A Contextual Approach to Detecting Synonymous and Polluted Activity Labels in Process Event Logs

Sareh Sadeghianasl, Arthur H.M. ter Hofstede, Moe T. Wynn, and Suriadi Suriadi

Queensland University of Technology, Brisbane, Australia
sareh.sadeghianasl@hdr.qut.edu.au
{a.terhofstede, m.wynn, s.suriadi}@qut.edu.au

Abstract. Process mining, as a well-established research area, uses algorithms for process-oriented data analysis. Similar to other types of data analysis, the existence of quality issues in input data will lead to unreliable analysis results (*garbage in - garbage out*). An important input for process mining is an event log which is a record of events related to a business process as it is performed through the use of an information system. While addressing quality issues in event logs is necessary, it is usually an ad-hoc and tiresome task. In this paper, we propose an automatic approach for detecting two types of data quality issues related to activities, both critical for the success of process mining studies: synonymous labels (same semantics with different syntax) and polluted labels (same semantics and same label structures). We propose the use of activity context, i.e. control flow, resource, time, and data attributes to detect semantically identical activity labels. We have implemented our approach and validated it using real-life logs from two hospitals and an insurance company, and have achieved promising results in detecting frequent imperfect activity labels.

Keywords: Data quality · Process event log · Activity Label.

1 Introduction

Process mining, as a relatively new research area, uses information in event logs, to discover, control and improve processes [3]. An event log is a record of events related to a business process, executed via an information system. Process mining has delivered promising results in discovering actual behaviors (i.e. *process discovery*), checking conformance of processes to organizational rules, analyzing process performance, and suggesting process improvements.

The starting point of process mining is an event log which contains data related to a *process*. A process is a series of tasks that are performed in a specific order to achieve a specific result. Each execution of a process is called a *case*. The records of the execution of these tasks are stored in a log as so-called *events*. An event can be described by the type of *activity* (i.e. "a well-defined step in the

process" [2]) that is executed as part of a case at a particular point in time (as represented by a *timestamp*). There may also be some supporting attributes for events e.g. *resources* (i.e. originator of activity), *transaction type* (e.g. start or complete), and *data attributes* (e.g. illness of a patient).

Unfortunately, in practice, event logs suffer from data quality issues, e.g. missing events (attributes) [6], imprecise timestamps [7], and duplicate events [22]. Analyzing an event log with plenty of data quality issues is not advisable as it will most likely lead to unreliable process mining results (*garbage in - garbage out*). Suriadi et al. [29] introduced 11 event log *imperfection patterns* as a systematic view of data quality issues in event logs, thus providing a pathway for subsequent (semi-)automated approaches to repair event logs. These patterns can be further categorized into timestamp, case, and activity label imperfection patterns. Existing approaches to detecting and repairing those event log imperfection patterns mainly focus on timestamp imperfection patterns [10, 13, 21]. Although activity label quality issues have been addressed at the process model level [1, 5, 12, 15] and at the event log level [8, 16, 22, 31], as will be discussed in Section 2, they are not well-suited to automatically detect synonymous and polluted activity labels in event logs.

In this paper, we focus on two activity label imperfection patterns, i.e. synonymous and polluted labels, both occurring in real-life logs, as reviewed by Suriadi et al. [29], and both critical for the success of process mining analysis. Synonymous activity labels are those that are semantically identical, but syntactically different, e.g. *"Dr. seen"* and *"Visited by doctor"*. They are most common when an event log is derived from multiple systems that use different names for the same concept. Polluted labels also refer to the same semantics but they have similar label structures where the differences further qualify their meaning, e.g. *"Notification of Loss XXXX Incident No. xxxx"*, *"Notification of Loss YYYY Incident No. yyyy"*, and *"Notification of Loss ZZZZ Incident No. zzzz"* as seen in an insurance log [29]. These labels are most common when the system records events from a free-text input providing an initial suggestion.

The existence of synonymous and polluted activity labels in event logs leads to unnecessarily large and confusing discovered process models which include behaviors that should have been merged. These patterns may also negatively affect performance analysis results since two or more activities that should be recognized as the same are considered as separate. Approaches to resolving such misclassifications mainly focus on activity labels and use domain ontologies or experts [8,16], however *our approach abstracts from labels and uses context information instead*, i.e. control flow, resource, time, and data to automatically detect semantically identical activity labels. Although using ontologies can be helpful, they might not always be available particularly when a specialized ontology is required for a specific domain, e.g. the medical domain where different names are used for the same test. Using label similarity metrics can also help in detecting synonymous and, mainly, polluted labels, but they can sometimes be misleading e.g. activities *"Admission IC"* and *"Admission NC"* regarding admission of a patient to intensive care and normal care in a hospital respectively [23].

Putting process mining into context leads to more qualified analysis results [4, 27]. In this paper, we provide a framework for activity context and use it, through some example distance measures, to investigate semantic distance between activities. We distinguished four perspectives (although flexible) for activity context: control flow, resource, time and data. In each of these perspectives, we have defined some principles and some example activity distance measures. We then use these measures to cluster activities and aggregate the results using configurable dimension and measure weights. We also guide the user by suggesting the best weights. Finally, we detect synonymous and polluted activity labels using a flexible context similarity threshold.

We have implemented and evaluated our approach using real-life logs from two hospitals and an insurance company. Our results show the approach detects frequent imperfect activity labels efficiently, e.g. it achieves an F-score of 1 in the Sepsis log. Furthermore, considering *multiple context dimensions* seems to be more helpful than looking at a single dimension as it might be misleading, e.g. control flow dimension when a flower model is discovered from the log.

This paper is organized as follows. Section 2 discusses the related work. In Section 3 we consider event log basics and formal notations. In Section 4 we describe a context framework for activities and distance measures for different context dimensions. Section 5 contains our approach to detecting synonymous and polluted labels. In Section 6 we present our evaluations using real-life logs and in Section 7 we summarize our findings and suggest future work.

2 Related Work

The subject of data quality of process event logs was initially proposed in the Process Mining Manifesto [3] where a 1 to 5 star rating was defined for the quality of an event log, where logs rated as 3, 4 or 5-star are ready for applying process mining analysis, whereas logs rated as 1 and 2-star are not. There are some works, e.g. [7, 20, 24, 32], that provide a framework for event log quality, although they do not provide any method to detect such quality issues in logs. Bose et al. [6, 7] identify four classes of process event log quality issues: missing data, incorrect data, imprecise data, irrelevant data, and how they may appear in different elements of an event log including cases, events, activity names, timestamps, resources, etc. Synonymous and polluted activity labels are manifestations of imprecise activity names in the classification of Bose et al. [7], where multiple names for (semantically) the same activity yields ambiguity in an event log. Mans et al. [24] define data quality for event logs as a two-dimensional spectrum, where the first dimension considers the abstraction level of events and the second considers the accuracy of timestamps, i.e. their granularity, currency, and correctness. Lu and Fahland [20] distinguish three concepts in event logs whose quality is important, i.e. events, ordering of events and labels of events. Each concept has two aspects: data (covering intrinsic qualities of event data) and analysis (determining whether a behavior is repeating across many cases). Synonymous and polluted labels threaten the quality of labels of events.

Suriadi et al. [29] classify common quality issues found in process event logs or raised when preparing event logs from raw data sources, as 11 generic imperfection patterns in logs thus paving the way for subsequent automatic approaches to cleaning event logs. These patterns can be further categorized into timestamp, case, and activity label imperfection patterns. Existing approaches mainly focus on detecting and repairing timestamp imperfection patterns in event logs [10, 13, 21], however, there are some methods, e.g. [22, 31], that could be used for detecting and repairing duplicate tasks, i.e. *homonymous* labels [29], in event logs where multiple activities with different semantics have the same name. The notion of *label similarity* has been extensively discussed in the field of process model matching and similarity, which is reviewed in [5], however they mainly use syntactic and semantic similarity measures (using ontologies) [15] or control flow, duration and data attributes of activities [1, 12] to match labels, ignoring other context dimensions, e.g. resources and timing patterns, that are typically found in event logs.

Synonymous activity labels are viewed as *semantic noise* by Gunther [14] who describes two possible types of noise in event logs: *syntactic noise* that occurs due to errors in logging (e.g. missing head or tail of traces) and *semantic noise* that is introduced to a log on purpose due to e.g. *customizations* of the same process, where for instance a company localizes activity names for each country. Gunther [14] indicates that customization noise can be resolved via *semantic pre-processing*, e.g. using an ontology. In the area of semantic process mining, ontologies are widely used to annotate tasks with the concept they represent [9, 26]. These annotations are further used to discover a conceptual process model [8] or to revise the vocabulary of process model elements [16]. However, our approach detects labels with the same semantics using context information provided in a log, i.e. control flow, resource, time and data, and abstracts from activity labels (hence e.g. it does not depend on access to an ontology).

3 Preliminaries

An event log L is formally defined as $L = (\mathcal{E}, \mathcal{A}, \mathcal{V}, \mathcal{T}, AN, \#, \mathscr{L})$, where $\mathcal{E}$ is the set of event identifiers, $\mathcal{A}$ is the set of activities, $\mathcal{V}$ is the set of values, $\mathcal{T}$ is the set of timestamps, AN is the set of attributes names, $\# : \mathcal{E} \rightarrow (AN \rightarrow \mathcal{V})$ gets the value of attribute $n \in AN$ recorded for an event $e \in \mathcal{E}$, i.e. $\#_n(e) \in \mathcal{V}$, and $\mathscr{L} \subseteq \mathcal{E}^*$ is the set of all traces over $\mathcal{E}$. A trace $\sigma \in \mathscr{L}$ is a sequence of events of a process instance such that each event occurs only once.

For any event $e \in \mathcal{E}$, $\#_{ac}(e)$, $\#_{ti}(e)$, $\#_{re}(e)$, $\#_{trans}(e)$, represent activity, timestamp, resource, and life-cycle transaction of event e. These attributes are referred to as *standard attributes*, among which activity and timestamp are mandatory. For any activity $a \in \mathcal{A}$, $AN_a \subset AN$ is the set of attribute names, excluding standard attributes, for events executing activity a, i.e. $AN_a = \{n \in AN \mid \#_{ac}(e) = a \wedge n \notin \{ac, re, ti, trans\} \wedge \#_n(e) \neq \perp\}$. We define $\mathcal{E}_{com} \subseteq \mathcal{E}$ as the set of *complete events*, i.e. $\mathcal{E}_{com} = \{e \in \mathcal{E} \mid \#_{trans}(e) = complete\}$. Similarly, $\mathcal{E}_{st} \subseteq \mathcal{E}$ consists of all *start events*, i.e. $\mathcal{E}_{st} = \{e \in \mathcal{E} \mid \#_{trans}(e) = start\}$.

A *directly before* relation $\sqsubset$ is derived over events, where for any $e_1, e_2 \in \mathcal{E}$, $e_1 \sqsubset e_2$ holds if and only if there exists a trace $\sigma \in \mathscr{L}$ such that $\sigma(i) = e_1$ and $\sigma(i+1) = e2$, $1 \leqslant i < \mid \sigma \mid$. Similarly, an *indirectly before* relation $\sqsubseteq$ is defined over events where for any $e_1, e_2 \in \mathcal{E}$, $e_1 \sqsubseteq e_2$ holds if and only if there exists a trace $\sigma \in \mathscr{L}$ such that $\sigma(i) = e_1$ and $\sigma(j) = e2$, $1 \leqslant i < \mid \sigma \mid$, $i < j \leqslant \mid \sigma \mid$.

For any activity $a \in \mathcal{A}$, we define its *events set* $\mathcal{E}_a = \{e \in \mathcal{E} \mid \#_{ac}(e) = a\}$ as the set of all events e that are executing activity a. The frequency $F : \mathcal{A} \rightarrow N$ for any activity $a \in \mathcal{A}$ is the size of its events set, i.e. $F(a) = \left|\mathcal{E}_a\right|$.

In the rest of this paper, we use the shorthands PDF and CDF to refer to probability density function and cumulative probability distribution function respectively. Here we also formulate the Manhattan distance measure [11] as it is used in defining distance measures in the rest of this paper. We use the Manhattan distance measure [11] because it integrates all the absolute differences between two PDFs, related to two activities, over the same domain and is suitable for comparing activities based on each of their attributes, e.g. resource, time, and data values. The normalized Manhattan distance between any two PDFs $p = (p_1, p_2, \cdots, p_n)$ and $q = (q_1, q_2, \cdots, q_n)$ is computed as[1]: $M(p, q) = \frac{\Sigma_{i=1}^{n} |p_i - q_i|}{2}$.

4 Activity Context Framework

The context in which activities are executed can provide meaningful insights into more qualified process mining analysis [4, 27]. Our approach uses the activity context to identify synonymous and polluted activity label candidates. Figure 1 shows our activity context framework which includes four perspectives: (1) *The control flow perspective* (i.e. the ordering of activities), (2) *The resource perspective* (i.e. people, roles, or devices performing activities), (3) *The temporal perspective* (i.e. timing of activities), and (4) *The data perspective* (i.e. data attributes of activities). The principle is that the similarity of activities based on each perspective might be an indication of semantically identical activities even if they have different labels. As depicted in Figure 1, context dimensions are assumed to be independent of each other, i.e. the absence or internal changes of one perspective does not influence the validity of others. In this section, we show some example distance measures for each dimension. Although one can replace them with other measures or even add other dimensions, the principles remain. The following sections discuss each perspective in terms of its principles and our example distance measure(s).

4.1 Control Flow Perspective

Principle. The control flow perspective is concerned with behavioral (ordering) relations between activities in an event log. Similar control flow contexts might be an indication of identical activities i.e. activities that are following similar

[1] The Manhattan distance between any two PDFs p and q lies in the interval $[0, 2]$, because in the best case, p and q are identical, then $M(p, q) = 0$, and in the worst case $\exists i, j \mid 1 \leqslant i, j \leqslant n, i \neq j$ such that $p_i = 1$ and $q_j = 1$, then $M(p, q) = 2$.

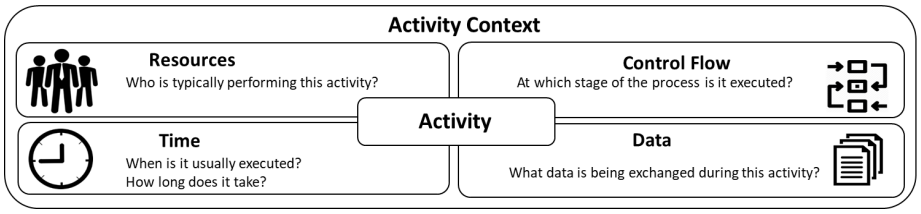

Fig. 1. Activity Context Framework

kind of work and are followed by similar kind of work, are more likely to be semantically identical than those with a different control flow context.

Measure. To formulate the control flow context, we use direct ordering relations between activities, as defined by van der Aalst [2], based on a primary "directly follows" relation. For any two activities $a, b \in \mathcal{A}$ executed in log L:

- $a >_L b$ (*directly follows* relation) if and only if $\exists e_1 \in \mathcal{E}_a, \exists e_2 \in \mathcal{E}_b \mid e1 \sqsubset e_2$.
- $a \rightarrow b$ (*"Causes"* relation) if and only if $a >_L b$ and $b \not> _L a$.
- $a \leftarrow b$ (*"Caused by"* relation) if and only if $b >_L a$ and $a \not> _L b$.
- $a \# b$ (*"Exclusive"* relation) if and only if $a \not> _L b$ and $b \not> _L a$.
- $a \parallel b$ (*"Concurrent"* relation) if and only if $a >_L b$ and $b >_L a$.

For any pair of activities $a, b \in \mathcal{A}$, $a \rightarrow b$, or $b \rightarrow a$, or $a \# b$, or $a \parallel b$, i.e. exactly one of these relations holds. Therefore, we can capture ordering relations of a log in a *footprint* [2] matrix $F = [f_{i,j}]$, $1 \leqslant i, j \leqslant \mid \mathcal{A} \mid$, where $f_{i,j} \in \{\rightarrow, \leftarrow, \#, \parallel\}$ specifies ordering relations between any two activities $a, b \in \mathcal{A}$ such that $i = index(a)$, $j = index(b)$ and $index : \mathcal{A} \rightarrow \{1, 2, \cdots, |\mathcal{A}|\}$ is a function that assigns an index to each activity and forms a bijection. Equation 1 defines the control flow distance measure $D_{cf} : \mathcal{A} \times \mathcal{A} \rightarrow [0, 1]$ between any two activities $a, b \in \mathcal{A}$:

$$D_{cf}(a, b) = \frac{\left|\{(i, j) \mid 1 \leqslant j \leqslant n \wedge (f_{i,j} \neq \# \vee f_{k,j} \neq \#) \wedge f_{i,j} \neq f_{k,j}\}\right|}{\left|\{(i, j) \mid 1 \leqslant j \leqslant n \wedge (f_{i,j} \neq \# \vee f_{k,j} \neq \#)\}\right|}, \quad (1)$$

where $n = \mid \mathcal{A} \mid$, $i = index(a)$, $k = index(b)$ and we calculate the ratio of the different cells in rows i and k to the total number of cells in rows i and k of each matrix. We exclude *exclusive* relations $\#$, because the fact that two activities are excluded from many other activities does not mean that they are identical.

4.2 Resource perspective

Principle. The resource perspective focuses on people, devices, or software involved in executing activities. Activities that are usually performed by similar groups of resources (roles) are more likely to be the same rather than activities executed by different groups of resources (roles).

Measure. In order to formulate resources executing an activity, we use PDFs. Let $\mathcal{R}$ be the set of all people or devices that originated at least one event $e \in \mathcal{E}$ in log L, i.e. $\mathcal{R} = \{\#_{res}(e) \mid e \in \mathcal{E}\}$. We define *null-resource* activities $\mathcal{A}_{R\perp}$ as the

set of all activities where none of the associated events have a resource attribute, i.e. $\mathcal{A}_{R\perp} = \{a \in \mathcal{A} \mid \nexists e \in \mathcal{E}_a[\#_{res}(e) \neq \perp]\}$. For any activity $a \in \mathcal{A}$ we define a multi-set consisting of its events' resources as $\mathcal{R}_a = \{(r, |\mathcal{E}_{a,r}|) \mid r \in \mathcal{R}\}$, where $\mathcal{E}_{a,r} = \{e \in \mathcal{E}_a \mid r = \#_{res}(e)\}$ is the set of all events in which resource r executes activity a. The resource distance measure $D_{re} : \mathcal{A} \times \mathcal{A} \to [0,1] \cup \{\perp\}$ between any pair of activities $a, b \in \mathcal{A}$ is defined as:

$$D_{re}(a,b) = \begin{cases} M(PR_a, PR_b) & \text{if } a \notin \mathcal{A}_{R\perp} \wedge b \notin \mathcal{A}_{R\perp} \\ 1 & \text{if } a \notin \mathcal{A}_{R\perp} \oplus b \notin \mathcal{A}_{R\perp} \\ \perp & \text{otherwise,} \end{cases} \tag{2}$$

where $PR_a : \mathcal{R} \to [0,1]$ is a PDF estimated from multi-set $\mathcal{R}_a$. If activities a and b have resource information in the log, then their resource distance is computed as the normalized Manhattan distance between their corresponding PDFs, i.e. PR_a and PR_b. Otherwise if only one of the activities a and b has resource information, then they are assumed to be far apart $D_{re}(a,b) = 1$. Otherwise, if no resource information is available for neither a nor b, then no information is gained, i.e. $D_{re}(a,b) = \perp$.

4.3 Temporal perspective

Principle. Timing information of activities, e.g. duration, waiting time, or point of time in which they are usually executed[2], is another perspective that can indicate the similarity or difference between types of work that have been performed.

Measure. We define two measures: one concerning the duration of activities and the other one concerning the point of time at which they are executed.

Life-cycle Duration

We measure the distance between activities based on duration PDFs. We consider at most two life-cycle types for any event $e \in \mathcal{E}$ appearing in event log L: start and complete. Events with no transaction attribute (atomic events), i.e. $\#_{trans}(e) = \perp$, are assumed as *complete* events, i.e. $\#_{trans}(e) =$ complete. For any *complete* event $e_c \in \mathcal{E}_{com}$, we define its corresponding start event as $Start(e_c) = e_s \in \mathcal{E}_{st} \mid \#_{ac}(e_s) = \#_{ac}(e_c) \wedge e_s \sqsubseteq e_c \wedge \nexists e'_s \in \mathcal{E}_{st}[e'_s \sqsubseteq e_s \wedge \#_{ac}(e'_s) = \#_{ac}(e_s)] \wedge \nexists e'_c \in \mathcal{E}_{com}[e_s \sqsubseteq e'_c \sqsubseteq e_c \wedge \#_{ac}(e'_c) = \#_{ac}(e_c)]$, i.e. the earliest start event e_s that is not indirectly followed by another complete event e'_c before e_c, all with the same activity name. Event duration $\mathcal{D} : \mathcal{E}_{com} \to N$ for any *complete* event $e_c \in \mathcal{E}_{com}$ is defined as :

$$\mathcal{D}(e_c) = \begin{cases} \#_{ti}(e_c) - \#_{ti}(Start(e_c)) & \text{if } Start(e_c) \neq \perp \\ 0 & \text{otherwise.} \end{cases}$$

With this formalization, we ignore start events where the corresponding complete event is not recorded in the log (i.e. incomplete events). Max event duration

[2] However, it may not be helpful if activities are performed in batch processing mode.

$\mathcal{D}_{max} \in N$, is the duration of the event which takes the longest time compared to other events in the whole log L, i.e. $\mathcal{D}_{max} = max_{e_c \in \mathcal{E}_{com}}(\mathcal{D}(e_c))$. We also define $\mathcal{A}_{at}$ as the set of all atomic activities in the log, i.e. $\mathcal{A}_{at} = \{a \in \mathcal{A} \mid \forall e \in \mathcal{E}_a \cap \mathcal{E}_{com}[\mathcal{D}(e) = 0]\}$. For any activity $a \in \mathcal{A}$ activity duration is defined as a multi-set $\mathcal{D}_a = \{(d, |\mathcal{E}_{a,d}|) \mid d \in [0, \mathcal{D}_{max}]\}$ where $\mathcal{E}_{a,d} = \{e \in \mathcal{E}_a \cap \mathcal{E}_{com} \mid \mathcal{D}(e) \div \beta = d\}$ and $\beta > 0$ is a bin width used for binning activity durations (as we are not interested in tiny differences in durations). The duration distance metric $D_{du} : \mathcal{A} \times \mathcal{A} \to [0, 1] \cup \{\perp\}$ for any two activities $a, b \in \mathcal{A}$ is defined as:

$$D_{du}(a,b) = \begin{cases} M(PD_a, PD_b) & \text{if } a \notin \mathcal{A}_{at} \wedge b \notin \mathcal{A}_{at} \\ 1 & \text{if } a \notin \mathcal{A}_{at} \oplus b \notin \mathcal{A}_{at} \\ \perp & \text{otherwise,} \end{cases} \tag{3}$$

where $\mathcal{PD}_a : [0, \mathcal{D}_{max}] \to [0, 1]$ is PDF estimated using multi-set $\mathcal{D}_a$. If a and b are both non-atomic, their distance is computed as the normalized Manhattan distance between their corresponding PDFs PD_a and PD_b. If only one of the activities a and b has a duration, then they are assumed to be far away from each other, i.e. $D_{du}(a, b) = 1$. Otherwise if neither a nor b are non-atomic, then we can not judge their duration distance, i.e. $D_{du}(a, b) = \perp$.

Timing Pattern

The timing pattern measures how regularly an activity is executed, e.g. mornings or evenings, every Monday or every day. In order to formulate timing patterns of activities, we use PDFs. For any activity $a \in \mathcal{A}$ and any unit of time $U \in \mathcal{U} = \{h, dw, m\}$, referring to the part of a day[3](h), the day of a week (dw), or the month of a year (m), the activity execution pattern is a multi-set $\mathcal{T}_{U,a} = \{(u, n) \in N \times N \mid n = \big|\{e \in \mathcal{E}_a \mid \pi_U(\#_{ti}(e)) = u\}\big|\}$, where operator $\pi_U(t) : \mathcal{T} \to N$ extracts unit of time U from timestamp $t \in \mathcal{T}$. A complementary multi-set $\mathcal{T}_{U,a'} = \{(u, n) \in N \times N \mid n = \big|\{e \in \mathcal{E} - \mathcal{E}_a \mid \#_{ti}(e) = t \wedge \pi_U(t) = u\}\big|\}$, is defined for any activity $a \in \mathcal{A}$ concerning all activities in the log except a. Function $D_U : \mathcal{A} \times \mathcal{A} \to [0, 1] \cup \{\perp\}$ measures the difference between timing patterns of any pair of activities $a, b \in \mathcal{A}$ based on any unit of time $U \in \mathcal{U}$:

$$D_U(a,b) = \begin{cases} M(PU_a, PU_b) & \text{if } CU_a \not\approx_{KS} CU_{a'} \wedge CU_b \not\approx_{KS} CU_{b'} \\ 1 & \text{if } CU_a \not\approx_{KS} CU_{a'} \oplus CU_b \not\approx_{KS} CU_{b'} \\ \perp & \text{otherwise,} \end{cases}$$

where for $x \in \{a, b\}$, PU_x is PDF estimated from multi-set $\mathcal{T}_{U,x}$, and CU_x, $CU_{x'}$ are CDFs estimated from multi-sets $\mathcal{T}_{U,x}$, $\mathcal{T}_{U,x'}$ respectively. Operator $\not\approx_{KS}$ checks whether two CFDs have statistically significant differences ($\alpha = 0.01$) under the Kolmog0rov-Smirnov test [25] or not. A statistical anomaly of an activity's execution times means that it has a significantly different timing distribution from all other activities in the log. Here we are only interested in

[3] A part of a day is a 4-hours period of a day.

activities that *have* a statistical anomaly, i.e. are executed only at *specific* times. If we see statistical anomalies in both the timing distributions of activities a and b then they need further checks and if they have similar timing patterns, e.g. they are both executed only on Mondays, then they are close. However, if a statistical anomaly is only observed in time CDF for one of the activities a or b, then they are assumed to be far from each other, i.e. their distance is1. Otherwise, if none of the activities a and b have anomalies in their time distributions, then no information is gained (their distance is set to $\perp$).

The overall timing pattern distance $D_{tp} : \mathcal{A} \times \mathcal{A} \to [0, 1] \cup \{\perp\}$ between any pair of activities $a, b \in \mathcal{A}$ is the minimum time distance for different time units U. We use the min function because we are going to use the distance measure D_{tp} for clustering, and we are interested in the smallest time distance between a and b. If such a small distance exists then a and b will be in the same cluster.

$$D_{tp}(a, b) = \min_{U \in \{h, dw, m\}} D_U(a, b) \tag{4}$$

4.4 Data perspective

Principle. Data attributes and their values may also indicate similarity or difference between activities. Activities with different data attributes or, the same data attributes, but different distribution of values, are probably not identical[4].

Measure. In this paper, we focus only on event-level data, although case-level data attributes might also be helpful (they are left as future work). For any activity $a \in \mathcal{A}$ and any of its event data attributes $d \in AN_a$, we define a multi-set $\mathcal{V}_{a,d} = \{(v, |\mathcal{E}_{a,d,v}|) \mid v \in \mathcal{V}\}$, where $\mathcal{E}_{a,d,v} = \{e \in \mathcal{E}_a \mid \#_d(e) = v\}$ is the set of all events of activity a where event data attribute d has value v. The data distance measure $D_{ed} : \mathcal{A} \times \mathcal{A} \to [0, 1] \cup \{\perp\}$ between any pair of activities $a, b \in \mathcal{A}$ is defined as:

$$D_{ed}(a, b) = \begin{cases} \frac{\Sigma_{d_1 \in AN_a \wedge d_2 \in AN_b \wedge d1 = d2} M(Q_{a,d_1}, Q_{b,d_2})}{|AN_a \cap AN_b|} & \text{if } AN_a \cap AN_b \neq \varnothing \\ 1 & \text{if } AN_a \cap AN_b = \varnothing \vee \\ & (AN_a \neq \varnothing \oplus AN_b \neq \varnothing) \\ \perp & \text{otherwise,} \end{cases} \tag{5}$$

where for $i \in \{1, 2\}$, $Q_{a,d_i} : \mathcal{V} \to [0, 1]$, is PDF defined using multi-set $\mathcal{V}_{a,d_i}$. The data distance between activities a and b is the average of the normalized Manhattan distances between the value distribution of data attributes $d_1 \in AN_a$ and $d_2 \in AN_b$ with the same name, i.e. $d_1 = d_2$. If activities a and b have no data attributes in common or only one of them has data attributes, then they are assumed to be far apart, i.e. $D_{ed}(a, b) = 1$. Otherwise, if none of the activities a and b have data attributes, then no information is gained, i.e. $D_{ed}(a, b) = \perp$.

[4] However, this principle may not hold for data attributes that take a wide range of values. One may be able to distinguish such attributes and informative ones via data-aware process mining [18]. The most informative attributes that indicate similarity or difference between activities are probably those involved in decision points.

5 Approach

Activities with similar contexts are candidates for synonymous or polluted labels. Here, context can be any of the aforementioned perspectives i.e. control flow, resource, time and data or any additional dimension. Let Φ be the context dimensions universe and $\mathcal{M} = \{m : \mathcal{A} \times \mathcal{A} \rightarrow [0,1] \cup \{\perp\}\}$ be the set of all context distance measures. Function $mrs : \Phi \rightarrow 2^{\mathcal{M}}$ assigns a subset of measures to each dimension $\varphi \in \Phi$. Weighting functions $W_\varphi : mrs(\varphi) \rightarrow [0,1]$ and $W_\Phi : \Phi \rightarrow [0,1]$ such that, $\Sigma_{m \in mrs(\varphi)} W_\varphi(m) = 1$ and $\Sigma_{\varphi \in \Phi} W_\Phi(\varphi) = 1$, set weights to measures within each dimension and to dimensions in Φ respectively. For any measure $m \in \mathcal{M}$, $G_m = (\mathcal{A}, E_m)$ is an activity distance graph where activities $\mathcal{A}$ are nodes and weighted edges $E_m = \{(a, b, w) \mid a, b \in \mathcal{A} \wedge w = m(a, b)\}$ represent the distance between activities based on m. We cluster activities using the minimum spanning tree (MST) clustering algorithm [30] which can be decomposed into two main steps: (1) computing a minimum spanning tree (MST) for the input activity distance graph, which is solved by the Kruskal algorithm [17], and (2) iteratively creating a new cluster by breaking an edge of MST with the largest weight, until the desired number of clusters is reached. In order to estimate the desired number of clusters, we use Silhouette analysis [28] where the maximum value of the average Silhouette score represents the optimal number of clusters. For any $m \in \mathcal{M}$, applying MST clustering on G_m yields a partition Ω_m and any pair of activities are similar based on measure m if and only if they belong to the same cluster in Ω_m, i.e. $a \sim_m b \equiv \exists \omega \in \Omega_m [\{a, b\} \subseteq \omega]$.

Algorithm 1 presents our approach for detecting synonymous or polluted activity labels in event logs. We initially perform a pre-processing phase (Line 1) to filter out case-level data attributes (i.e. event data attributes that have the same value across a case) and also id-like event data attributes (i.e. those where the number of distinct values in the whole log equals to the total number of events), as they can not give us useful information about the nature of an activity. In Lines 2-5, for each measure of each context dimension, we make a graph and perform MST clustering. For any pair of activities $a, b \in \mathcal{A}$, we compute the weighted average similarity score $avgSim$, (Line 22) which accumulates information about whether or not a and b are similar in each dimension. To do so, for any dimension $\varphi \in \Phi$, we compute $\varphi Sim_{a,b}$ as the weighted average of measures within φ (Line 20). If no information is available for comparing a and b based on distance measure $m \in mrs(\varphi)$, then we exclude this measure by setting its corresponding weight to 0 (Lines 12-13). Furthermore, if all the measures within a dimension have null values, then we ignore that dimension (Line 19). If average context similarity between activities a and b is more than a given threshold θ, then they are detected as synonymous or polluted activity labels (Line 23).

The time complexity of the algorithm depends on the time complexity of computing measures. For our defined context dimensions and measures, i.e. Equations 1-5, we break the algorithm into steps (assume n, m, k, v, r, d are the number of activities, events, data attributes, data values, resources, and binned durations in log L): The pre-processing step (Line 1) is $O(m \times k)$, making the activity dis-

tance graph (Lines 2-5) is $O(n^2 \times (m+n))$[5], $O(m + (n^2 \times r))$[6], $O(m + n^2 \times d)$[7], $O(m + n^2)$[8], and $O((m \times k) + (n^2 \times v))$[9] for control flow, resource, duration, time and data measures respectively, MST clustering is $O(n^3)$ [10], and finally the detection step (Lines 6-24) is $O(n^2)$. Therefore, the overall time complexity of our approach is $O(m \times (r+k) + n^2 \times (n+m+d+v))$.

Algorithm 1: Detect Synonymous and Polluted Activity Labels

Input: Event log $L = (\mathcal{E}, \mathcal{A}, \mathcal{V}, \mathcal{T}, AN, \#, \mathscr{L})$, context universe Φ, weighting functions W_φ and W_Φ, threshold θ
Output: $\mathcal{A}_{imperfect}$

```
1  L ← Preprocess(L) ;
2  foreach φ ∈ Φ do
3      foreach m ∈ mrs(φ) do
4          G_m ← makeGraph(A, m);
5          Ω_m ← MST-Clustering(G_m);
6  foreach a, b ∈ A do
7      foreach φ ∈ Φ do
8          allNull ← true;
9          φSim_{a,b} ← 0;
10         foreach m ∈ mrs(φ) do
11             mSim_{a,b} ← 0;
12             if m(a, b) =⊥ then
13                 W_φ(m) ← 0;
14             else
15                 allNull ← false;
16                 if a ∼_m b then
17                     mSim_{a,b} ← 1 − m(a, b);
18         if allNull then
19             W_Φ(φ) ← 0;
20         else
21             φSim_{a,b} ← Σ_{m∈mrs(φ)} mSim_{a,b} × W_φ(m);
22     avgSim_{a,b} ← Σ_{φ∈Φ} W_Φ(φ) × φSim_{a,b};
23     if avgSim_{a,b} ⩾ θ then
24         A_imperfect ← A_imperfect ∪ {{a, b}};
25 return A_imperfect
```

[5] $O(m + n^2)$ for the footprint matrix and $O(n^3)$ for the distance measure.
[6] $O(m)$ for the resource multi-sets and $O(n^2 \times r)$ for the distance measure.
[7] $O(m)$ for the duration multi-sets and $O(n^2 \times d)$ for the distance measure.
[8] $O(m)$ for the time multi-sets and $O(n^2)$ for the distance measure.
[9] $O(m \times k)$ for the data multi-sets and $O(n^2 \times v)$ for the distance measure.
[10] $O(n^2 \log n)$ for the Kruskal algorithm and $O(n^3)$ for silhouette analysis.

6 Evaluation

We conducted three experiments to evaluate our approach using real-life logs. To perform these experiments, we implemented Algorithm 1 for the aforementioned distance measures[11] in Java and released a plug-in for the ProM framework[12]. Our design for four experiments is illustrated in Figure 2. The first experiment (Figure 2(a)) was aimed at assessing how our approach works with artificial imperfect activity labels. For this, we used a real-life log and generated other logs by incrementally renaming activity labels. These logs were provided as input to our detection approach. The next two experiments (Figure 2(b)) aimed at investigating whether we can get the same performance in detecting real imperfect activity labels in real-life logs. We assessed the number of activity labels that could be detected by our method, by measuring precision, recall, and F-score.

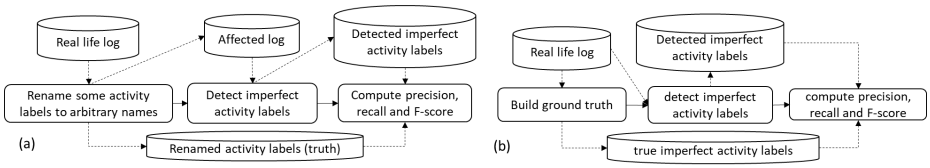

Fig. 2. Experiment setup for (a) artificial and (b) real imperfect activity labels

A summary of characteristics of real-life logs that we used for experiments can be found in Table 1, where the columns identify the number of traces, trace variants, events, activities, resources, event-level attributes, life-cycle, affected activities, affected events, and their percentage. Although none of these logs have activity duration information, we have tested our approach on simulated logs where activity duration is recorded. For the first experiment, we used the Hospital Billing log[13] which contains events related to billing of medical services provided by a Dutch hospital. We chose this log because it has a rich context, e.g. 18 event-level attributes, 1151 resources, and fine granular timestamps[14]. For the second experiment, we used the Sepsis log[15] [23] which contains events relates to the treatment process of sepsis cases from a Dutch hospital. It includes different attributes for events e.g. results of tests and some information from checklists. To the best of our knowledge, this was the only event log in 4TU Data Center that contains multiple activity labels with the same semantics that are frequently executed. As confirmed by Mannhardt and Blind [23], activity labels *"Release C"*, *"Release D"*, and *"Release E"* are different variants of discharging a patient. Therefore, our ground truth consists of any pair of these 3 activity labels. For the

[11] We used the bin width of 1 minute for duration binning, the number 2 for null-valued distances, and uniform weights for measures within the temporal dimension.

[12] https://svn.win.tue.nl/repos/prom/Packages/SynonymousLabelRepair.

[13] https://data.4tu.nl/repository/uuid:76c46b83-c930-4798-a1c9-4be94dfeb741.

[14] To access to logs, ground truths and results, refer to https://s3-ap-southeast-2.amazonaws.com/event-log-quality/CoopIS2019/ReadMe.docx.

[15] https://data.4tu.nl/repository/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460.

third experiment, we used an event log from an Australian insurance company[16] with existing true imperfect activity labels.

Table 1. Characteristics of real-life logs used for evaluation

Log	#Trc	#Trc vars	#Evt	#Act	#Res	#Dt attrs	Life-cycle	#Aff acts	#Aff evts	%Aff evts
Hospital Billing	100000	1020	451359	18	1151	18	complete	0	0	0%
Sepsis	1050	846	15214	16	26	27	complete	3	55	0.36%
Insurance	17153	2197	49950	506	62	8	complete	418	7542	15.10%

Table 2 shows the characteristics of logs generated by renaming different percentages of activities of the Hospital Billing log and results of precision, recall, and F-score. Our strategy was to randomly pick a percentage of activity labels and for each label, randomly rename a percentage, up to 50%, of its events. This led to the generation of a new log, e.g. $H_{40,30}$ where 40% of activities are selected and for each label, 30% of its events are renamed. For each of the percentages reported in Table 2, five logs were generated and results are averaged over the five logs. We aimed to simulate different levels of imperfect labels that can be present in real-life scenarios. In computations of Table 2 and Figures 3 and 4 we assumed the final similarity threshold $\theta = 0.7$ and uniform weights for the timing pattern and duration measures within the temporal dimension. Then, in order to find the best weights of the four dimensions automatically[17], i.e. control flow, resource, temporal and data, we iterated Algorithm 1 where in each iteration, we set the weight of each dimension to an integer between 1 and 5 (inclusive), resulting in $5^4 = 625$ total weight settings, and we picked the first weight setting that maximizes the F-score and reported the results. As evidenced in Table 2, our approach achieves stable high precision, recall, and F-score, especially when renamed activities are frequent. However, when renamed activities are infrequent, e.g. in $H_{20,0.1}$ since only 0.017% of events of the log are affected, the approach gets a low F-score.

Table 2. Characteristics and results for the Hospital Billing log with artificial errors

Log	Traces Variants	Affected Activities	Affected Events	Affected Events%	Precision	Recall	F-score
$H_{20,0.1}$	1036	3	79	0.017%	0.4333	0.2833	0.3426
$H_{20,10}$	1448	4	11234	2.48%	0.8833	0.7833	0.8303
$H_{20,30}$	1741	4	38575	8.55%	1	0.8600	0.9247
$H_{40,30}$	2293	7	58308	12.91%	0.8560	0.9428	0.8973
$H_{40,50}$	2507	7	94378	20.91%	0.9250	0.9143	0.9196
$H_{60,50}$	3935	11	138396	30.66%	0.9333	0.8364	0.8822
$H_{80,50}$	4734	14	173395	38.41%	0.9703	0.8285	0.8938
$H_{100,50}$	6608	18	225684	50.00%	0.9889	0.8666	0.9238

In Figure 3, we compare our approach with four baselines, each considering only one of the context dimensions, control flow, resource, temporal and data,

[16] We can't release the log due to the NDA agreement with the organization.

[17] Of course where domain knowledge is available, it can guide the user to set the weights, but we want our approach to be applicable even if domain knowledge is not available by finding the best weights automatically.

since we could not find any equivalent approach in the literature that detects synonymous or polluted activity labels in event logs without relying on any external data, e.g. an ontology or a thesaurus. As shown in Figure 3, combining multiple context dimensions with the best weights, from 1 to 5 for each dimension, yields better F-scores than relying only on a single dimension. This is because when we rely solely on one dimension, e.g. data, then its quality highly influences the results, however, by investigating multiple dimensions, and assigning low priorities (weights) to low-quality ones, we can improve the outcomes. As evidenced in Figure 3, for infrequent activity labels, i.e. below 3%, the resource and data dimensions are more informative than the control flow dimension, while for more frequent activity labels, the control flow dimension is more helpful. We can also see in Figure 3 that, the temporal dimension is the least informative dimension for all the percentages of affected events, so assigning a low weight to this dimension would result in a better F-score. For instance, for the $H_{40,30}$ log with 12.91% affected events, the control flow and resource dimensions are more informative than the data and temporal dimensions. This fact is also evidenced in Figure 4 where we show how F-score of our approach, applied on the $H_{40,30}$ log, varies with different dimension weights ranging from (1, 1, 1, 1) to (5, 5, 5, 5) for the control flow, resource, temporal, and data dimensions respectively. We can see that assigning low weights to the control flow and resource dimensions and high weights to the temporal and data dimensions, i.e. (1, 1, 5, 5) yields the worst F-score of 0.57, while assigning high weights to the control flow and resource dimensions and low weights to the temporal and data dimensions, i.e. (5, 4, 1, 1) yields the best F-score of 0.92.

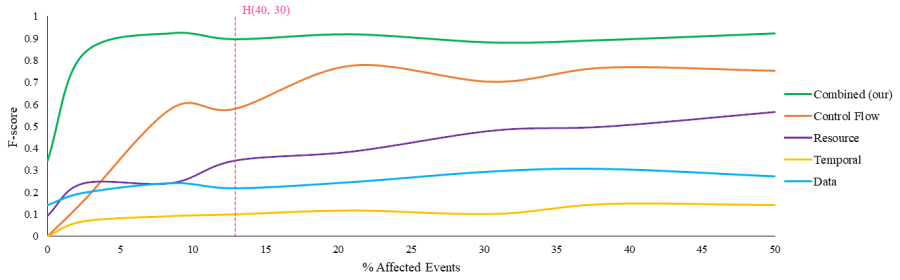

Fig. 3. F-score of our approach (combined) compared to single-dimension baselines for the Hospital Billing log with artificial errors

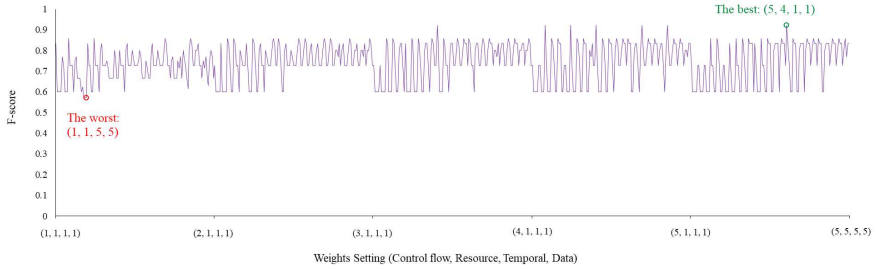

Fig. 4. F-score of our approach for different combinations of weights for the $H_{40,30}$ log

Table 3 reports the results obtained from the Sepsis, Insurance, and $H_{40,30}$ logs for different final similarity thresholds θ, i.e. 0.7, 0.8, and 0.9, with the best weights[18], compared to the four single-dimension baselines. These results, same as the last experiments shown in Figure 3, suggest that combining multiple dimensions with the best weights yields higher F-scores than single dimensions. We can see that increasing the threshold leads to higher precision and lower recall as the number of detections decreases. However, a too high threshold, i.e. $\theta = 0.9$ may result in detecting nothing, as e.g. for the $H_{40,30}$ log. Therefore 0.7 seems to be more suitable than other choices. For this threshold, applying our approach on the $H_{40,30}$ and Sepsis logs results in high F-scores of 0.89 and 1 respectively, however, for the Insurance log, we see a lower F-score of 0.36. This is because there are many activity labels with low frequency in the Insurance log, e.g. 324 of a total of 506 distinct activity labels have a frequency of 1, and our approach is better in detecting frequent imperfect activity labels as we are looking at probability distributions for different context dimensions. Furthermore, we can see that for the Insurance log, which contains infrequent activity labels, the data and especially the resource dimensions seem to be more informative than the control flow and temporal dimensions, while for the other two logs, which have more frequent activities, the control flow and resource dimensions seem to be more helpful than the data and temporal dimensions, as e.g. in the Sepsis log considering the data and temporal dimensions results in F-score of 0 for all thresholds. This is the same conclusion as for our last experiments with artificial imperfect activity labels depicted in Figure 3.

Table 3. Results for different thresholds compared to four single-dimension baselines

		$\theta = 0.7$			$\theta = 0.8$			$\theta = 0.9$		
Log	**Approach**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
$H_{40,30}$	Control flow only	0.8800	0.4333	0.5667	0.8000	0.2381	0.3543	0	0	0
	Resource only	0.2098	0.9714	0.3446	0.3592	0.9429	0.5180	0.4215	0.8000	0.5516
	Temporal only	0.0530	0.9429	0.1002	0.0566	0.9429	0.1068	0.0628	0.8762	0.1172
	Data only	0.1444	0.4619	0.2193	0.1356	0.2952	0.1847	0.1364	0.2333	0.1702
	Combined (Our)	0.8559	0.9428	0.8944	0.8714	0.9143	0.8788	1	0.5904	0.7388
Sepsis	Control flow only	0.2000	0.3333	0.2500	0.5000	0.3333	0.4000	0.5000	0.3333	0.4000
	Resource only	0.1579	1	0.2727	0.1579	1	0.2727	0.1579	1	0.2727
	Temporal only	0	0	0	0	0	0	0	0	0
	Data only	0	0	0	0	0	0	0	0	0
	Combined (Our)	1	1	1	1	1	1	1	1	1
Insurance	Control flow only	0.5468	0.0103	0.0202	0.5468	0.0103	0.0202	0.5468	0.0103	0.0202
	Resource only	0.2095	0.4614	0.2882	0.2098	0.4614	0.2885	0.2211	0.4612	0.2989
	Temporal only	0.0263	0	0.0001	0.0263	0	0.0001	0.0263	0	0.0001
	Data only	0.5351	0.0839	0.1451	0.6384	0.0112	0.0219	0	0	0
	Combined (Our)	0.3202	0.4098	0.3595	0.4468	0.2543	0.3241	0.6340	0.0053	0.0106

Table 4 reports the results of our approach (context similarity with the best weights) on the Sepsis and Insurance logs[19] compared to the label similarity method and shows what happens if we combine our approach with the label

[18] We assigned weights 1,2,3,4,5 to each of the four dimensions and picked the first one that maximizes the F-score.

[19] We did not include the Hospital Billing logs in this experiment because their activity labels were artificially renamed to arbitrary names and therefore applying label similarity on those names would not result in meaningful outcomes.

similarity approach. The label similarity method assumes two activity labels to be synonyms or polluted if their normalized Levenshtein string distance [19] is lower than a threshold θ_l[20]. The results suggest that the label similarity method gets lower F-scores than our approach, which relies on context similarity, for the Insurance log and much lower F-score for the Sepsis log. The reason is that the Sepsis log contains activity labels that are syntactically similar but semantically different, e.g. *"Admission IC"* and *"Admission NC"* regarding admission of a patient to intensive care and normal care in a hospital respectively. This also explains why the label similarity method gets a low precision for the Sepsis log. We can also see that the label similarity method has a neutral (no positive and no negative) effect on the results of our approach when it is added as another component[21] as we are already achieving the highest F-score of 1 by considering activity context only. However, for the Insurance log, combining our approach with label similarity yields an improvement of 0.04 of the F-score.

Table 4. Comparison of context similarity (our) and label similarity approaches

Log	Approach	Precision	Recall	F-score
Sepsis	Context similarity (our)	1	1	1
	Label similarity	0.2727	1	0.4216
	Context and label similarity	1	1	1
Insurance	Context similarity (our)	0.3202	0.4098	0.3595
	Label similarity	0.9525	0.1563	0.2685
	Context and label similarity	0.4818	0.3438	0.4012

Table 5 reports the time performance, excluding the selection of the best weights, of our experiments with different logs. The results show that our approach works in a quite reasonable time since it looks at multiple context dimensions of each event. The time is increasing polynomially with the number of activities and linearly with the number of events. For the Insurance log, with 506 activities, the approach takes 7 to 8 minutes on average and the required time for deciding on each activity is 897 milliseconds.

Table 5. The time performance for experiments with different logs

Log	Time (sec)			
	Avg	StDev	Min	Max
$H_{40,30}$	28.345	2.739	24.345	31.223
Sepsis	2.060	3.126	1.634	2.432
Insurance	454.017	25.228	452.768	480.311

7 Conclusion

We have proposed a contextual approach for detecting synonymous and polluted activity labels since they manifest themselves in real-life event logs [29]. We have discussed different activity context dimensions, i.e. control flow, resource, time,

[20] We have considered final similarity threshold $\theta = 0.7$ and distance threshold $\theta_l = 0.3$ for the computations of Table 4 to compare methods under the same conditions.

[21] In combining our approach with label similarity we still select the best weights, i.e. from 1 to 5 for each dimension as well as the label similarity measure.

and data, each with (a) dedicated distance measure(s). Synonymous and polluted labels are detected through the same approach which is looking at the overall context similarity. However, synonymous and polluted labels may need different treatments when it comes to repair, which is left as future work. We have evaluated our approach using real-life logs from two hospitals and an insurance company. The results show that we can detect frequent synonymous and polluted labels, which are more serious problems than infrequent ones, efficiently. The results also suggest that the control flow and resource dimensions are more informative for detecting frequent imperfect activity labels, while for detecting infrequent ones, the resource and data dimensions are more helpful. Furthermore, the temporal dimension seems to be the least informative perspective for detecting frequent and infrequent imperfect activity labels. Some possible future avenues of research are developing techniques for detecting infrequent imperfect activity labels more efficiently, repairing the detected labels, and considering other dimensions and measures, e.g. case data attributes.

References

1. van der Aa, H., Gal, A., Leopold, H., et al.: Instance-Based Process Matching Using Event-Log Information. In: Dubois, E., Pohl, K. (eds.) CAiSE. LNCS, vol. 10253, pp. 283–297. Springer (2017)
2. Van der Aalst, W.M.P.: Process Mining: Data Science in Action. Springer, 2nd edn. (2016)
3. van der Aalst, W.M.P., Adriansyah, A., de Medeiros, A.K.A., et al.: Process Mining Manifesto. In: BPM Workshops. LNBIP, vol. 99, pp. 169–194. Springer (2011)
4. van der Aalst, W.M.P., Dustdar, S.: Process Mining Put into Context. IEEE Internet Computing 16(1), 82–86 (2012)
5. Becker, M., Laue, R.: A comparative survey of business process similarity measures. Computers in Industry 63(2), 148–167 (2012)
6. Bose, R.J.C., Mans, R.S., van der Aalst, W.M.P.: Wanna Improve Process Mining Results - It's High Time We Consider Data Quality Issues Seriously. Tech. Rep. BPM-13-02, BPM Center (2013)
7. Bose, R.J.C., Mans, R.S., van der Aalst, W.M.P.: Wanna Improve Process Mining Results - It's High Time We Consider Data Quality Issues Seriously. In: Computational Intelligence and Data Mining Symposium. pp. 127–134. IEEE (2013)
8. Cairns, A.H., Ondo, J.A., Gueni, B., Fhima, M., Schwarcfeld, M., Joubert, C., Khelifa, N.: Using Semantic Lifting for Improving Educational Process Models Discovery and Analysis. In: Symposium on Data-driven Process Discovery and Analysis. CEUR, vol. 1293, pp. 150–161 (2014)
9. Celino, I., de Medeiros, A.K.A., Zeissler, G., et al.: Semantic Business Process Analysis. In: Workshop on Semantic Business Process and Product Lifecycle Management. CEUR, vol. 251, pp. 44–47. CEUR-WS (2007)
10. Conforti, R., La Rosa, M., ter Hofstede, A.H.M.: Timestamp Repair for Business Process Event Logs. Tech. rep., The University of Melbourne (2018)
11. Craw, S.: Manhattan Distance. In: Encyclopedia of Machine Learning and Data Mining. pp. 790–791. Springer (2017)
12. Dijkman, R., Dumas, M., van Dongen, B., et al.: Similarity of business process models: Metrics and evaluation. Information Systems 36(2), 498–516 (2011)

13. Dixit, P.M., Suriadi, S., Andrews, R., et al.: Detection and Interactive Repair of Event Ordering Imperfection in Process Logs. In: CAiSE. LNCS, vol. 10816, pp. 274–290. Springer (2018)
14. Günther, C.W.: Process Mining in Flexible Environments. Ph.D. thesis, Einhoven University Of Technology (2009)
15. Klinkmüller, C., Weber, I., Mendling, J., et al.: Increasing recall of process model matching by improved activity label matching. In: BPM Conferenece. LNCS, vol. 8094, pp. 211–218. Springer (2013)
16. Koschmider, A., Ullrich, M., Heine, A., et al.: Revising the Vocabulary of Business Process Element Labels. In: CAiSE. LNCS, vol. 9097, pp. 69–83. Springer (2015)
17. Kruskal, J.B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. The American Mathematical Society 7(1), 48–50 (1956)
18. Leoni, M.D., van der Aalst, W.M.P.: Data-Aware Process Mining: Discovering Decisions in Processes Using Alignments. In: SAC. pp. 1454–1461. ACM (2013)
19. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady 10(8), 707–710 (1966)
20. Lu, X., Fahland, D.: A Conceptual Framework for Understanding Event Data Quality for Behavior Analysis. In: ZEUS. CEUR, vol. 1826, pp. 11–14 (2017)
21. Lu, X., Fahland, D., Andrews, R., et al.: Semi-supervised Log Pattern Detection and Exploration Using Event Concurrence and Contextual Information. In: OTM Conferences. LNCS, vol. 10573, pp. 154–174. Springer (2017)
22. Lu, X., Fahland, D., van den Biggelaar, F.J., van der Aalst, W.M.P.: Handling Duplicated Tasks in Process Discovery by Refining Event Labels. In: BPM Conference. LNCS, vol. 9850, pp. 90–107. Springer (2016)
23. Mannhardt, F., Blinde, D.: Analyzing the Trajectories of Patients with Sepsis using Process Mining. In: CAiSE. CEUR, vol. 1859, pp. 72–80 (2017)
24. Mans, R., van der Aalst, W.M.P., Vanwersch, R., Moleman, A.: Process Mining in Healthcare: Data Challenges When Answering Frequently Posed Questions. In: ProHealth. LNCS, vol. 7738, pp. 140–153. Springer (2012)
25. Massey Jr, F.J.: The Kolmogorov-Smirnov Test for Goodness of Fit. Journal of the American statistical Association 46(253), 68–78 (1951)
26. Medeiros, A.K.A.D., Pedrinaci, C., van der Aalst, W.M.P., et al.: An Outlook on Semantic Business Process Mining and Monitoring. In: OTM Workshops. LNCS, vol. 4806, pp. 1244–1255. Springer (2007)
27. Rosemann, M., Recker, J., Flender, C.: Contextualisation of Business Processes. Int. J. of Business Process Integration and Management 3(1), 47–60 (2008)
28. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. of Computational and Applied Mathematics 20, 53–65 (1987)
29. Suriadi, S., Andrews, R., ter Hofstede, A.H.M., Wynn, M.T.: Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs. Information Systems 64, 132–150 (2017)
30. Tan, P.N., Steinbach, M., Kumar, V.: Cluster Analysis: Additional Issues and Algorithms. In: Introduction to Data Mining. pp. 569–650. Pearson (2005)
31. Tax, N., Alasgarov, E., Sidorova, N., et al.: Generating Time-Based Label Refinements to Discover More Precise Process Models. Tech. rep., Eindhoven University of Technology (2017)
32. Verhulst, R.: Evaluating Quality of Event Data within Event Logs: An Extensible Framework. Master's thesis, Eindhoven University of Technology (2016)