Discovering Predictive Dependencies on Multi-Temporal Relations

Beatrice Amico
Department of Computer Science, University of Verona, Italy

Carlo Combi
Department of Computer Science, University of Verona, Italy

Romeo Rizzi
Department of Computer Science, University of Verona, Italy

Pietro Sala
Department of Computer Science, University of Verona, Italy

Abstract

In this paper, we propose a methodology for deriving a new kind of approximate temporal functional dependencies, called Approximate Predictive Functional Dependencies (APFDs), based on a three-window framework and on a multi-temporal relational model. Different features are proposed for the Observation Window (OW), where we observe predictive data, for the Waiting Window (WW), and for the Prediction Window (PW), where the predicted event occurs. We then discuss the concept of approximation for such APFDs, introduce two new error measures. We prove that the problem of deriving APFDs is intractable. Moreover, we discuss some preliminary results in deriving APFDs from real clinical data using MIMIC III dataset, related to patients from Intensive Care Units.

2012 ACM Subject Classification Information systems → Relational database model; Information systems → Data mining

Keywords and phrases temporal databases, temporal data mining, functional dependencies

Digital Object Identifier 10.4230/LIPIcs.TIME.2023.4

1 Introduction

Knowledge from databases may be expressed by discovering patterns and data dependencies. Database dependencies express relevant characteristics of datasets, thereby enabling various critical analyses of data. Functional dependencies (FDs) have been proposed as a way of mining data, i.e., by discovering those FDs that hold on most data. The considered approximation may be heterogeneous and deal with both null values, quantitative data, data deletion/updates, and so on [7, 4, 18, 7, 12, 19].

Temporal Functional Dependencies (TFDs) received some interest since the nineties, initially as a way for specifying constraints on temporal data [32, 9, 5], and, more recently, as a mining approach in their approximate version, looking for hidden temporal patterns inside data [8, 25, 10].

To the best of our knowledge, TFDs have not yet been considered for the prediction task. Such decision-support task is mainly devoted to the prediction of some (future) event based on a (past) data history. Thus, as time is an inherent feature of this task, TFDs are interesting candidates as a formal tool, for discovering the predictivity of the stored data. Within this context, in this paper we propose and discuss an original temporally-oriented data mining framework to support the prediction of future events through the identification of recurring past temporal data patterns, expressed as *Approximate Temporal Predictive Functional Dependencies* (APFDs), according to a 3-window -based temporal framework. New kinds of error and related thresholds are introduced, to deal with the required approximation. The main novelty can be summarized in the formalization of a new framework to exploit the predictive aspect of the APFDs, according to the following specific aspects:

- We introduce a new temporal framework based on three temporal windows: observation window (OW), waiting window (WW), and prediction window (PW). The waiting window is explicitly introduced to create a time span before the prediction for being able to (possibly) manage the predicted event.
- We define and exemplify the entire framework for the approximate predictive functional dependencies (APFDs) in a formal way by introducing and characterizing *multi-temporal relations*. It allows the representation of temporal patterns (made by attribute values) related to a set of observed entities (e.g., patients) and characterizes their predictivity, with respect to a target attribute (e.g., a disease).
- We discuss different kinds of error measures, named G_3, H_3, and J_3, to be evaluated when deriving APFDs;
- We discuss the (data) complexity of the problem of checking for APFDs and prove that is exponential. We then propose a new algorithm for checking APFDs.
- We provide some experimental results on real clinical data from patients in Intensive Care Units, using data from MIMIC III [16], to obtain different APFDs.

With respect to the preliminary proposal of APFDs sketched in [3], as specific novelties, here we first characterize a new temporal data model, based on relations having multiple valid times; we introduce the three-window framework and the related APFDs for such model; we extensively consider the related data complexity; we propose a new algorithm for checking APFDs; we discuss further experimental evaluation.

Our paper unfolds as follows. Section 2 contains the related work; in Section 3 we introduce and motivate the 3-window-based framework for prediction, the formalization of APFDs and their approximation; in Section 4 we discuss the data complexity of deriving APFDs, and provide a deterministic algorithm that could stop the analysis of a relation, as soon as it verifies that the relation cannot satisfy the given APFD; in Section 5, we introduce and discuss some experimental results and finally in Section 6 we draw some conclusions. The Appendix A completes the description of our approach through the proof of the NP-hardness of the APFDs-checking problem.

2 Related work

FDs were originally proposed to specify data constraints in the relational setting and then to derive normalized relational schemata [2].

Let us briefly recall the concept of FD in the context of relational databases [2]. Let r be a relation over the relational schema $R(U)$ and let $X, Y \subseteq U$. r fulfills the functional dependency $X \to Y$ (written as $r \models X \to Y$) if $\forall t, t' \in r(t[X] = t'[X], \to t[Y] = t'[Y])$.

In more recent years FDs have been extended in many different directions and with different goals. Here we mainly consider three different research directions: the first one deals with the representation of constraints on temporal data through *temporal functional dependencies* (TFDs), the second one focuses on the discovery of *approximate functional dependencies* (AFDs), and the third one deals with the use of *FDs to support prediction and classification tasks.*

TFDs add a temporal dimension to classical FDs to deal with temporal data. In literature, several kinds of TFDs have been proposed and various representation formalisms have been developed [5, 15, 29, 30, 31, 9]. In [9] Combi et al. propose a new formalism for the representation of TFDs, involving multiple time granularities. They identify four relevant classes, named *pure temporally grouping*, *pure temporally evolving*, *temporally mixed*, and *temporally hybrid* TFDs, respectively.

In [22], the authors face another temporal aspect, which stems from the observation that frequent constraint violations in a database may be related to the fact that the considered (mini) world is changing, while the specified constraints remain static. FDs violated by current data are then identified and some approaches are proposed to suitably modify the given FD according to the new reality represented through the current data. In [26], the authors deal with the problem of continuously discovering FDs on dynamic datasets in an efficient way, and propose an incremental approach to solve it.

AFDs derive from the concept of plain FD. Given a relation r where an FD holds for most of the tuples in r, we may identify some tuples for which that FD does not hold. In [18], Kivinen and Mannila introduce three measures, known as G_1, G_2 and G_3 considering, respectively, the number of violating couples of tuples, the number of tuples that violate the functional dependency, and finally the minimum number of tuples in r to be deleted for the FD to hold. Discovering AFDs is a computationally expensive task, and different algorithms have been proposed to perform the discovery in an efficient way [19]. More recently, AFDs have been included in the wider scenario of *relaxed FDs* (RFDs), where not only exceptions, i.e., violating tuples, are considered, but also similarities among attribute values and conditional constraints [7, 6].

Temporal data mining techniques merging AFDs with TFDs have been proposed in [8], where the authors propose approximate temporal functional dependencies (ATFDs), which are defined and measured either on temporal granules or on sliding windows, and apply them to mine data from psychiatry and pharmacovigilance domains. They introduce a new error measure G_4, which considers the minimum number of tuples in r which must be modified for the plain TFD to hold on all the tuples of r. In [1], the authors present AETAS, a system for the discovery of approximate temporal functional dependencies. The discovered TFDs are mainly pure temporally grouping TFDs with moving windows, according to the classification proposed in [9]. Also conditional TFDs are considered, where the moving window may have different values according to specific values of atemporal attributes. As an interesting aspect of AETAS, the authors deal with the discovery of TFDs from dirty web data, as well as with the discovery of the "optimal" duration for the moving window.

Moving to contributions dealing with the use of *FDs to support prediction and classification tasks*, in [20] the authors show that if there is a functional dependency between features, it is likely to affect the classifier negatively. In [21], the authors address the notion of trusting ML models by using also functional dependencies, discussing on the relationships between supervised classification and functional dependencies. They consider the issue of estimating the feasibility of classification over a given dataset using functional dependencies. As far as we know, few studies till now considered functional dependencies in this context, where, given a set of features $(A_1, ..., A_n, C)$ where C values represent the class to be classified, the problem is to understand whether functional dependencies such as $A_1, ..., A_k \rightarrow A_j$ influence the classification performances.

3 The predictive aspects of functional dependencies

In this section, firstly we delineate the problem at hand, and introduce a 3-window model for the interpretation of predictive temporal data; then we illustrate the definitions needed to obtain a Predictive Functional Dependency, and finally, we analyze the concept of approximation for the Predictive Functional Dependencies.

Table 1 The multi-temporal relation `PatientHistory`, with a single atemporal attribute and one attribute for each valid time.

#	**Patient**	$\overline{\mathbf{HR}}^1$	$\overline{\mathbf{VT}}^1$	$\overline{\mathbf{SpO_2}}^2$	$\overline{\mathbf{VT}}^2$	$\overline{\mathbf{Drug}}^3$	$\overline{\mathbf{VT}}^3$	$\dot{\mathbf{AKI}}$	$\dot{\mathbf{VT}}$
1	Daisy	High	19	High	21	Aspirin	23	False	28
2	Daisy	Low	2	High	4	Aspirin	6	False	18
3	Daisy	Low	2	Medium	4	Aspirin	6	False	12
4	Daisy	Medium	5	Medium	7	Indapamide	9	False	18
5	Luke	Low	7	High	8	Ibuprofen	12	True	17
6	Luke	Low	7	High	8	Ibuprofen	12	True	21
7	Luke	Medium	9	High	13	Sulindac	14	True	18
8	Luke	Medium	9	High	13	Sulindac	14	True	21
9	Stevie	Medium	4	Medium	7	Metolazone	8	True	13
10	Stevie	High	1	Low	2	Aspirin	5	False	8
11	Stevie	High	1	Low	2	Indapamide	7	False	8
..	...	...	...	...	...	...	...	...	...
36	Stevie	High	1	Low	2	Aspirin	5	False	25
..	...	...	...	...	...	...	...	...	...

3.1 A motivating scenario from Clinical Medicine

To illustrate the relevance and the potential meaning of our approach, we consider a real-world example from the domain of Intensive Care Unit (ICU) focusing on patients suffering from Acute Kidney Injury (AKI) [28], used as reference throughout the paper. In ICU, Acute Kidney Injury is a frequent clinical problem, characterized by sudden loss of the ability of the kidneys to excrete wastes, concentrate urine, store electrolytes, and maintain fluid balance [27].

In 2012, KDIGO (Kidney Disease: Improving Global Outcomes) published specific guidelines [17] for the definition of AKI, where a patient receives the diagnosis if one of the following criteria is satisfied: (i) an increase in serum creatinine by ≥ 0.3 mg/dl (≥ 26.5 $\mu mol/l$) within 48 h, (ii) an increase in serum creatinine to ≥ 1.5 times baseline within the previous 7 days and (iii) a urine volume ≤ 0.5 ml/kg/h for 6 hours.

As we are interested in discovering whether some clinical data features allow the early identification of AKI patients, let us assume that we derive through a suitable query the (possibly materialized) view `PatientHistory`. It represents different ordered states of patients, we would like to associate to a final state, specifying whether the patient has AKI. Each state is represented by some attribute values and is associated to a *valid time* (VT), representing the timepoint when the state information is true in the modeled world [14]. Table 1 (partially) shows a possible instance of `PatientHistory` describing a clinical history of three patients, Daisy, Luke, and Stevie, who have some measured vital signs and undergo five different drugs, some of them specific for the AKI treatment. Such history can be derived from the data contained in a clinical database [16].

3.2 A 3-window framework for the interpretation of predictive temporal data

In general, the prediction models exploit the use of two-time windows, namely (i) a data collection (or observation) window, and (ii) a prediction window. Even though there are approaches [11, 24] which consider a third temporal window, to the best of our knowledge, a general and formal prediction framework considering three different time windows has not yet been considered in the data mining literature.

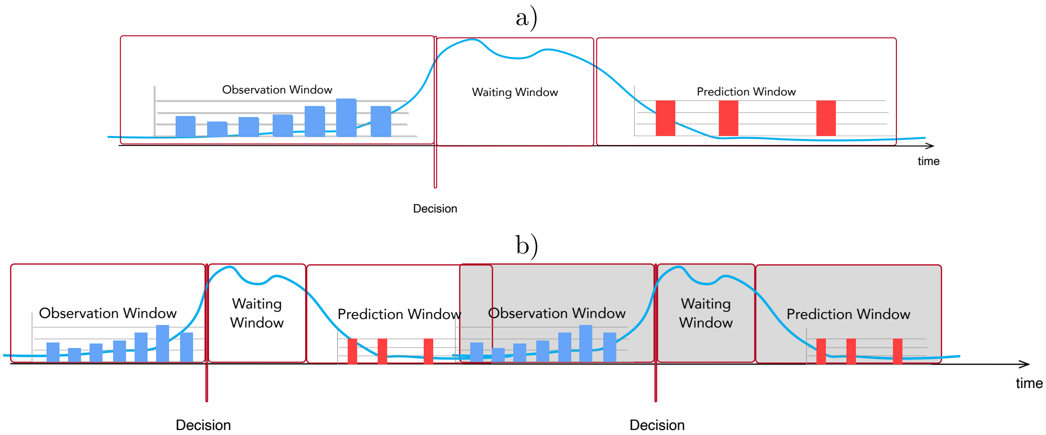

Figure 1 The time windows of the proposed framework: (a) the anchored and (b) the unanchored –sliding window– case.

According to this view, depicted in Figure 1, we can observe:

1. Decisions are taken after gathering information for some time span (*Observation Window*: OW).
2. After the moment when the decision is taken, we have to execute all the related actions and (possibly) wait for a while (*Waiting Window*: WW). The WW is held to be the minimum time interval required to act in order to prevent the event in the prediction window. Indeed, not all the performed actions have an instantaneous effect.
3. The last temporal window refers to when the possible effects of the decision are observable and thus we can evaluate the suitability of the taken decision (*Observation Window*: OW).

It is worth noting that the span of such windows may be different and could be also composed of a single time-point. Moreover, the *Waiting Window* could be missing, i.e., of zero length, in case of decisions having an immediate observable effect.

In general, we may identify different orthogonal features for the introduced time windows.

The first distinction is between (i) *anchored* and (ii) *unanchored* time windows. Indeed, with anchored time windows, we are able to represent specific periods of the considered time axis. An example of anchored time windows for the motivating scenario could consist of specifying OW as the first 4 hours from the admission to the ICU, the following 2 hours as WW (i.e., the fifth and sixth hour after the ICU admission), ending with the PW from the seventh to the tenth hour after the ICU admission. Figure 1 a) depicts the three anchored windows, the time-point corresponding to the decision moment, and possible temporal evolution of some observed quantitative parameter, having some varying behavior. On the other side, unanchored time windows represent windows that "move" through the time axis, constraining only the distance between the considered data. An example of such kind of windows for our scenario could consist in specifying again 4 hours, 2 hours, and 4 hours for OW, WW, and PW, respectively, but not anchored to any point of the time axis. Figure 1 b) represents two partially overlapping views, representing unanchored time windows. In this case, we may consider a possibly infinite number of unanchored (sliding) windows, even by specifying the width of the step size of sliding.

▶ **Definition 1** (Unanchored Time-Frame). *An unanchored time-frame (uTF) α is a triple $\langle OW, WW, PW \rangle$ where OW, WW, and PW are expressed as durations, i.e., time distances. They allow the representation of three different unanchored windows, which we will use to observe temporal data.*

▶ **Definition 2** (Anchored Time-Frame). *An anchored time-frame (aTF) α is a time-frame associated to an anchor time point and can be represented through the structure $\langle atp, \langle OW, WW, PW\rangle\rangle$, where atp is a (anchor) time point.*

A second subtle distinction, which may provide different results for prediction and is orthogonal with respect to the distinction between anchored and unanchored time windows, is between (i) *fixed-length* and (ii) *variable-length* time windows. Indeed, OW, and consequently the following WW and PW, could be either of fixed length, without any further constraint related to the temporal position of data inside it, or of variable length, and thus ending with the last time point associated with the data to consider in the window.

3.3 A multi-temporal relational model and its connection to the temporal framework

Let us introduce the concept of *multi-temporal relation.* Informally, a multi-temporal relation is characterized by multiple valid times. Each tuple of such relation represents a piece of history of a given entity, through the values of attributes holding at different (valid) times. A set of attributes of such relation allows the (optional) identification of the considered entities (e.g., a patient, an employee) and their characterization. Any other attribute of such relation is associated with a specific valid time.

▶ **Definition 3** (Multi-temporal relation (*mt*-relation)). *Given an overall set of attributes $\mathcal{A}$ and a set of valid time attributes $\mathcal{VT}$, a multi-temporal relation mtr is a relation with schema WT where $W \subseteq \mathcal{A}$ and $T = \{VT_1, ..., VT_i, ...VT_k, VT_{k+1}\} \subseteq \mathcal{VT}$ are $k+1$ valid time attributes.*

For a multi-temporal relation schema, a mapping $Vtime : T \to 2^W$ allows us to specify the attribute subset associated to a specific valid time. For such mapping, it holds

$Vtime(VT_i) \cap Vtime(VT_j) = \emptyset$ *for any i, j with $i \neq j$*

The (possibly empty) set $Z \subseteq W$, $Z = W - \bigcup_{i=1}^{k+1} Vtime(VT_i)$ contains attributes not associated with any valid time attribute.

For any relation mtr it holds

$$\forall t \in mtr(t[VT_i] < t[VT_j]) \text{ for } 1 \leq i < j \leq k+1$$

As we will discuss in the following, the main idea here is to propose a general framework allowing the definition of "specialized" functional dependencies having the antecedent composed of a set of attributes, called *predictive attributes*, ordered according to the corresponding valid times and the consequent defined as the predicted attributes. In order to distinguish such roles for attributes, we introduce a suitable partition of attributes, according to the following definition.

▶ **Definition 4** (Prediction-oriented partition of mt-relation valid times). *Given a multi-temporal relation mtr with schema WT, where $W \subseteq \mathcal{A}$ and $T = \{VT_1, ..., VT_i, ...VT_k, VT_{k+1}\}$, attributes in T are partitioned in two sets $\mathcal{O}$, for observation-related valid times, and $\mathcal{P}$, for prediction-related valid times, where it holds*

$$\forall\, VT_o, VT_p((VT_o \in \mathcal{O} \wedge VT_p \in \mathcal{P}) \implies \forall t \in mtr(t[VT_o] < t[VT_p]))$$

For the sake of simplicity and without losing generality, in the following, we assume that $\mathcal{O} \equiv \{VT_1, VT_2, ..., VT_k\}$, while $\mathcal{P} \equiv \{VT_{k+1}\}$. According to this choice, we use an overline-based notation for (ordered) observation-related valid times and the associated attributes. We use a dot notation for the prediction-oriented valid time and the associated attributes.

▶ **Example 5.** The relation view depicted in Table 1 considers attributes according to the introduced notation. More precisely, in this case $\mathcal{O} \equiv \{\overline{VT}^1, \overline{VT}^2, \overline{VT}^3\}$, $\mathcal{P} \equiv \{\dot{VT}\}$, and $Vtime(\overline{VT}^1) = \{\overline{HR}^1\}$, $Vtime(\overline{VT}^2) = \{\overline{SpO_2}^2\}$, $Vtime(\overline{VT}^3) = \{\overline{Drug}^3\}$, and $Vtime(\dot{VT}) = \{A\dot{K}I\}$.

Given a multi-temporal relation mtr, we are now interested in verifying which tuples are "fine" with or "contained" in a given time frame. More precisely, we are interested in eliciting those tuples having the (some of the) k observation-related valid times contained in the observation window OW, and the last valid time in the prediction window PW. We will call them *consistent* with the considered time frame.

In the following, we will introduce different kinds of *time-frame consistency*, mainly considering both the partial containment of some valid times in the observation window and different requirements for the observation window.

Indeed, as for the first aspect, we may be interested in verifying the partial/complete containment of the k observation-related valid times within the given OW, while for the second one, we may consider either fixed length OWs, or flexible observation windows, which end with the last valid time we have to consider in the given OW.

▶ **Definition 6** (Time-frame tuple consistency with range and modality)**.** *Given a tuple t of a multi-temporal relation mtr with schema WT, where $W \subseteq \mathcal{A}$ and $T = \{VT_1, ..., VT_i, ...VT_k, VT_{k+1}\} \subseteq \mathcal{VT}$, and a (either anchored or unanchored) time frame α, we say that t is time-frame consistent with α according to modality $m \in \{\text{'flex'}, \text{'fixed'}\}$ in the range $[i_1, i_2]$, where $1 \leq i_1 < i_2 \leq k$, if formula $\Theta(t, \alpha, m, [i_1, i_2])$ holds.*

Formula $\Theta(t, \alpha, m, [i_1, i_2])$ is defined according to the following cases:

- $\Theta(t, \alpha, \text{'fixed'}, [i_1, i_2]) \equiv t[\overline{VT}^{i_2}] - t[\overline{VT}^{i_1}] \leq OW \wedge t[\dot{VT}] - t[\overline{VT}^{i_1}] > OW + WW \wedge t[\dot{VT}] - t[\overline{VT}^{i_1}] < OW + WW + PW$
  –if the time-frame is unanchored–, or
- $\Theta(t, \alpha, \text{'fixed'}, [i_1, i_2]) \equiv t[\overline{VT}^{i_1}] \geq atp \wedge t[\overline{VT}^{i_2}] \leq atp + OW \wedge t[\dot{VT}] > atp + OW + WW \wedge t[\dot{VT}] < atp + OW + WW + PW$
  –if the time-frame is anchored–, or
- $\Theta(t, \alpha, \text{'flex'}, [i_1, i_2]) \equiv t[\overline{VT}^{i_2}] - t[\overline{VT}^{i_1}] \leq OW \wedge t[\dot{VT}] - t[\overline{VT}^{i_2}] > WW \wedge t[\dot{VT}] - t[\overline{VT}^{i_2}] < WW + PW$
  –if the time-frame is unanchored–, or
- $\Theta(t, \alpha, \text{'flex'}, [i_1, i_2]) \equiv t[\overline{VT}^{i_1}] \geq atp \wedge t[\overline{VT}^{i_2}] \leq atp + OW \wedge t[\dot{VT}] - t[\overline{VT}^{i_2}] > WW \wedge t[\dot{VT}] - t[\overline{VT}^{i_2}] < WW + PW$
  –if the time-frame is anchored–

3.4 Defining Predictive FDs

The overall idea is now to temporally characterize functional dependencies $X \to Y$ for the introduced multi-temporal relational model, by considering for the attribute set X those attributes related to "past" properties, while attributes Y would be those attributes related to "future" properties. "Past" and "future" values are evaluated according to a given time-frame consistency.

▶ **Definition 7** (Predictive Functional Dependency (PFD))**.** *Given an mt-relation schema $MTR(Z\overline{U}^1\overline{U}^2..\overline{U}^k\dot{U} \cup \{\overline{VT}^1, \overline{VT}^2, .., \overline{VT}^k, \dot{VT}\})$, a time frame, and a modality $m \in \{\text{"flex"}, \text{"fixed"}\}$, a Predictive Functional Dependency is expressed as:*

$$S\overline{P}^h\overline{Q}^i ...\overline{R}^j \xrightarrow[\alpha,m]{} \dot{Y} \quad \text{with } 1 \leq h < i < ... < j \leq k$$

where $S \subseteq Z, \overline{P}^h \subseteq \overline{U}^h, \overline{Q}^i \subseteq \overline{U}^i, \overline{R}^j \subseteq \overline{U}^j$ and $\dot{Y} \subseteq \dot{U}$ is the predicted attribute set.

A PFD holds on an mt-relation mtr with schema MTR in a timeframe TF with modality m, with an extended range semantics (denoted as $mtr \models^{E}_{\alpha,m} S\overline{P}^h\overline{Q}^i ... \overline{R}^j \to \dot{Y}$*) iff*

$$\forall t, t' \in mtr((t[S\overline{P}^h\overline{Q}^i...\overline{R}^j] = t'[S\overline{P}^h\overline{Q}^i..\overline{R}^j] \wedge \Theta(t, \alpha, m, [1, k]) \wedge \Theta(t', \alpha, m, [1, k])) \\ \to t[\dot{Y}] = t'[\dot{Y}])$$

A PFD holds on an mt-relation mtr with schema MTR in a timeframe TF with modality m, with a restricted range semantics (denoted as $mtr \models^{R}_{\alpha,m} S\overline{P}^h\overline{Q}^i ... \overline{R}^j \to \dot{Y}$*) iff*

$$\forall t, t' \in mtr((t[S\overline{P}^h\overline{Q}^i...\overline{R}^j] = t'[S\overline{P}^h\overline{Q}^i..\overline{R}^j] \wedge \Theta(t, \alpha, m, [h, j]) \wedge \Theta(t', \alpha, m, [h, j])) \\ \to t[\dot{Y}] = t'[\dot{Y}])$$

According to the previous definition, it is straightforward to observe that the given PFD has to hold, by considering only a subset of *mtr*, composed of tuples consistent with the considered time frame, the modality, and the range. Such subset is called *time-frame relation view* (*TF*-view). More formally, the *TF*-view w is defined as $w = TFv(mtr, \alpha, m, [i_1, i_2]) \equiv \{t \mid t \in mtr \wedge \Theta(t, \alpha, m, [i_1, i_2])\}$. Hereinafter, we will consider a time-frame $\alpha = \langle 6, 2, 10\rangle$, $m =$ '*fixed*', and an extended semantics, i.e., considering the range $[1, k]$.

▶ **Example 8.** Let us consider the *mtr* depicted in Table 1. Tuples #10, #11, and #36 are out of the time frame α. It is straightforward to observe that the PFD $\overline{Drug}^3 \xrightarrow[\alpha,m]{} \dot{AKI}$ holds. On the other side, PFDs $\overline{HR}^1, \overline{SpO_2}^2 \xrightarrow[\alpha,m]{} \dot{AKI}$, $\overline{HR}^1 \xrightarrow[\alpha,m]{} \dot{AKI}$ and $\overline{SpO_2}^2 \xrightarrow[\alpha,m]{} \dot{AKI}$ do not hold.

3.5 Discovering Approximate PFDs

To mine PFDs in a generic multi-temporal relation we have first to isolate those tuples that fit, with respect to a given modality and to a given semantics, the considered temporal frame, composed of OW, WW, and PW. As a second step, we need to deal with some kind of approximation, as it could happen that some PFDs hold on a subset of tuples of the time-frame relation view, we consider. Thus, we have to evaluate whether considering such subset is acceptable with respect to the prediction task supported by the considered PFDs.

In other words, we require a PFD f to be satisfied by most tuples of the *TF*-view w. A very small portion of tuples of w is allowed to violate the dependency. In the context of predictive functional dependencies, we consider one of the measures proposed in [18] and introduce two other error measures, specifically tailored to the predictive purpose of approximate PFDs.

Given a *TF*-view $w \subseteq mtr$, the first error measure G_3 considers the minimum number of tuples in w to be deleted to obtain a relation s where the given FD holds [18]. In our context, it is expressed according to the following definition.

▶ **Definition 9** (Error measure G_3)**.** *Given a TF-view* $w = TFv(mtr, \alpha, m, [1, k])$ *of an mt-relation mtr with schema* $Z\overline{U}^1\overline{U}^2..\overline{U}^k\dot{B} \cup \{\overline{VT}^1, \overline{VT}^2, .., \overline{VT}^k, \dot{VT}\}$*, and a PFD* $S\overline{P}^h\overline{Q}^i ...\overline{R}^j \xrightarrow[\alpha,m]{} \dot{Y}$*, where* $S \subseteq Z, \overline{P}^h \subseteq \overline{U}^h, \overline{Q}^i \subseteq \overline{U}^i, \overline{R}^j \subseteq \overline{U}^j$ *and* $\dot{Y} \subseteq \dot{U}$*, and any relation* $s \subseteq w$*, such that* $s \models^{E}_{\alpha,m} S\overline{P}^h\overline{Q}^i ...\overline{R}^j \to \dot{Y}$*, the error measure G3 is expressed as:* $G_3 = |w| - |s|$*. The related scaled measurement* g_3 *is defined as:* $g_3 = \frac{G_3}{|w|}$*.*

Let us now introduce some new kinds of error, which may be of interest in the context of prediction. The first issue is in considering another error, no longer focused on the number of tuples that we have to delete to satisfy the PFD, but focused on the number of entities that we accept to discard for the sake of the PFD. The new error measure H_3 permits, for example, to disregard data of entities with a very low number of tuples, which could create noise in our dataset.

▶ **Definition 10** (Error measure H_3). *Given a TF-view $w = TFv(mtr, \alpha, m, [1, k])$ of an mt-relation mtr with schema $Z\overline{U}^1\overline{U}^2..\overline{U}^k\dot{B} \cup \{\overline{VT}^1, \overline{VT}^2, .., \overline{VT}^k, \dot{VT}\}$, and a PFD $S\overline{P}^h\overline{Q}^i ...\overline{R}^j \xrightarrow[\alpha,m]{} \dot{Y}$, where $S \subseteq Z, \overline{P}^h \subseteq \overline{U}^h, \overline{Q}^i \subseteq \overline{U}^i, \overline{R}^j \subseteq \overline{U}^j$ and $\dot{Y} \subseteq \dot{U}$, and any relation $s \subseteq w$, such that $s \models^E_{\alpha,m} S\overline{P}^h\overline{Q}^i ...\overline{R}^j \to \dot{Y}$, the error measure H3 is expressed as: $H_3 = |\{t[Z] \mid \exists t \in w\}| - |\{t[Z] \mid \exists t \in s\}|$ The related scaled measurement h_3 is defined as: $h_3 = \frac{H_3}{|\{t[Z]|\exists t \in w\}|}$.*

Finally, considering the number of tuples for each entity we accept to discard to satisfy the PFD, we formalize a last error measure, namely J_3. It ensures to maintain enough "consistent" information for each entity.

▶ **Definition 11** (Error measure J_3). *Given a TF-view $w = TFv(mtr, \alpha, m, [1, k])$ of an mt-relation mtr with schema $Z\overline{U}^1\overline{U}^2..\overline{U}^k\dot{B} \cup \{\overline{VT}^1, \overline{VT}^2, .., \overline{VT}^k, \dot{VT}\}$, a PFD $S\overline{P}^h\overline{Q}^i ...\overline{R}^j \xrightarrow[\alpha,m]{} \dot{Y}$, where $S \subseteq Z, \overline{P}^h \subseteq \overline{U}^h, \overline{Q}^i \subseteq \overline{U}^i, \overline{R}^j \subseteq \overline{U}^j$ and $\dot{Y} \subseteq \dot{U}$, and any relation $s \subseteq w$, such that $s \models^E_{\alpha,m} S\overline{P}^h\overline{Q}^i ...\overline{R}^j \to \dot{Y}$, the error measure J3 is expressed as in the following.*

Let $w_{[v]} \equiv \{t[Z]|t \in w \wedge t[Z] = v\}$ and $s_{[v]} \equiv \{t[Z]|t \in s \wedge t[Z] = v\}$, then

$$J_3 = \max_{(v \in \{t[Z]|t \in s\})} \{|w_{[v]}| - |s_{[v]}|\}$$

The related scaled measurement j_3 is defined as follows:

$$j_3 = \max_{(v \in \{t[Z]|t \in s\})} \{\frac{|w_{[v]}| - |s_{[v]}|}{|w_{[v]}|}\}$$

According to the introduced error measures, we are now able to define an approximate predictive functional dependency as follows:

▶ **Definition 12** (Approximate Predictive Functional Dependency (APFD)). *Given a TF-view $w = TFv(mtr, \alpha, m, [1, k])$ of an mt-relation mtr with schema $Z\overline{U}^1\overline{U}^2..\overline{U}^k\dot{B} \cup \{\overline{VT}^1, \overline{VT}^2, .., \overline{VT}^k, \dot{VT}\}$, w fulfills the APFD*

$$S\overline{P}^h\overline{Q}^i ...\overline{R}^j \xrightarrow[\alpha,m]{\varepsilon} \dot{Y}$$

(written as $w \models^E_{\alpha,m} S\overline{P}^h\overline{Q}^i ...\overline{R}^j \xrightarrow{\varepsilon} \dot{Y}$) , where $\varepsilon = \langle \varepsilon_g, \varepsilon_h, \varepsilon_j \rangle$ and $S \subseteq Z, \overline{P}^h \subseteq \overline{U}^h, \overline{Q}^i \subseteq \overline{U}^i, \overline{R}^j \subseteq \overline{U}^j$, $\dot{Y} \subseteq \dot{U}$, if a relation $s \subseteq w$ exists such that $s \models^E_{\alpha,m} S\overline{P}^h\overline{Q}^i ...\overline{R}^j \to \dot{Y}$ with $g_3 \leq \varepsilon_g \wedge h_3 \leq \varepsilon_h \wedge j_3 \leq \varepsilon_j$. In other words, $\varepsilon_g, \varepsilon_h, \varepsilon_j$ are the maximum acceptable errors defined by the user for g_3, h_3, and j_3, respectively.

▶ **Example 13.** Suppose that our final goal is to preserve at least the 75% of the tuples ($\varepsilon_g = 0.25$), the 80% of the patients ($\varepsilon_h = 0.2$), and the 50% of the tuples for each patient ($\varepsilon_j = 0.5$). In Table 1, the PFD $\overline{HR}^1, \overline{SpO_2}^2 \xrightarrow[\alpha,m]{} \dot{AKI}$ is satisfied by considering a (sub)instance s by deleting tuples #2 and #9 . Thus, in this case, $g_3 = 2/9$, $h_3 = 1/3$, as any tuples for patient Stevie disappear; and $j_3 = 1/4$ as we delete a tuple of Daisy. It is easy to see that $g_3 < \varepsilon_g$, $h_3 > \varepsilon_h$, while $j_3 < \varepsilon_j$. On the other side, if we consider the instance

s', by deleting tuples #2 and #4, we would observe that the PFD is still satisfied, while $g_3 = 2/9$, $h_3 = 0/3$, and $j_3 = 2/4$. In this case, all the errors are below or equal to the given thresholds. Thus, we can say that $w \models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \xrightarrow{\epsilon} \dot{AKI}$ with $\epsilon \equiv \langle 0.35, 0.2, 0.5\rangle$.

If we set the error thresholds as $\varepsilon_g = 0.25$, $\varepsilon_h = 0.4$, and $\varepsilon_j = 0.3$ (mainly we accept to discard some more patients, but we increase the number of tuples per patient we want to preserve), we can observe that $s \models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \rightarrow \dot{AKI}$, while $s' \not\models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \rightarrow \dot{AKI}$. Thus, $w \models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \xrightarrow{\epsilon} \dot{AKI}$ also with $\epsilon \equiv \langle 0.35, 0.4, 0.3\rangle$.

It is easy to prove that if $w \models^E_{\alpha,m} S\overline{P}^h\overline{Q}^i \ ...\overline{R}^j \xrightarrow{\varepsilon} \dot{Y}$, it will also hold $w \models^E_{\alpha,m} SS_1\overline{P}^h\overline{P}^h_1\overline{Q}^i\overline{Q}^i_1 \ \overline{V}^x...\overline{R}^j\overline{R}^j_1 \xrightarrow{\varepsilon} \dot{Y}$, where $S_1 \subseteq Z, \overline{P}^h_1 \subseteq \overline{U}^h, \overline{Q}^i_1 \subseteq \overline{U}^i, \overline{R}^j_1 \subseteq \overline{U}^j$, $\overline{V}^x \subseteq \overline{U}^x$ with $i < x < j$.

As an example, as $w \models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \xrightarrow{\varepsilon} \dot{AKI}$ for the *TF*-view w depicted in Table 1, it is also the case that $w \models^E_{\alpha,m} Patient, \overline{HR}^1, \overline{SpO_2}^2 \xrightarrow{\varepsilon} \dot{AKI}$. After adding the new attribute *Patient* in the antecedent, nothing changes for mt-relation $s \subseteq w$, for which $\overline{HR}^1, \overline{SpO_2}^2 \rightarrow \dot{AKI}$ holds, independently from the values of attribute *Patient*.

As we are interested in finding the minimum predictive attribute set, here we introduce the definition of minimal APFDs as follows:

▶ **Definition 14** (Minimal APFD). *An APFD* $S\overline{P}^h\overline{Q}^i \ ...\overline{R}^j \xrightarrow[\alpha,m]{\varepsilon} \dot{Y}$ *is minimal for* w*, if* $w \models^E_{\alpha,m} S\overline{P}^h\overline{Q}^i \ ...\overline{R}^j \xrightarrow{\varepsilon} \dot{Y}$ *and* $\forall\, \overline{V} \subset S\overline{P}^h\,\overline{Q}^i ...\overline{R}^j$ *we have that* $w \not\models^E_{\alpha,m} \overline{V} \xrightarrow{\varepsilon} \dot{Y}$.

Minimal APFDs provide the most compact representation of the existing dependencies.

▶ **Example 15.** Considering the *mt*-relation w depicted in Table 1, it is straightforward to observe that the following two APFDs hold for $\epsilon \equiv \langle 0.25, 0.4, 0.4\rangle$ and are minimal.

$$w \models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \xrightarrow{\varepsilon} \dot{AKI}, w \models^E_{\alpha,m} \overline{Drug}^3 \xrightarrow{\varepsilon} \dot{AKI}$$

As for the minimality of the first APFD, both $\overline{SpO_2}^2 \xrightarrow[\alpha,m]{\varepsilon} \dot{AKI}$ and $\overline{HR}^1 \xrightarrow[\alpha,m]{\varepsilon} \dot{AKI}$ cannot satisfy the first threshold, i.e., $g_3 \leq 0.25$.

4 The (data) complexity of deriving an APFD

As we said before, to obtain a set $s \subseteq w$ which satisfies an APFD, we have to consider the three different thresholds.

We reduced the problem in hand to a general 3*SAT* problem, showing that checking an APFD considering all the three thresholds belongs to the class *NP*.

Before starting with the theoretical analysis let us recall that an instance of *SAT* problem is a logical formula formed by a conjunction of disjunctive clauses. Namely, each clause is a disjunction of literals, and the general formula is a conjunction of disjunctive clauses. Therefore, an instance of *SAT* is a conjunction of clauses, each of them representable as a set of literals. In the specific case of 3*SAT*, each clause has exactly 3 literals [23].

Let us now introduce a simple relation representing any mt-relation. To discuss the complexity of checking an APFD, it is enough to consider a relation having a single attribute (Z) representing the entity attribute, a single attribute (A) representing the antecedent, the predicted attribute ($\dot{B}$). Moreover, let us assume that the domain of all attributes is $\mathcal{N}$ or a subset of it (the predicted values for $\dot{B}$ will be either 0 or 1, to represent boolean values). Thus, we will consider a relation w with schema $W(A, \dot{B}, Z)$. Before introducing the two problems and then proving the NP-hardness of checking APFDs by a suitable reduction to an NP problem, let us introduce a simple reformulation of the satisfaction of error thresholds for G_3 and H_3 by a relation w in terms of conflict resolution (in the following we will make use of the standard projection operation π of relational algebra).

▶ **Definition 16.** *Given a relation $w \subset \mathbb{N}^3$, a natural number $0 \leq k < |w|$, and a natural number $0 \leq h < |\pi_Z(w)|$ we say that w admits a* conflict resolution *of order (k, h) if there exists a subset $w^- \subseteq w$ such that:*
1. $|w^-| \leq k$
2. *for every pair of triplets $(a, \dot{b}, z), (a', \dot{b}', z') \in w \setminus w^-$ if $a = a'$ then $\dot{b} = \dot{b}'$;*
3. $|\pi_Z(w)| - |\pi_Z(w \setminus w^-)| \leq h$.

According to the introduced simplified form of mt-relation and the previous definition of conflict resolution, we may now represent the problem of checking an APFD as in the following. It is worth noting that the order (k, h) of the conflict resolution represents the thresholds for errors G_3 and H_3, respectively.

▶ **Problem 1.** Given a relation $w \subset \mathbb{N}^3$, a natural number $0 \leq k < |w|$, and a natural number $0 \leq h < |\pi_Z(w)|$ determine whether or not w admits a *conflict resolution* of order (k, h).

Now, we introduce the problem, well-known in the literature, we will use for the reduction.

▶ **Problem 2.** Given an instance C of 3SAT in which each clause features only *positive* literals, $C = \{\{a_1^1, a_2^1, a_3^1\}, \ldots, \{a_1^n, a_2^n, a_3^n\}\}$, with variable set $\mathcal{A} = \{a_j^i : 1 \leq i \leq n, 1 \leq j \leq 3\}$, and a number $0 \leq p < |C|$ determine whether or not there exists an assignment $\sigma : \mathcal{A} \to \{0, 1\}$[1] such that $|\{i : \sigma(a_1^i) = \sigma(a_2^i) = \sigma(a_3^i)\}| \leq p$ and C is satisfied.

For the sake of brevity, given a clause $\{a_1^i, a_2^i, a_3^i\}$ in $C = \{\{a_1^1, a_2^1, a_3^1\}, \ldots, \{a_1^n, a_2^n, a_3^n\}\}$ and an assignment $\sigma : \mathcal{A} \to \{0, 1\}$ we say that $\{a_1^i, a_2^i, a_3^i\}$ is homogeneous w.r.t σ, or simply *homogeneous* when σ is clear from the context, if and only if $\sigma(a_1^i) = \sigma(a_2^i) = \sigma(a_3^i)$. Then, Problem 2 may be equivalently redefined as: given a set of clauses $C = \{\{a_1^1, a_2^1, a_3^1\}, \ldots, \{a_1^n, a_2^n, a_3^n\}\}$ deciding whether or not there exists an assignment σ for the variables in C that makes C satisfied and at most p clauses of C homogeneous w.r.t σ.

The complexity of Problem 2 is well known, as in the following theorem.

▶ **Theorem 17.** *Problem 2 is NP-Complete [23].*

The following theorem proves that checking an APFD according to the introduced error thresholds is NP-hard.

▶ **Theorem 18.** *Problem 1 is NP-Hard.*

Proof. The proof is by reduction from Problem 2 and is reported in Appendix A. ◀

Proved that the Problem 1 is *NP-Hard*, it is now necessary to find a deterministic algorithm that could stop the analysis of a relation, as soon as it verifies that the relation cannot satisfy the given APFD. Algorithm 1 provides the pseudo-code of such algorithm. The general idea of this algorithm is searching for a solution considering one tuple at a time, until it is possible to generate a solution, which satisfies the selected thresholds. Throughout the

[1] here 0 and 1 represent the logical values false and true, respectively.

Algorithm 1 DeterministicADC.

Input: an instance w of the relation W, and three real numbers ϵ_{g_3}, ϵ_{h_3}, and ϵ_{j_3} in $[0, 1]$
Output: a relation $s \subseteq w$ s.t. $s \models A \to \dot{B}$, $g_3(w, s) \geq 1 - \epsilon_{g_3}$, $h_3(w, s) \geq 1 - \epsilon_{h_3}$, $j_3(w, s) \geq 1 - \epsilon_{j_3}$

▷ Prepare data for initial call according to epsilons
begin
  $del \leftarrow \lfloor \epsilon_{g_3} |w| \rfloor$
  $count \leftarrow \epsilon_{h_3} \lfloor |\pi_Z(w)| \rfloor$
  for $z \in \pi_Z(w)$: **do**
    $thresholds[z] \leftarrow \lfloor \epsilon_{j_3} |\sigma_{Z=z}(w)| \rfloor$
  return RecADC($w, del, count, thresholds$)
Function RecADC($w, del, count, thresholds$):
  ▷ This is the last recursive call before success
  if $w = \emptyset$ **then**
    return $\emptyset$
  let $a \in \pi_A(w)$
  ▷ For each value of B
  for $boolean\_val \in \{0, 1\}$ **do**
    ▷ del_tuples: tuples removed according to selection
    $del\_tuples \leftarrow \sigma_{A=a \wedge \dot{B}=boolean\_val}(w)$
    $s \leftarrow \sigma_{A=a \wedge \dot{B}=\neg boolean\_val}(w)$
    $out \leftarrow \{\}$
    for $z \in \pi_Z(del\_tuples)$: **do**
      $thresholds'[z] \leftarrow thresholds[z] - |\sigma_{Z=z}(del\_tuples)|$
      if $thresholds'[z] < 0 \leq thresholds[z]$ **then**
        $out \leftarrow out \cup \{z\}$
    ▷ out: the z groups that must disappear, since their tuples passed below the threshold ϵ_{j_3} in the current state
    if $count - |out| \geq 0$ **then**
      ▷ $count'$: represent the z groups still to be considered
      $count' \leftarrow count - |out|$
      $del\_tuples \leftarrow del\_tuples \cup \sigma_{Z=z:z \in out}(w)$
      if $del - |del\_tuples| \geq 0$ **then**
        ▷ If the final test succeeds, we proceed with the recursive call on the updated values
        $del' \leftarrow del - |del\_tuples|$
        $w' \leftarrow w \setminus (del\_tuples \cup s)$
        $s' \leftarrow$ RecADC($w', del', count', thresholds'$)
        if $s' \neq$ ***fail*** **then**
          return $s \cup s'$
  return fail

code, w is the entire relation. del, $count$, $thresholds$ represent the counters that control the errors. del counts the number of remaining tuples, $count$ controls the number of remaining entities, and $thresholds$ verifies the number of remaining tuples for each entity. After a trivial check about the (non) emptiness of relation w, for each value $a \in \pi_A(w)$, we try one boolean value and verify the dependency, if it fails, we try the second boolean value and verify the dependency. If both choices failed, then the algorithm fails. If one of the boolean values satisfies the thresholds, we update the counters, building at every step an intermediate relation s', as long as the thresholds are satisfied.

5 Deriving APFDs: an experimental evaluation

Here, we provide some results from an experimental evaluation on real-world clinical data. We derived APFDs by using a simpler, even sub-optimal, mining algorithm.

5.1 Computing APFDs

As for the first experimental evaluations of the proposed approach, we adopted a sub-optimal solution, on top of the well-known TANE [13] algorithm, a popular approximate functional dependency detection algorithm, customizing it to mine only approximate functional dependencies with a fixed consequent, the predicted attribute $\dot{Y}$.

To find all minimal non-trivial dependencies, TANE works as follows. It starts the search from singleton sets of attributes and works its way to larger attribute sets through the set containment lattice level by level. When the algorithm is processing a set X, it tests dependencies of the form $X \backslash A \rightarrow A$, where $A \in X$. This guarantees that only non-trivial dependencies are considered. In our proposal, we compute all the Approximate Predictive Functional Dependencies, considering the three errors, g_3, h_3, j_3.

Given *TF*-view w and the predicted attribute $\dot{Y}$, our approach was mainly based on the following steps:

- Derive s by TANE, such that $g_3 \leq \varepsilon_g$;
- Check on s that $h_3 \leq \varepsilon_h$;
- If the previous check is fine, check that $j_3 \leq \varepsilon_j$.

It is easy to observe that this approach, while extracting APFDs that are satisfied by w according to the given thresholds, could exclude other APFDs that are associated to some s, which is not maximal, i.e., minimal with respect to g_3, but still satisfies $g_3 \leq \varepsilon_g$. And such s could satisfy also the other thresholds.

It is well known that the complexity of deriving AFDs is exponential in the number of attributes [13, 19], while the complexity of checking a single dependency is linear in the number of tuples (data complexity). In our experiments, even though the "maximality" of s is related to a composite error threshold $\varepsilon = < \varepsilon_g, \varepsilon_h, \varepsilon_j >$ and many possible relations s would be derived to evaluate a single APFD –making the data complexity higher as shown in the previous section–, the data complexity remains linear, as we rely on TANE, and check only further thresholds.

5.2 Dataset and data transformation

Our proposal has been applied to the clinical domain of the Intensive Care Unit (ICU) using the MIMIC III (Medical Information Mart for Intensive Care) [16] dataset, with the aim of finding significant APFDs for the AKI diagnosis. MIMIC III is a freely accessible relational database of de-identified patients, hospitalized in the intensive care units at Beth Israel Deaconess Medical Center between 2001 and 2012.

The data are associated with more than 46 000 patients and almost 60 000 admissions. The information contained in the database includes demographics, vital sign measures (such as heart rate, systolic and diastolic pressures, oxygen saturation, and body temperature) registered at the bedside, laboratory test results, administered drugs, medications and procedures.

From the original dataset, we used seven tables, transformed through an ETL (Extract, Transform, Load) process. *D_ITEMS* and *D_LABITEMS* were the reference tables needed to label every measure related to a patient. *PATIENTS* and *ICUSTAYS* were used to retrieve information about the admission and discharge from the ICU and the age. *PRESCRIPTIONS* provided information about the administered medications. We mainly considered four categories: diuretics, Non-steroidal anti-inflammatory drugs (NSAID), radiocontrast agents, and angiotensin. *LABEVENTS* was used to extract information about serum creatinine and urine and *CHARTEVENTS* for heart rate, diastolic pressure and oxygen saturation. We categorized the numerical variables into “low, medium, high” according to clinical literature.

We considered two 3-window settings. The first one was characterized by an OW of 72 hours, a WW of 12 hours, and then a PW of 36 hours, where there is the (possible) onset of the illness according to one of the KDIGO criteria. The second one was characterized by an OW of 120 hours, a WW of 12 hours, and a PW of 36 hours. Starting from the literature [33], we considered six measures: creatinine, administered drugs, respiratory rate, oxygen saturation, and diastolic blood pressure. From a cohort of 50.711 patients, we considered three different *TF*-views:

- *TF*-view #1, with four states of the same measure (serum creatinine) to build a sequence of four values of a measure, where any value is the next of the preceding one (if any), within the first 3-window setting. In this case, we obtain 2546 subjects (1878 patients without AKI, 668 patients with AKI) with 3839 rows;
- *TF*-view #2, with four states of the same measure (administered drugs) to build a sequence of four values of a measure, where any value is the next of the preceding one (if any), within the second 3-window setting. In this case, we obtain 148 subjects (109 patients without AKI, 39 patients with AKI) with 1047 rows;
- *TF*-view #3 with four states, each one related to a different measure (administered drug, diastolic blood pressure, respiratory rate, oxygen saturation) with $\overline{VT}^k = \overline{VT}^{k-1} + 1$ for $k = 1, .., 3$ within the second 3-window setting. In this case, we have 413 subjects (305 patients without AKI, 108 patients with AKI) with 193.173 rows.

With the two 3-window settings, we achieved similar results. First of all, the error values were completely comparable between the two settings. Secondly, we recorded a similar trend in all the *TF-views*. Indeed, the temporal states kept dropping until the results of functional dependencies consisted of a single antecedent, with the increase of error ϵ.

Regarding serum creatinine, our experiments suggested that creatinine needed a medium-long history to provide predictive patterns, so considering the 4 measures the difference in terms of error between functional dependencies that had more than one antecedent state, and those that had only one state, was very small. With six measures we were able to have temporal patterns formed by more than one state.

In Table 2, we reported some of the APFDs obtained through the algorithm, with the corresponding error thresholds. The algorithm took a few minutes for each *TF-view* to extract these APFDs.

During the experimental evaluation, we observed that data related to some patients are completely discarded when mining APFDs. Indeed, dealing with a large population, whatever the entity under study, it may be common to completely discard some (entity) outliers.

6 Conclusions

In this paper, we introduced a 3-window framework for the specification and evaluation of Approximate Predictive Functional Dependencies, dealing with the capability of exploiting data dependencies for the prediction task. The declarative framework, which we represented through relational calculus queries and formulas, allows one to consider different kinds of anchored and unanchored time windows.

Table 2 APFDs from the three *TF*-views.

APFD	ε_g	ε_h	ε_j	*TF*-view
$\overline{Creat}^1, \overline{Creat}^3 \rightarrow A\dot{K}I$	27.45%	27%	50%	#1
$\overline{Creat}^1, \overline{Creat}^4 \rightarrow A\dot{K}I$	27.45%	27%	50%	#1
$\overline{Drug}^1, \overline{Drug}^2, \overline{Drug}^4 \rightarrow A\dot{K}I$	21%	30%	50%	#2
$\overline{Drug}^1, \overline{Drug}^2, \overline{Drug}^4 \rightarrow A\dot{K}I$	21%	30%	80%	#2
$\overline{Drug}^1, \overline{Drug}^2, \overline{Drug}^3 \rightarrow A\dot{K}I$	21%	30%	80%	#2
$\overline{Drug}^1, \overline{Drug}^3, \overline{Drug}^4 \rightarrow A\dot{K}I$	21%	30%	80%	#2
$\overline{Drug}^1, \overline{RespRate}^3 \rightarrow A\dot{K}I$	10%	51%	75%	#3
$\overline{RespRate}^3 \rightarrow A\dot{K}I$	30%	75%	75%	#3
$\overline{Drug}^1 \rightarrow A\dot{K}I$	30%	75%	75%	#3
$\overline{Spo_2}^4 \rightarrow A\dot{K}I$	30%	75%	75%	#3

Such dependencies have been specified with respect to three different kinds of error related to: the number of tuples to be deleted for having the corresponding PFD holding, the number of entities having all tuples deleted for having the corresponding PFD holding, and the number of tuples we admit to discard for any entity.

We also discussed the computational aspects related to the extraction of APFDs. We detailed a theoretical analysis of the complexity to derive a relation $s \subseteq w$ considering the error thresholds G_3 and H_3. We reduced the problem in hand to a general $3SAT$ problem, showing that checking an APFD considering all the three thresholds belongs to the class NP.

We applied our approach to real clinical data, specifically to MIMIC III dataset, obtaining results that demonstrate the applicability of this new type of temporal pattern mining in medicine, but also in other contexts where the core of the problem is finding temporal patterns in the past associated, in a prediction-oriented approach, to following (future) events.

References

1. Ziawasch Abedjan, Cuneyt Gurcan Akcora, Mourad Ouzzani, Paolo Papotti, and Michael Stonebraker. Temporal rules discovery for web data cleaning. *Proc. VLDB Endow.*, 9(4):336–347, 2015. `doi:10.14778/2856318.2856328`.
2. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. URL: `http://webdam.inria.fr/Alice/`.
3. Beatrice Amico and Carlo Combi. A 3-window framework for the discovery and interpretation of predictive temporal functional dependencies. In Martin Michalowski, Syed Sibte Raza Abidi, and Samina Abidi, editors, *Artificial Intelligence in Medicine - 20th International Conference on Artificial Intelligence in Medicine, AIME 2022, Halifax, NS, Canada, June 14-17, 2022, Proceedings*, volume 13263 of *Lecture Notes in Computer Science*, pages 299–309. Springer, 2022. `doi:10.1007/978-3-031-09342-5_29`.
4. Laure Berti-Équille, Hazar Harmouch, Felix Naumann, Noël Novelli, and Saravanan Thirumuruganathan. Discovery of genuine functional dependencies from relational data with missing values. *Proc. VLDB Endow.*, 11(8):880–892, 2018. `doi:10.14778/3204028.3204032`.
5. Claudio Bettini, Sushil Jajodia, and Sean Wang. *Time granularities in databases, data mining, and temporal reasoning*. Springer Science & Business Media, 2000.
6. Loredana Caruccio, Vincenzo Deufemia, Felix Naumann, and Giuseppe Polese. Discovering relaxed functional dependencies based on multi-attribute dominance. *IEEE Trans. Knowl. Data Eng.*, 33(9):3212–3228, 2021. `doi:10.1109/TKDE.2020.2967722`.
7. Loredana Caruccio, Vincenzo Deufemia, and Giuseppe Polese. Relaxed functional dependencies - A survey of approaches. *IEEE Trans. Knowl. Data Eng.*, 28(1):147–165, 2016. `doi:10.1109/TKDE.2015.2472010`.

8 Carlo Combi, Matteo Mantovani, Alberto Sabaini, Pietro Sala, Francesco Amaddeo, Ugo Moretti, and Giuseppe Pozzi. Mining approximate temporal functional dependencies with pure temporal grouping in clinical databases. *Comput. Biol. Medicine*, 62:306–324, 2015. doi:10.1016/j.compbiomed.2014.08.004.

9 Carlo Combi, Angelo Montanari, and Pietro Sala. A uniform framework for temporal functional dependencies with multiple granularities. In *International Symposium on Spatial and Temporal Databases*, pages 404–421. Springer, 2011.

10 Carlo Combi and Pietro Sala. Mining approximate interval-based temporal dependencies. *Acta Informatica*, 53(6-8):547–585, 2016. doi:10.1007/s00236-015-0246-x.

11 Abdur Rahim Mohammad Forkan and Ibrahim Khalil. A clinical decision-making mechanism for context-aware and patient-specific remote monitoring systems using the correlations of multiple vital signs. *Computer methods and programs in biomedicine*, 139:1–16, 2017. doi:10.1016/j.cmpb.2016.10.018.

12 Chris Giannella and Edward Robertson. On approximation measures for functional dependencies. *Inf. Syst.*, 29(6):483–507, August 2004. doi:10.1016/j.is.2003.10.006.

13 Yka Huhtala, Juha Kärkkäinen, Pasi Porkka, and Hannu Toivonen. Tane: An efficient algorithm for discovering functional and approximate dependencies. *The computer journal*, 42(2):100–111, 1999. doi:10.1093/comjnl/42.2.100.

14 Christian S. Jensen and Richard T. Snodgrass. Valid time. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems, Second Edition*. Springer, 2018. doi:10.1007/978-1-4614-8265-9_1066.

15 Christian S Jensen, Richard T Snodgrass, and Michael D Soo. Extending existing dependency theory to temporal databases. *IEEE Transactions on Knowledge and Data Engineering*, 8(4):563–582, 1996. doi:10.1109/69.536250.

16 Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016. doi:10.1038/sdata.2016.35.

17 Arif Khwaja. Kdigo clinical practice guidelines for acute kidney injury. *Nephron Clinical Practice*, 120(4):c179–c184, 2012.

18 Jyrki Kivinen and Heikki Mannila. Approximate inference of functional dependencies from relations. *Theor. Comput. Sci.*, 149(1):129–149, 1995. doi:10.1016/0304-3975(95)00028-U.

19 Sebastian Kruse and Felix Naumann. Efficient discovery of approximate dependencies. *Proc. VLDB Endow.*, 11(7):759–772, 2018. doi:10.14778/3192965.3192968.

20 Ohbyung Kwon and Jae Mun Sim. Effects of data set features on the performances of classification algorithms. *Expert Systems with Applications*, 40(5):1847–1857, 2013. doi:10.1016/j.eswa.2012.09.017.

21 Marie Le Guilly, Jean-Marc Petit, and Vasile-Marian Scuturici. Evaluating classification feasibility using functional dependencies. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XLIV*, pages 132–159. Springer, 2020. doi:10.1007/978-3-662-62271-1_5.

22 Mirjana Mazuran, Elisa Quintarelli, Letizia Tanca, and Stefania Ugolini. Semi-automatic support for evolving functional dependencies. In Evaggelia Pitoura, Sofian Maabout, Georgia Koutrika, Amélie Marian, Letizia Tanca, Ioana Manolescu, and Kostas Stefanidis, editors, *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016*, pages 293–304. OpenProceedings.org, 2016. doi:10.5441/002/edbt.2016.28.

23 Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991. doi:10.1016/0022-0000(91)90023-X.

24 Parivash Pirasteh, Slawomir Nowaczyk, Sepideh Pashami, Magnus Löwenadler, Klas Thunberg, Henrik Ydreskog, and Peter Berck. Interactive feature extraction for diagnostic trouble codes

in predictive maintenance: A case study from automotive domain. In *Proceedings of the Workshop on Interactive Data Mining*, pages 1–10, 2019. doi:10.1145/3304079.3310288.

25 Pietro Sala, Carlo Combi, Matteo Mantovani, and Romeo Rizzi. Discovering evolving temporal information: Theory and application to clinical databases. *SN Comput. Sci.*, 1(3):153, 2020. doi:10.1007/s42979-020-00160-9.

26 Philipp Schirmer, Thorsten Papenbrock, Sebastian Kruse, Felix Naumann, Dennis Hempfing, Torben Mayer, and Daniel Neuschäfer-Rube. Dynfd: Functional dependency discovery in dynamic datasets. In Melanie Herschel, Helena Galhardas, Berthold Reinwald, Irini Fundulaki, Carsten Binnig, and Zoi Kaoudi, editors, *Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019*, pages 253–264. OpenProceedings.org, 2019. doi:10.5441/002/edbt.2019.23.

27 Robert W Schrier, Wei Wang, Brian Poole, Amit Mitra, et al. Acute renal failure: definitions, diagnosis, pathogenesis, and therapy. *The Journal of clinical investigation*, 114(1):5–14, 2004. doi:doi:10.1172/JCI22353.

28 Shigehiko Uchino, Rinaldo Bellomo, Donna Goldsmith, Samantha Bates, and Claudio Ronco. An assessment of the rifle criteria for acute renal failure in hospitalized patients. *Critical care medicine*, 34(7):1913–1917, 2006. doi:10.1097/01.CCM.0000224227.70642.4F.

29 Victor Vianu. Dynamic functional dependencies and database aging. *Journal of the ACM (JACM)*, 34(1):28–59, 1987. doi:10.1145/7531.7918.

30 Jef Wijsen. Design of temporal relational databases based on dynamic and temporal functional dependencies. In James Clifford and Alexander Tuzhilin, editors, *Recent Advances in Temporal Databases, Proceedings of the International Workshop on Temporal Databases, Zürich, Switzerland, 17-18 September 1995*, Workshops in Computing, pages 61–76. Springer, 1995. doi:10.1007/978-1-4471-3033-8_4.

31 Jef Wijsen. Temporal fds on complex objects. *ACM Trans. Database Syst.*, 24(1):127–176, 1999. doi:10.1145/310701.310715.

32 Jef Wijsen. *Temporal Dependencies*, pages 3955–3961. Springer, 2018. doi:10.1007/978-1-4614-8265-9_396.

33 Zhenxing Xu, Jingyuan Chou, Xi Sheryl Zhang, Yuan Luo, Tamara Isakova, Prakash Adekkanattu, Jessica S Ancker, Guoqian Jiang, Richard C Kiefer, Jennifer A Pacheco, et al. Identifying sub-phenotypes of acute kidney injury using structured and unstructured electronic health record data with memory networks. *Journal of biomedical informatics*, 102:103361, 2020. doi:10.1016/j.jbi.2019.103361.

A Data Complexity

In this Appendix, we first provide the proof of Theorem 18 and then discuss some algorithmic issues.

Proof of Theorem 18. The proof is by reduction from Problem 2. Let $C = \{\{a_1^1, a_2^1, a_3^1\}, \ldots, \{a_1^n, a_2^n, a_3^n\}\}$ and p an instance of Problem 2. We introduce the following relation $w_C = \{(a_j^i, 0, 2i) : 1 \leq i \leq n, 1 \leq j \leq 3\} \cup \{(a_j^i, 1, 2i+1) : 1 \leq i \leq n, 1 \leq j \leq 3\}$. It is easy to observe that $|w_C| = 6|C|$ and w_C may be generated in polynomial space from C. Let us define a function *clause* : $w_C \to \{1, \ldots, n\}$ defined as:

$$clause(a_j^i, \dot{b}, z) = \begin{cases} \frac{z}{2} & \text{if } z \text{ is even} \\ \frac{(z-1)}{2} & \text{otherwise} \end{cases} .$$

Let us observe that function *clause* is well-defined and maps each element $(a_j^i, \dot{b}, z) \in w_C$ to the index of the clause which corresponds to it in the above construction. Now we prove that (C, p) is a positive instance of Problem 2 if and only if $(w_C, |w_C|, p)$ is a positive instance of Problem 1.

For the left-to-right direction, let us assume that $C = \{\{a_1^1, a_2^1, a_3^1\}, \dots, \{a_1^n, a_2^n, a_3^n\}\}$ and p is a positive instance of Problem 2. Let $\mathcal{A}$ be the set of all and only variables which appear in C. Thus, there exists an assignment $\sigma : \mathcal{A} \to \{0,1\}$ and at most p distinct indexes $i_1, \dots i_p$ such that $\sigma(a_1^{i_k}) = \sigma(a_2^{i_k}) = \sigma(a_3^{i_k})$ for each $1 \le k \le p$. Let $w_C^- = \{(a_j^i, 1, 2i) : \sigma(a_j^i) = 0\} \cup \{(a_j^i, 0, 2i+1) : \sigma(a_j^i) = 1\}$. Let us observe that $w_C^- \subseteq w_C$. For proving that w_C^- satisfies the three conditions of Definition 16 for the pair $(|w_C|, p)$ we need to prove the following useful property:

(OddEvenProperty) for each $1 \le i \le n$ we have that $\{2i, 2i+1\} \cap \pi_Z(w_C \setminus w_C^-) \neq \emptyset$.

Informally speaking property *(OddEvenProperty)* states that for every possible value $2i \in \pi_Z(w_C \setminus w_C^-)$ it is not the case that both $2i$ and $2i+1$ do not belong to $\pi_Z(w_C \setminus w_C^-)$. Let us assume by contradiction that there exists an index i with $1 \le i \le n$ for which $2i \notin \pi_Z(w_C \setminus w_C^-)$ and $2i+1 \notin \pi_Z(w_C \setminus w_C^-)$. Thus, for each j with $1 \le j \le 3$ all the tuples of the form $(a_j^i, 1, 2i)$ and $(a_j^i, 0, 2i+1)$ belong to w_C^-. Let us take any index j with $1 \le j \le 3$. We have $(a_j^i, 1, 2i), (a_j^i, 0, 2i+1) \in w_C^-$. By definition of w_C^- from $(a_j^i, 1, 2i) \in w_C^-$ we have that $\sigma(a_j^i) = 0$, and from $(a_j^i, 0, 2i+1) \in w_C^-$ we have that $\sigma(a_j^i) = 1$ (contradiction).

Now we are ready to prove that conditions 1., 2., and 3. of Definition 16 are satisfied by the pair $|w_C|$ and p and thus $(w_C, |w_C|, p)$ is a positive instance of Problem 1. Condition 1. of Definition 16 imposes that $|w_C^-| \le |w_C|$ which is trivially satisfied since $w_C^- \subseteq w_C$. Condition 2. of Definition 16 imposes that for every pair of triplets $(a_j^i, \dot{b}, z), (a_{j'}^{i'}, \dot{b}', z') \in w_C \setminus w_C^-$ if $a_j^i = a_{j'}^{i'}$, i.e., they represent the occurrence of the same variable possibly in two distinct clauses we have $\dot{b} = \dot{b}'$. Let us assume by contradiction that this is not the case, then there exists $(a_j^i, 0, z), (a_{j'}^{i'}, 1, z') \in w_C \setminus w_C^-$ for some $z, z' \in \{2, \dots, 2n+1\}$ with $a_j^i = a_{j'}^{i'}$. By definition of w_C^- the fact that $(a_j^i, 0, z) \in w_C \setminus w_C^-$ means that $\sigma(a_j^i) = 0$ while $(a_{j'}^{i'}, 1, z') \in w_C \setminus w_C^-$ means that $\sigma(a_{j'}^{i'}) = 1$ since $a_j^i = a_{j'}^{i'}$ we have a contradiction.

Condition 3. of Definition 16 imposes that $|\pi_Z(w_C)| - |\pi_Z(w_C \setminus w_C^-)| \le p$. Let us assume by contradiction that there exist $p+1$ distinct indexes $2 \le i_1 < \dots < i_{p+1} \le 2n+1$ such that $i_j \notin \pi_Z(w_C \setminus w_C^-)$ for every $1 \le j \le p+1$. This means that for every $1 \le j \le p+1$ if i_j is even (resp., odd) then $(a_q^i, 1, i_j) \in w_C^-$ (resp., $(a_q^i, 0, i_j) \in w_C^-$) for each $1 \le q \le 3$ and thus by definition of w_C^- we have $\sigma(a_q^i) = 0$ for each $1 \le q \le 3$, thus the clause $i_j/2$ (resp., $(i_j - 1)/2$) is homogeneous w.r.t to σ.

Since, σ is a "witness" that (C, p) is a positive instance of Problem1 we have that is the number of clauses homogeneous w.r.t σ is at most p. Since we just proved that $2 \le i_1 < \dots < i_{p+1} \le 2n+1$ may be associated to $p+1$ homogeneous clauses then there exist $1 \le j' < p+1$ such that $i_{j'}$ is even and $i_{j'+1} = i_{j'} + 1$ because at least two distinct indexes among $i_1, \dots, i_{p+1}$ must be mapped to the same clause. However, by applying the $(OddEvenProperty)$ on $i_{j'}, i_{j'+1}$ we have that at least one among $i_{j'}$ and $i_{j'+1}$ must belong to $\pi_Z(w_C \setminus w_C^-)$ and thus we have a contradiction.

For the right-to-left direction, let us assume that w_C and $(|w_C|, p)$ is a positive instance of Problem 1. Thus, there exists $w_C^- \subseteq w_C$ and a function $f : \mathcal{A}' \to \{0,1\}$ with $\mathcal{A}' \subseteq \mathcal{A}$ such that:

- for all $(a, \dot{b}) \in \pi_{A\dot{B}}(w_C \setminus w_C^-)$ we have $\dot{b} = f(a)$;
- $|\pi_Z(w_C)| - |\pi_Z(w_C \setminus w_C^-)| \le p$.

Let us assume w.l.o.g. that w_C^- is minimal, that is for every $(a, \dot{b}) \in \pi_{A\dot{B}}(w_C^-)$ we have that there exists $(a, \dot{b}') \in \pi_{A\dot{B}}(w_C \setminus w_C^-)$ with $\dot{b} \neq \dot{b}'$. In other words, any tuple in $\pi_{A\dot{B}}(w_C^-)$ "conflicts" with at least one tuple in $\pi_{A\dot{B}}(w_C \setminus w_C^-)$. Under this assumption, we may easily prove that $\mathcal{A}' = \mathcal{A}$. Let us assume by contradiction that $\mathcal{A}' \subset \mathcal{A}$. Thus, there exists $a \in \mathcal{A} \setminus \mathcal{A}'$ such that $(a, 0), (a, 1) \in \pi_{A\dot{B}}(w_C^-)$. If we take $w_C^{=} = w_C^- \setminus \{(a, 0, z) : (a, 0, z) \in w_C^-\}$ we have that $w_C \setminus w_C^{=}$ admits a $(|w_C|, p')$ conflict resolution with $p' \leq p$ since, informally speaking, we are possibly "reducing" the size of w_C^-. By construction, we have that $\{(a, 0, z) : (a, 0, z) \in w_C^-\} \neq \emptyset$ because since $a \in \mathcal{A}$ we have that there exists at least one clause $\{a_1^i, a_2^i, a_3^i\}$ in C for which $a_j^i = a$ for some $j \in \{1, 2, 3\}$ and thus $(a, 0, 2i+1) \in w_C$. Thus, we can conclude that w_C^- is not minimal (contradiction). By having $\mathcal{A}' = \mathcal{A}$ we can now claim that f is also a completely defined assignment for C. Let us prove that f is an assignment that makes at most p clauses in C homogeneous. Let us assume by contradiction that f makes at least $p+1$ distinct clauses homogeneous and let $i_1 < \ldots < i_{p+1}$ be the indexes of such clauses. By construction and by minimality of w_C^-, let us assume that for every $1 \leq h \leq p+1$ either $(a_j^{i_h}, 0, 2i+1) \in w_C \setminus w_C^-$ for every $j \in \{1, 2, 3\}$ – in such a case $f(a_1^{i_h}) = f(a_2^{i_h}) = f(a_3^{i_h}) = 0$–, or $(a_1^{i_h}, 0, 2i) \in w_C \setminus w_C^-$ for every $j \in \{1, 2, 3\}$ – in such a case $f(a_1^{i_h}) = f(a_2^{i_h}) = f(a_3^{i_h}) = 1$. This means that for each $1 \leq h \leq p+1$, if $f(a_1^{i_h}) = f(a_2^{i_h}) = f(a_3^{i_h}) = 1$, we have $2i_h \in \pi_Z(w_C \setminus w_C^-)$ and $2i_h + 1 \notin \pi_Z(w_C \setminus w_C^-)$. Symmetrically, for each $1 \leq h \leq p+1$ if $f(a_1^{i_h}) = f(a_2^{i_h}) = f(a_3^{i_h}) = 0$ we have $2i_h \notin \pi_Z(w_C \setminus w_C^-)$ and $2i_h + 1 \in \pi_Z(w_C \setminus w_C^-)$. Let $U = \{2i_1, 2i_2 + 1, \ldots, 2i_{p+1}, 2i_{p+1} + 1\}$. We can conclude that $\pi_Z(w_C \setminus w_C^-) \cap U$ and $\pi_Z(w_C^-) \cap U$ is a bi-partition of U with $|\pi_Z(w_C \setminus w_C^-) \cap U| = |\pi_Z(w_C^-) \cap U| = p+1$. Since we have $(\pi_Z(w_C^-) \cap U) \cap \pi_Z(w_C \setminus w_C^-) = \emptyset$ and trivially $\pi_Z(w_C^-) \cap U \subseteq \pi_Z(w_C)$, we have that $(\pi_Z(w_C^-) \cap U) \subseteq (\pi_Z(w_C) \setminus \pi_Z(w_C \setminus w_C^-))$ and, thus, $|\pi_Z(w_C^-) \cap U| = p+1 \leq |\pi_Z(w_C)| - |\pi_Z(w_C \setminus w_C^-)|$. Thus $|\pi_Z(w_C)| - |\pi_Z(w_C \setminus w_C^-)| \geq p+1$ (contradiction). ◀

As we just proved, the problem of verifying any APFD even only considering H_3 is NP-Hard. Algorithm 2 represents a guess and check non-deterministic algorithm to solve the general problem, namely to verify all three errors. This algorithm shows that the verification of the three errors is an NP-complete problem. In the following algorithms, the symbol ▷ precedes comments.

Algorithm 2 ApproximateDependencyCheck.

Input: an instance w of relation W, and three real numbers ϵ_{g_3}, ϵ_{h_3}, and ϵ_{j_3} in $[0, 1]$
Output: a relation $s \subseteq w$ s.t. $s \models A \to \dot{B}$, $g_3(w, s) \geq 1 - \epsilon_{g_3}$, $h_3(w, s) \geq 1 - \epsilon_{h_3}$, $j_3(w, s) \geq 1 - \epsilon_{j_3}$

begin
  guess $s \subseteq w$
  ▷ `Check if` $s \models A \to \dot{B}$
  for $v \in \pi_A(s)$ **do**
    if $|\pi_{\dot{B}}(\sigma_{A=v}(s))| \geq 2$ **then**
      fail
  ▷ `Check` $g_3(w, s)$
  if $\frac{|s|}{|w|} < 1 - \epsilon_{g_3}$ **then**
    fail
  ▷ `Check` $h_3(w, s)$
  if $\frac{|\pi_Z(s)|}{|\pi_Z(w)|} < 1 - \epsilon_{h_3}$ **then**
    fail
  ▷ `Check` $j_3(w, s)$
  for $z \in \pi_Z(s)$: **do**
    if $\frac{|\sigma_{Z=z}(s)|}{|\sigma_{Z=z}(w)|} < 1 - \epsilon_{j_3}$ **then**
      fail
  return s