5, as it is reasonable to assume that two temporally "close" events are related, but this assumption is not sufficient as stated in the previous section 3. Therefore, another metric was added: lexical distance. This metric is strictly dependent on the context information because is defined by the functional and topological dependencies among events related to the same system failure, suggested by the knowledge of domain experts. The lexical distance was defined as follows: let be subset $\mathcal{A}' = \{$*Level1, Level2, Level3, IP-Group*$\} \subseteq \mathcal{A}$, it is possible to use the function in Eq. 3 in the one in Eq. 10, for assigning to each device the values of the attributes selected for the metric, such as:

$$\begin{gathered} \mathcal{L} : \mathcal{D} \to \mathbb{R}^4 \\ d_j \to \mathcal{L}(d_j) = (\varphi_{Level1}(d_j), \varphi_{Level2}(d_j), \varphi_{Level3}(d_j), \varphi_{IP-Group}(d_j)) \end{gathered} \tag{10}$$

According to the Eq. 4 for all $i, j \in \{1, \ldots, n\}$, we define

$$m_{i,j} = d(\mathcal{L}(\mathcal{R}(e_i)), \mathcal{L}(\mathcal{R}(e_j))) \tag{11}$$

where the function is computed as in Fig. 4. The lexical distance can vary from 0 (i.e., two events involving devices in the same IP-Group) to 4 (two events occurred in different settings of the infrastructure).

After the metrics definition, it is possible to calculate the two distance matrices M_t and M_l, in which every event is described by a row, in terms of distances — respectively temporal and lexical — from the others. Starting from these

non-euclidean distances, it is possible to assign to the events, coordinates in a $\mathbb{R}^q$ space with $q \ll n$ whose Cartesian distances match that stored in the matrix, as stated in the Eq. 7. To obtain this, the Eq. 6 is applied to the matrix M_t by choosing $q_t = 5$ and to the matrix M_l with $q_l = 2$, obtaining acceptable results, in terms of stress. Now, it is possible to identify every event with $q = q_t + q_l = 7$ coordinates that locate itself in space and in time. After that, K-Means is performed. The choice of the number of clusters has been performed according to the Elbow rule applied to the Within-Cluster Sum of Squares for $\mathbf{k} = 2, \ldots, 40$. Consistently with this criterion, we set $\mathbf{k} = 27$. *Scikit-learn* 1.2.2[4] is used for the K-Means algorithm.

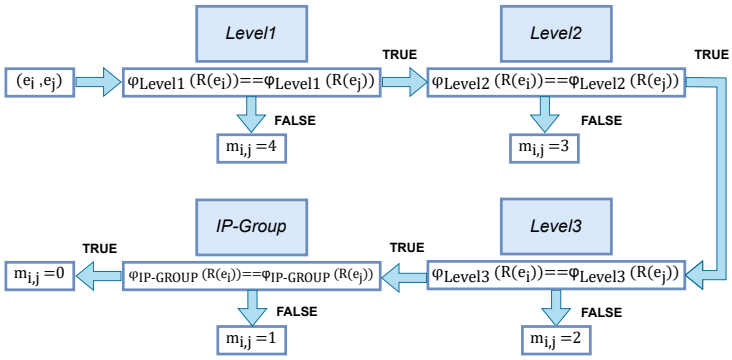

Fig. 4: Definition of the lexical distance between two events.

In order to assess the effectiveness of the proposed methodology and its ability to accurately detect *CaseIDs*, an evaluation of its reliability was conducted. Since the faults were simulated we were aware, supported by domain experts, of the interconnection among the components so we can exactly reconstruct the sequence of events oin each process instance, labelling and comparing the cases with that inferred by the methodology. The performance of clustering was evaluated in terms Silhouette coefficient [1] which is an internal clustering validity index that evaluates cluster compactness and separability, and Rand Index, Adjusted Rand Index, Precision and Recall, which are external clustering validity indexes based on comparing the obtained partition with the a-priori one. Precision and recall were calculated, as proposed in [2], by defining True Positive (TP), False Positive (FP), False Negative (FN) as follow:

- TP is the number of events in which the *CaseID* predicted matches with the true label indicated by the experts;
- FP is the number of events in which the *CaseID* predicted does not match with the true label indicated by the experts because the event should belong to another case;

[4] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

– FN is the number of events in which the *CaseID* does not exist in the labelled log obtained applying the proposed approach but exists in the one labelled by the experts.

The results are reported in the table 2. These show the reliability of the proposed methodology, achieving to detect correctly *CaseID* from an unlabelled event log with a precision of about 81%.

Table 2: Evaluation metrics obtained with the proposed methodology.

Rand Index	Adjusted Rand Index	Silhouette	Precision	Recall
0.973	0.649	0.738	81.0 %	88.1 %

5 Conclusions

In recent years, the use of PM in day-to-day business process management has increased significantly. The present paper moves from the necessity to improve the case detection in the unlabelled event log, which is a precondition to the application of PM algorithms.

This work proposes an approach based on the analysis of temporal and semantic features, which are considered to aggregate faulty events. The K-Means clustering technique is used to group together similar event log entries. The approach has been tested on a railway infrastructure monitoring system, equipped in the laboratories of Gematica company.

The proposed methodology was tested by evaluating unlabelled event logs of the infrastructure monitoring system, achieving the expected results. It is important to underline, that in this preliminary phase, simulated data were adopted in experiments. In prospect, we will explore the performance of the proposed methodology by using real data with a more homogeneous distribution in time and in the types of faults involved. Moreover, in a real context, the metrics calculated could be improved by considering other context information, such as, for example, functional and non-functional dependence between devices, improving the expressive power of such methodology.

Acknowledgements The research has been supported by the DARWINIST project, funded by Universitá della Campania "L. Vanvitelli",
D.R. 834 del 30/09/2022 and by COmplexy SYstem MAintenance (COSYMA) project, program "Fabbrica Intelligente" – MISE, #B36G21000050005.

The work of Roberta De Fazio is granted by PON Ricerca e Innovazione 2014/2020 MUR — Ministero dell'Università e della Ricerca (Italy) — with the PhD program XXXVII cycle D.M. N.1061 "Dottorati e contratti di ricerca su tematiche dell'Innovazione".

The work of Laura Verde is granted by the "Predictive Maintenance Multidominio (Multidomain predictive maintenance)" project, PON "Ricerca e Innovazione" 2014-2020, Asse IV "Istruzione e ricerca per il recupero"-Azione IV.4- "Dottorati e contratti di ricerca su tematiche dell'innovazione" programme CUP: B61B21005470007.

References

1. Al-Mhairat, A.M., Alabbadi, R., Shaban, R., AlQudah, A.: Performance evaluation of clustering algorithms (2019)
2. Bayomie, D., Awad, A., Ezat, E.: Correlating unlabeled events from cyclic business processes execution. LNCS **9694**, 274 – 289 (2016)
3. Bayomie, D., Di Ciccio, C., La Rosa, M., Mendling, J.: A probabilistic approach to event-case correlation for process mining. vol. LNCS 11788, p. 136 – 152 (2019)
4. Bayomie, D., Revoredo, K., Di Ciccio, C., Mendling, J.: Improving accuracy and explainability in event-case correlation via rule mining. In: 2022 4th International Conference on Process Mining (ICPM). pp. 24–31 (2022)
5. Burattin, A., Vigo, R.: A framework for semi-automated process instance discovery from decorative attributes. In: 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM). pp. 176–183 (2011)
6. Carroll, J.D., Arabie, P.: Multidimensional scaling. Measurement, judgment and decision making pp. 179–250 (1998)
7. Emamjome, F.F., Andrews, R., ter Hofstede, A.H., Reijers, H.A.: Alohomora: Unlocking data quality causes through event log context. In: European Conference on Information Systems (2020)
8. Ferreira, D., Gillblad, D.: Discovering process models from unlabelled event logs. In: "Business Process Management". vol. 5701, pp. 143–158 (09 2009)
9. Garcia, C.d.S., al.: Process mining techniques and applications – a systematic mapping study. Expert Systems with Applications **133**, 260 – 295 (2019)
10. Gayo-Avello, D.: A survey on session detection methods in query logs and a proposal for future evaluation. Information Sciences **179**(12), 1822–1843 (2009)
11. Lichtenstein, T., Bano, D., Weske, M.: Attribute-driven case notion discovery for unlabeled event logs. In: Business Process Management WS. Springer (2022)
12. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means clustering algorithm. Pattern recognition **36**(2), 451–461 (2003)
13. Marin-Castro, H.M., Tello-Leal, E.: Event log preprocessing for process mining: A review. Applied Sciences **11**(22) (2021)
14. Myers, D., Suriadi, S., Radke, K., Foo, E.: Anomaly detection for industrial control systems using process mining. Computers & Security **78**, 103–125 (2018)
15. Pourmirza, S., Dijkman, R., Grefen, P.: Correlation mining: Mining process orchestrations without case identifiers. LNCS **9435**, 237 – 252 (2015)
16. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987). https://doi.org/https://doi.org/10.1016/0377-0427(87)90125-7, https://www.sciencedirect.com/science/article/pii/0377042787901257
17. Suriadi, S., Andrews, R., ter Hofstede, A., Wynn, M.: Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. Information Systems **64**, 132–150 (2017)
18. Thorndike, R.L.: Who belongs in the family? Psychometrika **18**(4), 267–276 (1953)
19. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **63** (2001)
20. Van Der Aalst, W., et al.: Process mining manifesto. Lecture Notes in Business Information Processing **99 LNBIP**(PART 1), 169 – 194 (2012)