Information and Computation 301 (2024) 105228

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/yinco

Predictive mining of multi-temporal relations

Beatrice Amico *, Carlo Combi, Romeo Rizzi, Pietro Sala

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

ARTICLE INFO

Article history:
Received 2 February 2024
Received in revised form 26 July 2024
Accepted 29 September 2024
Available online 3 October 2024

Keywords:
Temporal databases
Temporal data mining
Functional dependencies
Explainable data mining

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 consider the concept of approximation for such APFDs, introduce new error measures, and discuss different strategies for deriving APFDs. We discuss the quality, i.e., the informative content, of the derived AFDs by considering their entropy and information gain. Moreover, we outline the results in deriving APFDs focusing on the Acute Kidney Injury (AKI). We use real clinical data contained in the MIMIC III dataset related to patients from Intensive Care Units to show the applicability of our approach to real-world data.

1. Introduction

Data mining has been receiving considerable attention from the research community. Indeed, such mining techniques provide a way to extract relevant knowledge and useful information hidden in the (often huge) amount of data available in many different contexts [1,2]. In particular, temporal data mining techniques are able to exploit the available knowledge to support decision-making. They offer the possibility of obtaining a considerable understanding of various domain-specific phenomena and the potential for the development of accurate classification models [3,4].

In a context where current systems enable us to store huge quantities of data, another important role of data mining is to support predictions. *Prediction* is often associated with well-known machine learning techniques. These algorithms are used in many domains, and different performance metrics are adopted for each different problem, e.g., in the information retrieval domain, precision and recall are widely accepted [5,6], while in medicine, the stakeholders prefer the ROC curve [7]. Often, it is not possible to understand why machine learning algorithms are proposing specific predictions, and such an already known black-box problem [8] interferes with the communication between data scientists and domain experts, as the need for *explainability* is not fully supported. Thus, even though black-box Machine-Learning approaches have been proposed to discover hidden temporal patterns in data collections [9–13], often their application in real-world contexts is limited by the absence of explainability, which prevents the users from trusting in the obtained results [14]. Explainable AI (XAI) is a research topic that recently received high attention, as it is inherently multifaceted. Indeed, it involves, among many others, aspects related to (i) the interpretation of results with respect to the available data, (ii) the comprehension of

* Corresponding author.
E-mail addresses: beatrice.amico@univr.it (B. Amico), carlo.combi@univr.it (C. Combi), romeo.rizzi@univr.it (R. Rizzi), pietro.sala@univr.it (P. Sala).

https://doi.org/10.1016/j.ic.2024.105228

the procedures that allowed to obtain the given results, (iii) the users who possibly require different kinds of explanation, (iv) the usefulness of explanations with regard to some specific domain requirements [14–16].

A workaround to support explainability in temporal data mining techniques is either (i) to associate black-box techniques with other ones devoted to providing some kind of (possibly visual) explanation of the obtained results or (ii) to exploit explainable temporal data mining techniques, able to possibly reveal intrinsic data dependencies. Focusing on this last direction, discovering temporal patterns represents an explainable way of studying hidden data dependencies, supporting users to focus on the most interesting and relevant discovered temporal data associations.

Moving closer to data dependencies, in the last decade Functional Dependencies (FDs), a well-known concept in the database context, allowing the representation of dependencies between attribute sets in a database, received renewed attention [4,17–20] from different points of view and for different goals. *Functional Dependencies* (FDs) were originally proposed to specify data constraints in the relational setting and then to derive normalized relational schemata [21]. Let r be a relation over the relational schema $R(U)$ and let $X, Y \subseteq U$. r fulfills the functional dependency $X \rightarrow Y$ (written as $r \models X \rightarrow Y$) if $\forall t, t' \in r(t[X] = t'[X] \rightarrow t[Y] = t'[Y])$.

From one side, FDs are effective in specifying data constraints, which must be verified and satisfied by the considered data repositories/lakes (now possibly consisting of many databases, which evolve as for the required constraints over time) [22,23]. Indeed, data quality is becoming an urgent topic in the current context, where huge amounts of data are processed on a daily basis, often without any verifiable data quality process [24,25]. On the other side, 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 [19,20,26–28]. Approximate temporal functional dependencies (ATFDs) have been proposed to mine different kinds of temporal patterns from data [29,4,25]. Recently, ATFDs have been recently proposed for the prediction task [30,31]. Such a 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, ATFDs 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 for the prediction of future events through the identification of recurring past temporal data patterns, expressed as *Approximate Predictive Functional Dependencies* (APFDs), according to a 3-window-based temporal framework. New kinds of error and related thresholds are introduced, together with suitable algorithms, to deal with the required approximation. The proposed APFDs are not related to any specific application domain as they introduce some advanced features in the line of functional dependencies. Without loss of generality, in this paper, we will focus on some examples taken from the domain of clinical medicine to show the applicability of APFDs in real-world scenarios. Indeed, through temporal data mining techniques, clinical data sources would enable us to rapidly generate prediction models for many clinical problems, support clinical decision-making, speed up medical processes, prevent and stratify risks, predict mortality, and improve patient quality of life (see, for example, [32–34,30,35–37] for different research efforts in this line. Here, we discuss and formalize the framework according to the following specific aspects:

- We introduce and exemplify the entire framework for the APFDs in a formal way by characterizing *multi-temporal relations*. As an element of novelty, multi-temporal relations are attribute timestamped relations [38], but still in first order normal form, where disjoint groups of attributes are associated to different time instants, explicitly ordered to represent some kind of history through attribute values. The framework allows the representation of dependencies between temporal attribute(s) patterns and some successive predicted attribute(s) patterns. The framework is based on three temporal windows, named Observation Window (OW), Waiting Window (WW), and Prediction Window (PW), respectively. WW is explicitly introduced to create a time span before the prediction for being able to (possibly) manage the predicted event.
- With respect to the preliminary proposal in [31], here we extend APFDs to have both observation and prediction *temporal attribute patterns*, a new concept suitably introduced in this paper, while in [31], we focused on a single attribute set holding at a specific valid time. Moreover, the definition of APFDs is made more sound with respect to their 'predictive' behavior by explicitly requiring that approximation cannot miss any value of the predicted attributes pattern with respect to the values originally contained in the considered multi-temporal relation.
- We introduce new error measures for the evaluation/derivation of AFPDs. Besides errors G_3, H_3, and J_3, already introduced in [31], we propose and discuss new local and global errors, namely the predictive error measure pG_3 and the error measure K_3, which allow considering the number of tuples and the number of values for temporal attribute patterns we have to disregard in the approximation.
- We propose and discuss a new approach to evaluate the 'goodness' of the derived APFDs, by using entropy-based concepts. In particular, we define and exemplify the Information Gain of an APFD and use the Kullback-Leibler divergence of the derived APFD with respect to the original multi-temporal relation and the derived approximation.
- We provide some experimental results on real clinical data from patients in Intensive Care Units, using data from MIMIC III [39], to obtain different APFDs. MIMIC III is an anonymized dataset related to patients admitted to intensive care units (ICUs). Without loss of generality, such results are a proof-of-concept to show the applicability of our approach to real-world data. We discuss there also some aspects related to the explainability by showing the most common attribute values in the derived dependencies.

Our paper unfolds as follows. Section 2 contains the related work. In Section 3, we provide and discuss a motivating example taken from a medical domain that we will use throughout the paper. Section 4 details the 3-window-based framework for prediction, the formalization of predictive functional dependencies, and the related concepts, such as temporal attribute pattern (TAP), multi-temporal relation, and time-frame view. Section 5 deals with the proposal and discussion of different error measures. Section 6 formally defines APFDs and their minimal subset; different algorithmic approaches and issues towards deriving APFDs are then proposed and discussed. In Section 7, we introduce an entropy-based approach to evaluate the quality of the derived APFDs. Section 8 contains some experimental results and the related discussion. Finally, Section 9 sketches some final comments.

2. Related work

In the context of temporal data mining, various techniques have been applied to time-oriented data to discover knowledge about temporal relationships among different raw data and/or more abstract concepts.

Association rule mining is one of the most common data mining (DM) techniques; it aims to extract exciting correlations based on some measure of interestingness (e.g., confidence/precision, support, or lift), frequent patterns, associations, or casual structures among sets of items in a database. Typically, such relations are expressed as if-then rules consisting of different rule antecedents (conditions) and consequents (targets). In literature, there are different methods to mine *temporal association rules* (TARs [40–43]). Sacchi et al. [41] present an approach to pre-process and interpret clinical time series. They aim to filter the original time series using temporal abstractions and then analyze the new and derived time series using statistical and artificial intelligence methods. After developing a TAR mining framework mainly oriented to the analysis of clinical data, the authors extend the framework. They propose a new kind of TAR to extract the frequent temporal precedence occurrences between patterns.

In the context of temporal abstractions and pattern discovery, temporal pattern mining is another way to discover new knowledge from a huge amount of data. In [44], starting from the concept of Trend-Event Pattern [45] and moving through the concept of *prediction*, the authors propose a new kind of predictive temporal patterns, namely Predictive Trend-Event Patterns (PTE-Ps). The framework aims to combine complex temporal features to extract a compact and non-redundant predictive set of patterns composed by such temporal features.

Mining time interval data is another interesting research field, especially for the extraction of Time Intervals Related Patterns (TIRPs). In [46], the authors introduce TIRPClo, an efficient algorithm for the discovery of frequent closed TIRPs, a compact subset of all the frequent TIRPs based on which their complete information can be revealed. In addition, it is possible to use patterns as features for classification. For example, in [32], the authors propose a framework for discovering TIRPs only from the cohort of patients having the outcome event. The results showed that representing the TIRPs using the horizontal support outperformed the binary and mean duration representations. While all these approaches consider some kind of classification/prediction task, no one of them explicitly considers different time windows for temporal data. Moreover, the main difference between these approaches, with respect to the approach we will propose here, is related to the generality of the discovered knowledge. While the previously introduced proposals focus on extracting specific *value-based patterns or rules*, our proposal allows the user to discover stronger *attribute-based dependencies*, namely temporal FDs, representing multiple value-based patterns or rules, and required to hold for all the attribute values found in the mined data.

In recent years, FDs have been extended in many different directions and with different goals. Our discussion primarily includes three key research directions: the first direction involves the representation of constraints on temporal data using temporal functional dependencies (TFDs); the second one centers on the exploration and discovery of approximate functional dependencies (AFDs); and the last direction is concerned with leveraging functional dependencies (FDs) to facilitate prediction and classification tasks.

TFDs add a temporal dimension to classical FDs to deal with temporal data. In literature, we find various examples [47–52], where the authors propose different representation formalisms related to the temporal component of FDs. In [52], Combi et al. describe a new formalism based on multiple time granularities. They identify four relevant classes: pure temporally grouping, pure temporally evolving, temporally mixed, and temporally hybrid TFDs. With respect to APFDs, which we will propose in this paper, TFDs have been introduced to specify temporal constraints in a temporal database and do not consider any kind of predictivity when considering the temporal dimension of data.

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. Kivinen and Mannila [27] introduce three measures, known as G_1, G_2, and G_3. G_1 represents the number of violating couples of tuples, G_2 the number of tuples that violate the functional dependency, and finally, G_3 the minimum number of tuples in r to be deleted for the FD to hold. The discovery of AFDs is a computationally expensive task. In the literature, there are different algorithms proposed to perform the discovery in an efficient way. Kruse and Naumann in [20] give an example of this type of algorithm. More recently, AFDs have been included in the more comprehensive scenario, where we refer to them as *relaxed FDs* (RFDs). Where the concept of approximation is wider and considers not only exceptions, i.e., violating tuples, but also similarities among attribute values and conditional constraints [26,17]. AFDs do not consider any temporal dimension of data. As for the error measures, APFDs use error G_3 introduced in [27]. Further error measures are proposed for APFDs; they extend the tuning capability for discovering meaningful predictive patterns in a temporal database.

In [22], the authors face another aspect related to the approximation of FDs. They assume that frequent constraint violations in a database may be related to the fact that the considered (mini) world is changeable while the specified constraints remain static. Their method is based on understanding which FDs are violated and repairing them by adding attributes to the antecedent of the dependency. FDs violated by current data are identified, and some approaches are proposed to suitably modify the given FD according to the new reality represented through the existing data. The authors calculate a confidence measure for each violated FD, creating a ranked list of candidate attributes. They use an iterative process where, at each step of the iteration, the next attribute is chosen in order to be added to the antecedent, and we do so by adding the attribute that produces the candidate FD with the highest rank. Such kind of approximation is again related to considering plain, nontemporal, FDs as constraints, which have to be refined according to the evolving represented reality.

Another example of approximate functional dependencies is the pattern functional dependencies (PFDs) [53]. Relaxing the FD's constraints of operating on entire attribute values, the authors introduce a new type of dependency that can capture partial attribute values that follow some regex-like patterns. In this context, a pattern is a sequence of characters defined over the generalization tree, a tree defined over an alphabet, where each leaf node is a character in Σ, and each intermediate node is a generalization of its child nodes. Formally, a PFD ψ defined over schema R is a pair $R(X \rightarrow Y; T_p)$, where: (i) X and Y are sets of attributes from R; (ii) $X \rightarrow Y$ is a standard FD, called an embedded FD; (iii) T_p is a tableau with all attributes in X and Y, where for attribute A in X or Y and each tuple $t_p \in T_p, t_p[A]$ is either a constrained pattern that matches values in $dom(A)$, or an unnamed variable $\perp$ that is used as a wild card. While PFDs are employed for discovering data errors and/or cleaning data, they differ from the APFDs proposed here, as (i) they do not consider any kind of temporality, and (ii) their focus is on checking partial/approximate values of string attributes, which has not been considered for APFDs.

In literature, there is a specific functional dependency that combines the characteristics of AFDs and TFDs. In [18], the authors introduce the concept of Approximate Temporal Functional Dependencies (ATFDs), which are defined and measured on either temporal granules or sliding windows. These dependencies are then applied to extract insights from data in the fields of psychiatry and pharmacovigilance. Additionally, they propose a novel error measure, denoted as G_4. This metric assesses the minimum number of tuples in the dataset r that need to be modified for the plain Temporal Functional Dependency (TFD) to hold across all tuples in r. While both ATFDs are APFDs are considering temporal data, ATFDs consider functional dependencies by grouping tuples according to their valid times, while APFDs require a precedence constraint between valid times of attributes in the antecedent and those in the consequent. Moreover, ATFDs and APFDs are derived by considering different error measures. Indeed, for ATFDs, authors introduced a kind of error measure, namely G_4, explicitly dealing with the possible presence of the same tuples in different temporal windows. On the other side, the error measures we propose here focus on finely profiling those tuples that do not satisfy the considered APFD.

Another example of ATFDs is [54], where the authors present AETAS, a system for discovering approximate temporal functional dependencies. The discovered TFDs are mainly pure temporally grouping TFDs with moving windows, according to the classification proposed in [52]. They mine the duration that leads to identifying temporal outliers, tackling the problem of the sparseness of the data with value imputation, and reducing the noise by enforcing the rule in the smallest meaningful time bucket. They also consider rules with constants (similar to conditional functional dependencies) such that specific duration can be used for particular entities, 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 and the discovery of the "optimal" duration for the moving window. Even in this case, while the discussed error measures are related either to the support of a dependency or to the repair of tuples, the focus is not on reasoning on attribute values holding at different times in the same tuple. On the other side, APFDs do not deal with the discovery of optimal durations for moving windows, as such durations are set before data mining starts.

Widening the panorama, in literature there are different examples of the use of *FDs to support prediction and classification tasks*. In the study [55], the authors demonstrate that the existence of functional dependencies among features tends to have a detrimental impact on the performance of classifiers. Likewise, in the work presented in [56], functional dependencies are leveraged to construct a dependency graph among classification attributes to diminish the overall number of attributes in the dataset. In [57], the authors address the notion of trusting ML models by using functional dependencies. They explore the interplay between supervised classification and functional dependencies, introducing a novel approach to assess the viability of classification on a given dataset exploiting functional dependencies. As far as we know, it is the first example of such a use of functional dependencies. In their method, given a set of features denoted as $(A_1, \ldots, A_n, C)$, where the values of C represent the class to be predicted, they seek functional dependencies in the form of $A_1, \ldots, A_n \rightarrow C$. With respect to APFDs, here the authors use standard error measures defined in [27] and do not consider predictivity within a proper temporal scenario, mainly restricting their focus on classification.

Finally, some recent research efforts have been devoted to the discovery of (precise) FDs in large datasets [58–60], focusing on algorithms allowing acceptable performances when dealing with huge datasets. Here the focus is on discovering exact FDs, and the approximation is related to the accuracy of the found FDs. Other contributions focus on the discovery of approximate functional dependencies [61–63]. These approaches differ from the proposed APFDs, as they do not consider any temporal/predictive aspect. However, they face some common topics as the proposal of new error measures [62] and the use of information theory to detect the relevance of the discovered approximate FDs [61].

3. A motivating scenario from clinical medicine

Nowadays, technology allows us to collect vast amounts of medical information automatically. A key consideration in this context is the temporal component, which is essential for accurately representing information within computer-based systems. This temporal dimension is crucial for tasks such as querying information, temporal reasoning, designing analytical tools for prediction, personalized medicine, and providing support for therapy. To illustrate the significance and potential implications of our approach, we turn our attention to a real-world example within the domain of Intensive Care Units (ICUs), specifically focusing on patients suffering from Acute Kidney Injury (AKI) [64]. This syndrome serves as a reference point throughout the paper, showcasing the applicability and relevance of our methodology.

Intensive care units provide critical care and life support for most severely ill and injured patients in the hospital. Continuous monitoring and frequent laboratory tests are integral components of patient care in these units, aimed at promptly identifying any deterioration in conditions or the onset of adverse events that could further impact the already fragile health state. Clinicians record a plethora of parameters, including but not limited to administered medications, levels of various indicators such as blood urea nitrogen, calcium, chloride, creatinine, hemoglobin, platelet count, potassium, prothrombin time, partial thromboplastin time, and white blood cell count. Additionally, diverse physiological measures are closely observed, encompassing arterial blood pressure, heart rate, systolic and diastolic blood pressure, respiratory rate, temperature, oxygen saturation, and glucose levels.

In ICU, AKI emerges as a prevalent clinical challenge. It is characterized by the loss of the kidney's ability to excrete wastes, concentrate urine, regulate electrolytes, and manage fluid balance, as highlighted in the work by [65].

In 2012, Kidney Disease: Improving Global Outcomes (KDIGO) released specific guidelines, as outlined in [66], for the definition of AKI. According to these guidelines, a patient is diagnosed with AKI if any of the following criteria are met: (i) an increase in serum creatinine by ≥ 0.3 mg/dl (≥ 26.5 µmol/l) within 48 hours, (ii) an increase in serum creatinine to ≥ 1.5 times the baseline within the previous 7 days, or (iii) a urine volume ≤ 0.5 ml/kg/h for 6 hours.

Let us assume that the considered clinical database collecting all the acquired data, named after `IcuDB`, is a *temporal database*, i.e., a database composed of temporal relations. Any temporal relation is characterized by a special attribute, named VT for Valid Time, representing the timepoint when the information represented in a tuple, is true in the modeled world [67].

Given our interest in discerning whether certain clinical data features enable the early identification of AKI patients, let us assume that we derive through a suitable query the (possibly materialized) view `PatientHistory`. This dataset represents various temporal states of patients over different valid times. Our goal is to associate a final state in this dataset with a specific indication of whether the patient has developed AKI.

For each patient, `PatientHistory` stores the patient's name and division, the heart rate and the blood pressure, the oxygen saturation, the administered drug –associated with the three considered states, respectively–, the diagnosis of AKI, and the different valid times, by means of attributes *Patient, Division, HR and BP, SpO_2, Drug, AKI*, and the associated valid time attributes, respectively. We assume that valid times are always given in terms of hours (starting from the admission time of each patient, taken as the origin of the time domain).

View `PatientHistory` is a special kind of *attribute timestamped* relation [68]. Indeed, different valid times are associated with disjoint sets of attributes, and values of different valid times in any tuple are strictly ordered. In the example, the valid time of vital signs precedes the valid time associated with SpO_2, which precedes that of Drug, which precedes that of AKI diagnosis.

Table 1 (partially) shows a possible instance of `PatientHistory` describing a clinical history of three patients, Daisy, Luke, and Stevie, possibly staying in different ICUs, who undergo five different drugs, some of them specific for the AKI treatment, respectively. Such history can be derived from the data contained in the temporal database `IcuDB`. Many different research questions arise from the introduced context, which are of general interest:

- Firstly, clinicians could be interested in discovering properties, relevant from the clinical point of view. *Within this perspective, could we support the prediction of a future diagnosis by building suitable clinical histories? More generally, may we be able to support predictive tasks through the data analysis of such temporal histories?* From this point of view, explainability is highly relevant for physicians. They are allowed to specify large or focused clinical histories, and then understand which parts, i.e., attributes, of these histories are relevant from a clinical point of view. For example, physicians recently focused on the presence of sepsis, the administration of diuretics and/or nephrotoxic drugs, systolic blood pressure, and central venous pressure to specify relevant clinical histories to be mined [31]. Moreover, moving back to view `PatientHistory`, a mined dependency may be expressed through a sentence like "The same patterns of values for heart rate and the following oxygen saturation correspond to the same AKI diagnosis, in most rows of the clinical history," which is inherently explainable and interpretable.
- *May we consider different (temporal) requirements when composing such data histories?* Indeed, it could be of interest to specify some constraints restricting, for example, the time distances between the different states.
- *May we also consider different requirements for predictions?* Indeed, predictions are useful only if they come early enough to allow suitable prevention of the (possibly) predicted negative outcome. In the considered context, predicting AKI just a moment before its occurrence could not be useful to avoid negative effects on the patient's health.

Table 1
View `PatientHistory` storing data of a temporal query on database `IcuDB`.

#	Patient	Division	HR	BP	VT_{VS}	SpO_2	VT_{SPO_2}	Drug	VT_{Drug}	AKI	VT_{AKI}
1	Daisy	ICU	High	Hypo	19	Low	21	Aspirin	23	False	28
2	Daisy	CardioICU	Low	Hypo	2	High	4	Aspirin	6	False	18
3	Daisy	CardioICU	Low	Hypo	2	Medium	5	Aspirin	6	False	12
4	Daisy	CardioICU	Medium	Normo	5	Medium	7	Indapamide	9	False	18
5	Luke	ICU	Low	Hyper	7	High	8	Ibuprofen	12	True	17
6	Luke	ICU	Low	Hyper	7	High	8	Ibuprofen	12	True	21
7	Luke	ICU	Medium	Hyper	9	High	13	Sulindac	14	True	19
8	Luke	ICU	Medium	Hyper	9	High	13	Sulindac	14	True	21
9	Stevie	STICU	High	Hyper	1	Low	2	Aspirin	5	True	25
10	Stevie	STICU	High	Hyper	1	Low	2	Aspirin	5	False	12
11	Stevie	STICU	High	Hyper	1	Low	2	Aspirin	5	False	10
12	Stevie	STICU	High	Hyper	1	Low	2	Indapamide	7	False	10
...	Stevie	STICU	...	...	...	...	...	...	...	...	...
36	Stevie	STICU	Medium	Hyper	4	Medium	7	Metolazone	8	True	14
...	...	...	...	...	...	...	...	...	...	...	...

- *Which kinds of error thresholds are we allowed to specify for deriving reliable predictions?* Indeed, it could be that, after discovering that some (few) tuples do not support the prediction, we would like to specify other thresholds to make the prediction reliable. As an example, we may allow (or not) some patients to have their tuples completely discarded, as the overall condition of these patients makes them "outliers" with respect to the considered prediction.
- After discovering whether some features (i.e., attributes) support the prediction of future events, *are we able to discover which data values are related to specific clinical outcomes?* Even though explainable methods would be applied in supporting the early identification of AKI patients, having the capability of relating some specific value patterns to the AKI presence/absence is of great importance for physicians [14]. Thus, after having discovered that "The same patterns of values for heart rate and the following oxygen saturation correspond to the same AKI diagnosis, in most rows of the clinical history,", it is possible to observe through simple queries on the view that, for example, most of the time low values of heart rate followed by high oxygen saturation is associated to a consequent presence of acute kidney injury. Such a further level of explainability, strictly connected to the specific content of the database, allows the physician to be confident in the overall reliability of the mined dependencies.

In the following, we will use view `PatientHistory`, depicted in Table 1, to exemplify the different aspects of our proposal. Different examples will also consider some fragments of the overall view. The tuple enumeration used here and in the following examples comes from running suitable queries in a PostgreSQL corresponding database, and is used just for referencing specific tuples and for giving the idea of the overall cardinality of the query result/relation.

4. The predictive aspects of functional dependencies

In this section, we first outline the current problem and introduce a 3-window model for the interpretation of predictive temporal data. Subsequently, we provide the necessary definitions for establishing a Predictive Functional Dependency. Finally, we delve into the analysis of the concept of approximation concerning Predictive Functional Dependencies.

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

In the majority of cases, prediction models rely on two-time windows: (i) a data collection (or observation) window and (ii) a prediction window. Despite the existence of approaches [69,70] that incorporate a third temporal window, as far as we are aware, a comprehensive and formal prediction framework explicitly considering three distinct time windows has not been explored in the data mining literature. We propose a 3-window model where we can observe: (i) an Observation Window where we collect information that could be relevant to a future event over a specific time span (OW), (ii) a Waiting Window (WW), namely the minimum time interval required to act to prevent the event in the prediction window, assuming that not all the performed actions have an instantaneous effect; and (iii) a Prediction window (PW) where it is possible to observe the effects of something happened in the observation period. It is worth noting that the duration of these windows may vary and could even consist of a single time point. Additionally, the *Waiting Window* might be absent, meaning it has a length of zero, especially when decisions have an immediate observable effect.

There are different orthogonal features that can describe this 3-window model. We decide to analyze a first distinction between (i) *anchored* and (ii) *unanchored* time windows. With anchored time windows, we can represent specific periods of the considered time axis. Anchored time windows allow the representation of specific periods along the considered time axis. Considering the view in Fig. 1, that is a subset of the view `PatientHistory` and supposing to have an anchored 3-window model of 6 hours for the observation window, 3 hours for the waiting window, and 9 hours for the prediction window anchored to the beginning, we enclose in the result tuples 2,3,10,11.

#	Patient	BP	VT_{VS}	SpO_2	VT_{SPO_2}	Drug	VT_{Drug}	AKI	VT_{AKI}
2	Daisy	Hypo	2	High	4	Aspirin	6	False	18
3	Daisy	Hypo	2	Medium	5	Aspirin	6	False	12
4	Daisy	Normo	5	Medium	7	Indapamide	9	False	18
5	Luke	Hyper	7	High	8	Ibuprofen	12	True	17
6	Luke	Hyper	7	High	8	Ibuprofen	12	True	21
7	Luke	Hyper	9	High	13	Sulindac	14	True	19
8	Luke	Hyper	9	High	13	Sulindac	14	True	21
10	Stevie	Hyper	1	Low	2	Aspirin	5	False	12
11	Stevie	Hyper	1	Low	2	Aspirin	5	False	10
36	Stevie	Hyper	4	Medium	7	Metolazone	8	True	14

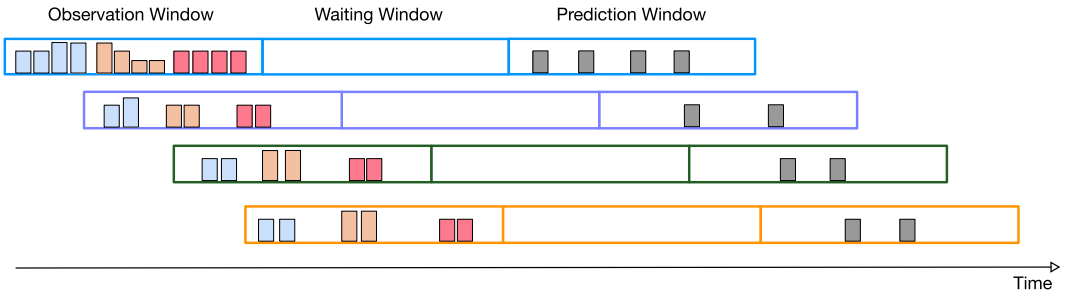

Fig. 1. The time windows of the proposed framework.

Unanchored time windows are characterized by their mobility along the time axis, constraining only the temporal span between the 3 temporal windows. In this case, we may consider a possibly infinite number of unanchored (sliding) windows, that describe the database's history. In our framework, sliding windows represent a way to highlight the entire time span, from the beginning to the end of the database's history. This means that we enclose all the tuples that satisfy the temporal restrictions given by the 3-window framework in our result. Therefore, the well-known concept of "stride" associated with sliding windows is inapplicable, as in the results we consider all tuples that meet the temporal constraints in at least one of the 3-window frameworks. In Fig. 1, we depict a partial overlap among four 3-window models, representing unanchored time windows. In this instance, we adopt the 3-window model comprising 6 hours for the observation window, 3 hours for the waiting window, and 9 hours for the prediction window. Each 3-window model is distinguished by a unique color, that marks the tuples within each temporal frame. For example for the second model, we consider tuples 4 and 36, while for the third one, we considered tuples 5 and 6. Each 3-window model is shifted forward for 3 hours. At every step, we include different tuples, and at the end, all the tuples are included in the result.

Formally, we define the Unanchored Time Frame and the Anchored Time Frame as follows:

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 with 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.

Another significant distinction, which can substantially impact prediction outcomes and is independent of the classification into anchored or unanchored time windows, is between (i) *fixed-length* and (ii) *variable-length* time windows. In this context, the three windows can be of fixed length, implying no constraints on the temporal position of the data within the window or of variable length, concluding with the last time point associated with the data to be considered in the window. In the *variable-length* case, the waiting window can start before the maximum span allowed for the observation window. For example, suppose to consider a 3-window framework of 6 hours for the OW, 2 hours for the WW, and 8 hours for the PW, involving three different temporal measures: heart rate, SPO_2, and drug. We might specify that valid times for heart rate, SPO_2, and drug must fall within 6 hours. However, if the three measures conclude before the sixth hour, we might consider starting the Waiting Window even before the 6-hour delay from the valid time of heart rate. Consequently, the Prediction Window may also need to be anticipated.

For the sake of simplicity, we assume here that all the different time windows meet in time. Actually, the waiting window is a way of representing a gap between observation and prediction windows. Definitions 1 and 2 may be straightforwardly

extended to consider gaps inside observation and prediction windows, to be able, for example, to represent periodic time spans.

4.2. 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.

Let

- $\mathcal{A}$ be a finite set of attribute names $\mathcal{A} = \{A_1, A_2, \ldots, A_z, \ldots\}$,
- $\mathcal{VT}$ a finite set of valid time attribute names $\mathcal{VT} = \{VT_1, \ldots, VT_i, \ldots, VT_m, VT_{m+1}, \ldots, \}$
- $\mathcal{D}$ a finite set of domains each of them containing atomic values for attributes.
- $\mathcal{T} \in \mathcal{D}$ be the (discrete) time domain.
- $Dom : \mathcal{A} \cup \mathcal{VT} \rightarrow \mathcal{D}$ is the mapping, which defines the domain of each attribute, where $Dom(VT_i) = \mathcal{T}$ for each $VT_i \in \mathcal{VT}$

Definition 3 *(Multi-temporal relation (mt-relation)).* A multi-temporal relation *mtr* is a relation with schema WT where $W \subseteq \mathcal{A}$ and $T = \{VT_1, \ldots, VT_i, \ldots, VT_k, VT_{k+1}, \ldots, VT_{k+n}\} \subseteq \mathcal{VT}$ are $k+n$ ordered valid time attributes. A relation *mtr* is a set of tuples composed of values of domains $Dom(A_i)$ for any $A_i \in W$, and of domain $\mathcal{T}$ for any $VT_i \in T$.

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

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

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

Valid time attribute set T is an ordered set, where $VT_1 < VT_2 < \ldots < VT_i < \ldots < VT_k < \ldots < VT_{k+n}$

For any relation *mtr* it holds $VT_i < VT_j \Leftrightarrow \forall t \in mtr(t[VT_i] < t[VT_j])$ for $1 \leq i < j \leq k+n$

As part of the novelty of our approach, a multi-temporal relation is a special kind of attribute timestamped relation, according to the terminology of temporal database community [68]. It may be considered as a suitable data structure for history-oriented data analysis, obtained from point-based temporal relations. Indeed, by explicitly specifying the association of attributes with (ordered) valid times, we can represent sequences of attribute values holding at different time points. With respect to other proposals dealing with attribute timestamped relations, in our proposal a multi-temporal relation is still in first normal form, as their attributes have all atomic values, and does not satisfy the tuple homogeneity, i.e., valid times of different attributes correspond to different time points [38].

Example 1. View `PatientHistory` depicted in Table 1 represents an mt-relation with 4 valid times, where $VT_{VS} < VT_{SpO_2} < VT_{Drug} < VT_{AKI}$ and $Vtime(VT_{VS}) = \{HR, BP\}$, $Vtime(VT_{SpO_2}) = \{SpO_2\}$, $Vtime(VT_{Drug}) = \{Drug\}$, and $Vtime(VT_{AKI}) = \{AKI\}$.

Definition 4 *(Temporal Attribute Patterns (TAPs)).* Given a multi-temporal relation schema WT, a Temporal Attribute Pattern (TAP) $\aleph \subseteq W$ is any subset of W. Given two TAPs $\aleph_1$, $\aleph_2$ we say that $\aleph_1 < \aleph_2$ iff

$\forall A \in \aleph_1 \ \forall B \in \aleph_2 \ (A \in Vtime(VT_A) \wedge B \in Vtime(VT_B)) \implies VT_A < VT_B$

As we will discuss in the following, the main idea here is to propose a general framework allowing the definition of "specialized" functional dependencies. These dependencies have an antecedent consisting of a set of attributes known as *observed attributes*, ordered according to the corresponding valid times, and a consequent defined as the *predicted attributes*. To differentiate these attribute roles, we introduce an appropriate partition of attributes, as per the subsequent definition.

Definition 5 *(Prediction-oriented partition of mt-relation valid times).* Given a multi-temporal relation *mtr* with schema WT, where $W \subseteq \mathcal{A}$ and $T \subseteq \mathcal{VT}$, 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

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

In the following we assume that $O \equiv \{VT_1, VT_2, \ldots, VT_k\}$, while $\mathcal{P} \equiv \{VT_{k+1}, \ldots, VT_{k+n}\}$. To explicitly distinguish, observation-related and prediction-related valid times and associated attributes, we use overlined names for observation-related valid times, attributes and attribute sets, and underlined names for prediction-related valid times, attributes, and attribute sets, respectively. Thus, the generic multi-temporal schema composed by attributes WT will have $W = Z\overline{A}^1\overline{B}^1 \ldots \overline{C}^i\overline{A}^i \ldots \overline{F}^k\underline{D}_1 \ldots \underline{G}_i \ldots \underline{E}_n$ and the corresponding valid times $T = \{\overline{VT}^1, \ldots, \overline{VT}^i, \ldots \overline{VT}^k, \underline{VT}_1, \ldots, \underline{VT}_n\}$. According to this notation, $\overline{U}^i \equiv Vtime(\overline{VT}^i)$ for any $\overline{VT}^i \in O$ and $\underline{U}_i \equiv Vtime(\underline{VT}_i)$ for any $\underline{VT}_i \in \mathcal{P}$.

Example 2. The relation view depicted in Table 2 considers attributes according to the introduced notation. More precisely, in this case $O \equiv \{\overline{VT}^1, \overline{VT}^2, \overline{VT}^3\}$, $\mathcal{P} \equiv \{\underline{VT}_1\}$, and $\overline{U}^1 = \{\overline{HR}^1, \overline{BP}^1\}$, $\overline{U}^2 = \{\overline{SpO_2}^2\}$, $\overline{U}^3 = \{\overline{Drug}^3\}$, and $\underline{U}_1 = \{\underline{AKI}_1\}$.

Given a multi-temporal relation *mtr*, our current focus is on determining which tuples are considered "fine" or "contained" within a specified time frame. Specifically, we aim to identify tuples where the first k valid times are within the observation window OW, and the last n valid times fall within the prediction window PW. We will refer to these tuples as being "consistent" with the given 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 other different requirements for the observation window. Indeed, as for the first aspect, we may be interested in verifying the partial/complete containment of the first k-valid times within the given OW. As for the second aspect, we may consider either fixed- or flexible-length OWs, which end at 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 $T \subseteq \mathcal{VT}$, a (either anchored or unanchored) time frame α we say that t is time-frame consistent with α according to modality $m \in \{$'*flex*', '*fixed*'$\}$ in the ranges $[o_1, o_2]$ and $[p_1, p_2]$, where $1 \leq o_1 < o_2 \leq k$ and $1 \leq p_1 < p_2 \leq n$, respectively, if formula $\Theta(t, \alpha, m, [o_1, o_2], [p_1, p_2])$ holds.

Formula $\Theta(t, \alpha, m, [o_1, o_2], [p_1, p_2])$ is defined according to the following cases:

- $\Theta(t, \alpha,$ '*fixed*', $[o_1, o_2], [p_1, p_2]) \equiv t[\overline{VT}^{o_2}] - t[\overline{VT}^{o_1}] \leq OW \wedge t[\underline{VT}_{p_1}] - t[\overline{VT}^{o_1}] > OW + WW \wedge t[\underline{VT}_{p_2}] - t[\overline{VT}^{o_1}] < OW + WW + PW$
  –if the time frame is unanchored–, or
- $\Theta(t, \alpha,$ '*fixed*', $[o_1, o_2], [p_1, p_2]) \equiv t[\overline{VT}^{o_1}] \geq atp \wedge t[\overline{VT}^{o_2}] - t[\overline{VT}^{o_1}] \leq atp + OW \wedge t[\underline{VT}_{p_1}] - t[\overline{VT}^{o_1}] > atp + OW + WW \wedge t[\underline{VT}_{p_2}] - t[\overline{VT}^{o_1}] < atp + OW + WW + PW$
  –if the time frame is anchored–, or
- $\Theta(t, \alpha,$ '*flex*', $[o_1, o_2], [p_1, p_2]) \equiv t[\overline{VT}^{o_2}] - t[\overline{VT}^{o_1}] \leq OW \wedge t[\underline{VT}_{p_1}] - t[\overline{VT}^{o_2}] > WW \wedge t[\underline{VT}_{p_2}] - t[\overline{VT}^{o_2}] < WW + PW$
  –if the time frame is unanchored–, or
- $\Theta(t, \alpha,$ '*flex*', $[o_1, o_2], [p_1, p_2]) \equiv t[\overline{VT}^{o_2}] \geq OW_s \wedge t[\overline{VT}^{o_2}] - t[\overline{VT}^{o_1}] \leq atp + OW \wedge t[\underline{VT}_{p_1}] - t[\overline{VT}^{o_1}] > atp + OW + WW \wedge t[\underline{VT}_{p_2}] - t[\overline{VT}^{o_2}] < atp + OW + WW + PW$
  –if the time frame is anchored–

4.3. Defining predictive FDs

The overall idea is now to temporally characterize functional dependencies for the introduced multi-temporal relational model. We consider 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 $MTR(Z\overline{U}^1\overline{U}^2 \ldots, \overline{U}^k\underline{U}_1, \ldots, \underline{U}_n \cup \{\overline{VT}^1, \overline{VT}^2, \ldots, \overline{VT}^k, \underline{VT}_1, \ldots, \underline{VT}_n\})$, a time frame, which can be either a *uTF* or an *aTF*, and a modality $m \in \{$'*flex*', '*fixed*'$\}$, a Predictive Functional Dependency is expressed as:

$$\aleph \xrightarrow[\alpha,m]{} \beth$$

where

- $\aleph \equiv X\overline{P}^h\overline{Q}^i \ldots \overline{R}^j$ is an **observation TAP**, with $\overline{P}^h \subseteq \overline{U}^h$, $\overline{Q}^i \subseteq \overline{U}^i$, ..., and $1 \leq h < i < \ldots < j \leq k$
- $\beth \equiv \underline{P}_d\,\underline{Q}_f \ldots \underline{R}_g$ is a **prediction TAP**, with $\underline{P}_d \subseteq \underline{U}_d$, $\underline{Q}_f \subseteq \underline{U}_f$, ..., and $1 \leq d < f < \ldots < g \leq n$
- $\aleph < \beth$, according to Definition 4.

A PFD holds on a mt-relation *mtr* with schema *MTR* in a timeframe *TF* with modality *m*, *with an extended valid time range* (denoted as $mtr \models^E_{\alpha,m} \aleph \rightarrow \beth$) iff

Table 2
A *TF*-view, subset of view `PatientHistory`, depicted in Table 1 (with the attributes suitably renamed).

#	Patient	Division	$\overline{HR}^1$	$\overline{BP}^1$	$\overline{VT}^1$	$\overline{SpO_2}^2$	$\overline{VT}^2$	$\overline{Drug}^3$	$\overline{VT}^3$	$\underline{AKI}_1$	$\underline{VT}_1$
1	Daisy	ICU	High	Hypo	19	Low	21	Aspirin	23	False	28
2	Daisy	CardioICU	Low	Hypo	2	High	4	Aspirin	6	False	18
3	Daisy	CardioICU	Low	Hypo	2	Medium	5	Aspirin	6	False	12
4	Daisy	CardioICU	Medium	Normo	5	Medium	7	Indapamide	9	False	18
5	Luke	ICU	Low	Hyper	7	High	8	Ibuprofen	12	True	17
6	Luke	ICU	Low	Hyper	7	High	8	Ibuprofen	12	True	21
7	Luke	ICU	Medium	Hyper	9	High	13	Sulindac	14	True	19
8	Luke	ICU	Medium	Hyper	9	High	13	Sulindac	14	True	21
10	Stevie	STICU	High	Hyper	1	Low	2	Aspirin	5	False	12
11	Stevie	STICU	High	Hyper	1	Low	2	Aspirin	5	False	10
12	Stevie	STICU	High	Hyper	1	Low	2	Indapamide	7	False	10
36	Stevie	STICU	Medium	Hyper	4	Medium	7	Metolazone	8	True	14

$$\forall t, t' \in mtr((t[\aleph] = t'[\aleph] \wedge \Theta(t, \alpha, m, [1,k], [1,n]) \wedge \Theta(t', \alpha, m, [1,k], [1,n])) \rightarrow t[\urcorner] = t'[\urcorner])$$

A PFD holds on a mt-relation *mtr* with schema *MTR* in a timeframe *TF* with modality *m*, *with a restricted valid time range* (denoted as $mtr \models^R_{\alpha,m} \aleph \rightarrow \urcorner$) iff

$$\forall t, t' \in mtr((t[\aleph] = t'[\aleph] \wedge \Theta(t, \alpha, m, [h,j], [d,g]) \wedge \Theta(t', \alpha, m, [h,j], [d,g])) \rightarrow t[\urcorner] = t'[\urcorner])$$

Extended and restricted valid time ranges allow the specification of different criteria to be used when evaluating whether a tuple is consistent or not with a specific time frame in the process of checking a given PFD. Indeed, with the extended time range, all the attribute values of the tuple have to be contained in the observation and prediction windows, respectively. On the other hand, with the restricted time range, only values of those attributes appearing in the PFD are required to be contained in the proper time window.

According to the previous definition, it is straightforward to observe that the given PFD has to hold, by considering only a subset of *mtr*, compose by tuples consistent with the considered time frame, the modality, and the range. Such a subset is named after *time-frame relation view*.

Definition 8 *(Time-frame relation view with range and modality (TF-view)).* Given a multi-temporal relation *mtr* with schema WT, where $W \subseteq \mathcal{A}$ and $T \subseteq \mathcal{VT}$, a (either anchored or unanchored) time frame *TF*, a modality $m \in \{$'*flex*', '*fixed*'$\}$, and ranges $[o_1, o_2], [p1, p2]$, where $1 \leq o_1 < o_2 \leq k$ and $1 \leq p_1 < p_2 \leq n$, a *TF*-view $w \subseteq mtr$ is defined as $w = TFv(mtr, \alpha, m, [o_1, o_2], [p_1, p_2]) \equiv \{t \mid t \in mtr \wedge \Theta(t, \alpha, m, [o_1, o_2], [p_1, p_2])\}$

According to this definition, it is straightforward to observe that the specified ranges $[o_1, o_2]$, $[p_1, p_2]$ are closely connected with the valid time range we consider for PFDs.

Indeed, we may reformulate the previous definition by saying, for example, that a PFD holds on an mt-relation *mtr* with schema *MTR* in a timeframe *TF* with modality *m*, *with a restricted valid time range* (denoted as $mtr \models^R_{\alpha,m} \aleph \rightarrow \urcorner$) iff

$$\forall t, t' \in TFv(mtr, \alpha, m, [h,j], [d,g])(t[\aleph] = t'[\aleph] \rightarrow t[\urcorner] = t'[\urcorner])$$

Example 3. Let us consider the relation depicted in Table 2, which contains a TF-view derived from `PatientHistory` discussed in Section 3, with suitably renamed attributes. The considered time frame is defined with $\alpha = \langle 6, 2, 10 \rangle$, $m =$ '*fixed*'. According to this time frame, tuple 9 of `PatientHistory`, depicted in Table 1, does not belong to the TF-view. It is straightforward to observe that the PFDs $\overline{BP}^1, \overline{SpO_2}^2 \xrightarrow[\alpha,m]{} \underline{AKI}_1$ and $\overline{Drug}^3 \xrightarrow[\alpha,m]{} \underline{AKI}_1$ hold. On the other side, PFDs $\overline{BP}^1 \xrightarrow[\alpha,m]{} \underline{AKI}_1$ and $\overline{SpO_2}^2 \xrightarrow[\alpha,m]{} \underline{AKI}_1$ do not hold.

Hereinafter, we will consider a time frame $\alpha = \langle 6, 2, 10 \rangle$, $m =$ '*fixed*', an extended valid time range, a projection on the attributes of the considered TF-view, with the single attribute *Patient* not associated to a valid time, a single attribute associated to each valid time, and a single attribute as prediction TAP, for the examples we will discuss.

5. Error measures for predictive functional dependencies

During the extraction process of predictive functional dependencies from a generic *multi-temporal relation*, we have to address various aspects.

Firstly, we need to isolate the tuples that match a specified modality and valid time range, with the considered temporal framework consisting of observation, waiting, and prediction windows. These tuples assemble a *time-frame relation view*.

Secondly, a PFD could be valid only for a subset of tuples of the *time-frame relation view*. In this case, we have to examine the approximation concept. We need to assess whether such a subset is appropriate in the context of the prediction task supported by the given PFDs. Namely, we expect that a PFD f should be fulfilled by most tuples within the time-frame relation view w. A minor fraction of tuples within w is permitted to deviate from the specified dependency.

In this regard, we discuss some error measures, which analyze different aspects. We've started considering one of the measures proposed in [27] G_3, a measure related to the number of tuples that violate the PFD. Connected to this, we define $G_3(\underline{v})$ that identifies the number of tuples associated with a tuple value $\underline{v}$, deleted from the initial set of tuples in w.

Hereinafter, we will mainly focus, without losing generality, on the extended valid time range, i.e., considering ranges $[1,k],[1,n]$ for observation and prediction-related attributes. All the following definitions apply also when considering the restricted valid time range.

Formally, given a *TF*-view $w \subseteq mtr$, the initial error measure G_3 focuses on the count of tuples within w that must be deleted to achieve a relation s where the specified predictive functional dependency is satisfied [27]. In our context, it is expressed as:

Definition 9 *(Error measure G_3).* Given a *TF*-view $w = TFv(mtr, \alpha, m, [1,k], [1,n])$ of an mt-relation mtr with schema WT, and a PFD $\aleph \xrightarrow[\alpha,m]{} \beth$, and any relation $s \subseteq w$, such that $s \models^{E}_{\alpha,m} \aleph \rightarrow \beth$, the error measure G_3 is expressed as:

$$G_3 = |w| - |s|$$

The related *scaled measure* g_3 is defined as:

$$g_3 = \frac{G_3}{|w|}$$

In the prediction context, evaluating how many tuples we lost associated with a specific value $\underline{v}$ of $\beth$ could be interesting. This estimation allows us to understand how predictive the given APFD is for a specific value. And consequently, how significant our dependence is. For this reason, we introduce the new error measure $pG_3(\underline{v})$, formally defined as in the following.

Definition 10 *(Predictive Error measure $pG_3(\underline{v})$).* Given a *TF*-view $w = TFv(mtr, \alpha, m, [1,k], [1,n])$ of an mt-relation mtr with schema WT, a PFD $\aleph \xrightarrow[\alpha,m]{} \beth$ and any relation $s \subseteq w$, such that $s \models^{E}_{\alpha,m} \aleph \rightarrow \beth$, the error measure $pG_3(\underline{v})$ is expressed as:

$$pG_3(\underline{v}) = |\{t \mid \exists t \in w \wedge t[\beth] = \underline{v}\}| - |\{t \mid \exists t \in s \wedge t[\beth] = \underline{v}\}|$$

The related *scaled measure* is expressed as:

$$pg_3(\underline{v}) = \frac{G_3(\underline{v})}{|\{t \mid \exists t \in w \wedge t[\beth] = \underline{v}\}|}$$

A global predictive error measure may be defined by considering all the possible $pG_3(\underline{v})$ as in the following.

Definition 11 *(Global Predictive Error measure pG_3).* Given a *TF*-view $w = TFv(mtr, \alpha, m, [1,k], [1,n])$ of an mt-relation mtr with schema WT, a PFD $\aleph \xrightarrow[\alpha,m]{} \beth$ and any relation $s \subseteq w$, such that $s \models^{E}_{\alpha,m} \aleph \rightarrow \beth$, the error measure pG_3 is expressed as:

$$pG_3 = \max_{\underline{v} \in \{t[\beth] \mid \exists t \in w\}} \{pG_3(\underline{v})\}$$

while the related *scaled measure* is expressed as: $pg_3 = \max_{\underline{v} \in \{t[\beth] \mid \exists t \in w\}} \{pg_3(\underline{v})\}$

Example 4. Considering the *TF*-view in Table 3, the PFD $\overline{HR}^1, \overline{SpO_2}^2 \xrightarrow[\alpha,m]{} \underline{AKI_1}$, is not satisfied because of tuples 4 and 36, and of tuples 2, 5, and 6. A first option for having such PFD satisfied would be to delete tuples 2 and 4 (or 36). In this case, $g_3 = 2/9$. A second option would consist of deleting tuples 4 (or 36), 5, and 6. In this case, g_3 would be 3/9. Let us now consider $pG_3(\text{'}False\text{'})$ and $pG_3(\text{'}True\text{'})$, respectively: if we delete tuples 2 and 4, $pG_3(\text{'}False\text{'}) = 2/6$ and $pG_3(\text{'}True\text{'}) = 0/3$; if we delete tuples 2 and 36, $pG_3(\text{'}False\text{'}) = 1/6$ $pG_3(\text{'}True\text{'}) = 1/3$; if we delete tuples 5, 6 and 4, $pG_3(\text{'}False\text{'}) = 1/6$ and $pG_3(\text{'}True\text{'}) = 2/3$; if we delete tuples 5, 6, and 36, $pG_3(\text{'}False\text{'}) = 0/6$ and $pG_3(\text{'}True\text{'}) = 3/3$.

We then introduce three other error measures. We identify a first issue focused on the number of entities we accept to discard for the sake of a predictive functional dependency. The error measure H_3 permits the disregard of entities' data with a very low number of tuples [31]. This error measure helps to mitigate potential noise introduced into our dataset by such entities. We formally define this measure as follows:

Table 3
A simplified *TF*-view, with only a single atemporal attribute and one attribute for each valid time.

#	Patient	$\overline{HR}^1$	$\overline{VT}^1$	$\overline{SpO_2}^2$	$\overline{VT}^2$	$\overline{Drug}^3$	$\overline{VT}^3$	$\underline{AKI}_1$	$\underline{VT}_1$
1	Daisy	High	19	Low	21	Aspirin	23	False	28
2	Daisy	Low	2	High	4	Aspirin	6	False	18
3	Daisy	Low	2	Medium	5	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
10	Stevie	High	1	Low	2	Aspirin	5	False	12
11	Stevie	High	1	Low	2	Aspirin	5	False	10
36	Stevie	Medium	4	Medium	7	Metolazone	8	True	14

Definition 12 *(Error measure H_3).* Given a *TF*-view $w = TFv(mtr, \alpha, m, [1,k], [1,n])$ of an mt-relation mtr with schema WT, a PFD $\aleph \xrightarrow[\alpha,m]{} \beth$, and any relation $s \subseteq w$, such that $s \models^E_{\alpha,m} \aleph \xrightarrow[\alpha,m]{} \beth$, the error measure H_3 is expressed as[1]:

$$H_3 = |\{t[Z] \mid \exists t \in w\}| - |\{t[Z] \mid \exists t \in s\}|$$

The related *scaled measure* h_3 is defined as:

$$h_3 = \frac{H_3}{|\{t[Z] \mid \exists t \in w\}|}$$

Example 5. Considering the *TF*-view in Table 3, the PFD $\overline{HR}^1, \overline{SpO_2}^2 \xrightarrow[\alpha,m]{} \underline{AKI}_1$, is not satisfied because of tuples 4 and 36, and of tuples 2, 5, and 6. A first option for having such PFD satisfied would be to delete tuples 2 and 4 (or 36). In this case, all the entities, i.e., patients, of the mt-relation would still be represented, and, thus, $h_3 = 0$. A second option could be deleting tuples 4 (or 36), 5, and 6. In this case, Luke's tuples would disappear completely, and thus h_3 would be 1/3.

We describe a second issue focused on the number of tuples for each entity we accept to discard to satisfy the predictive functional dependency. We formalize an error called J_3. It ensures the maintenance of enough "consistent" information for each entity. We formally define this error measure as follows:

Definition 13 *(Error measure J_3).* Given a *TF*-view $w = TFv(mtr, \alpha, m, [1,k], [1,n])$ of an mt-relation mtr with schema WT, a PFD $\aleph \xrightarrow[\alpha,m]{} \beth$, and any relation $s \subseteq w$, such that $s \models^E_{\alpha,m} \aleph \xrightarrow[\alpha,m]{} \beth$, the error measure J_3 is expressed as in the following.
Let $w_{[v]} \equiv \{t \mid t \in w \wedge t[Z] = v\}$ and $s_{[v]} \equiv \{t \mid t \in s \wedge t[Z] = v\}$, then

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

The related *scaled measure* j_3 is defined as follows:

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

Example 6. Considering the mt-relation in Table 4, the PFD $\overline{HR}^1, \overline{SpO_2}^2 \xrightarrow[\alpha,m]{} \underline{AKI}_1$ would hold if we accept to delete tuples 1, 5, and 6. Thus, for entity, i.e., patient, Daisy we delete one tuple over 4, while for Luke we delete two tuples over 4. Thus, j_3 is 0.5.

Besides the error measures previously defined that refer to tuples of w deleted for obtaining s, we may also consider the (tuple) values of the observation TAP $\aleph$ we lose from w to s, associated with a specific (tuple) value of the prediction TAP $\beth$.

Definition 14 *(Error measure $K_3(\underline{v})$).* Given a *TF*-view $w = TFv(mtr, \alpha, m, [1,k], [1,n])$ of an mt-relation mtr with schema WT, a PFD $\aleph \xrightarrow[\alpha,m]{} \beth$, and any relation $s \subseteq w$, such that $s \models^E_{\alpha,m} \aleph \xrightarrow[\alpha,m]{} \beth$, the error measure $K_3(\underline{v})$ is expressed as in the following.

[1] The introduced definition considers for simplicity the overall attribute set Z.

Table 4
The *TF*-view, corresponding to data depicted in Table 1 for `PatientHistory`.

#	Patient	$\overline{HR}^1$	$\overline{VT}^1$	$\overline{SpO_2}^2$	$\overline{VT}^2$	$\overline{Drug}^3$	$\overline{VT}^3$	$\underline{AKI}_1$	$\underline{VT}_1$
1	Daisy	High	19	Low	21	Aspirin	23	False	28
2	Daisy	Low	2	High	4	Aspirin	6	False	18
3	Daisy	Low	2	Medium	5	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	19
8	Luke	Medium	9	High	13	Sulindac	14	True	21
10	Stevie	High	1	Low	2	Aspirin	5	False	12
11	Stevie	High	1	Low	2	Aspirin	5	False	10
12	Stevie	High	1	Low	2	Indapamide	7	False	10
36	Stevie	Medium	4	Medium	7	Metolazone	8	True	14

Let $s(\underline{\nu}) \equiv |\ \{t[\aleph] \mid \exists\, t \in s \wedge t[\urcorner] = \underline{\nu}\}\ |$ and $w(\underline{\nu}) \equiv |\ \{t[\aleph] \mid \exists\, t \in w \wedge t[\urcorner] = \underline{\nu}\}\ |$.

$$K_3(\underline{\nu}) \equiv w(\underline{\nu}) - s(\underline{\nu})$$

The related scaled measure is defined as follows

$$k_3(\underline{\nu}) \equiv \frac{w(\underline{\nu}) - s(\underline{\nu})}{w(\underline{\nu})}$$

According to Definition 14, it is straightforward to observe that $\sum_{\underline{\nu} \in \{t[\urcorner] \mid \exists\, t \in w\}} w(\underline{\nu}) \geq \sum_{\underline{\nu} \in \{t[\urcorner] \mid \exists\, t \in w\}} s(\underline{\nu})$.
Even in this case, a global error measure K_3 may be defined by considering all the possible $K_3(\underline{\nu})$ as in the following.

Definition 15 *(Global Error measure K_3).* Given a *TF*-view $w = TFv(mtr, \alpha, m, [1,k], [1,n])$ of an mt-relation mtr with schema WT, a PFD $\aleph \xrightarrow[\alpha,m]{} \urcorner$ and any relation $s \subseteq w$, such that $s \models^E_{\alpha,m} \aleph \rightarrow \urcorner$, the error measure K_3 is expressed as:

$$K_3 = \max_{\underline{\nu} \in \{t[\urcorner] \mid \exists t \in w\}} \{K_3(\underline{\nu})\}$$

while the related *scaled measure* is expressed as:

$$k_3 = \max_{\underline{\nu} \in \{t[\urcorner] \mid \exists t \in w\}} \{k_3(\underline{\nu})\}$$

Example 7. Considering again the *TF*-view in Table 3, the PFD $\overline{HR}^1, \overline{SpO_2}^2 \xrightarrow[\alpha,m]{} \underline{AKI}_1$, is not satisfied because of tuples 1, 8, and 9, and of tuples 2, 3, 5, and 6. If we delete tuples 2, 3, and 8, $k_3(True) = 1/3$ and $k_3(False) = 2/6$. Thus, $k_3 = 1/3$.

6. Deriving approximate predictive functional dependencies

From the previous Section 5, we learn that violating tuples are used to calculate an error measure. If this error is less than or equal to a given threshold ε, the predictive functional dependency $\aleph \xrightarrow[\alpha,m]{\varepsilon} \urcorner$ is approximately satisfied on s. We formally introduce the concept of Approximate Predictive Functional Dependency as follows:

Definition 16 *(Approximate Predictive Functional Dependency (APFD)).* Given a *TF*-view $w = TFv(mtr, \alpha, m, [1,k], [1,n])$ of an mt-relation mtr with schema WT, w fulfills the APFD

$$\aleph \xrightarrow[\alpha,m]{\varepsilon} \urcorner$$

written $w \models^E_{\alpha,m} \aleph \xrightarrow{\varepsilon} \urcorner$, where $\varepsilon \in 2^{\mathcal{E}}$, and $\mathcal{E} \equiv \{\varepsilon_g, \varepsilon_{pg}, \varepsilon_h, \varepsilon_j, \varepsilon_k\}$ if a relation $s \subseteq w$ exists such that: $\{t[\urcorner] \mid \exists t \in s\} = \{t[\urcorner] \mid \exists t \in w\}$ and $s \models^E_{\alpha,m} \aleph \rightarrow \urcorner$ with $g_3 \leq \varepsilon_g$, if $\varepsilon_g \in \varepsilon$; $pg_3 \leq \varepsilon_{pg}$ if $\varepsilon_{pg} \in \varepsilon$; ...; and $k_3 \leq \varepsilon_k$, if $\varepsilon_k \in \varepsilon$. In other words, $\varepsilon_g, \varepsilon_{pg}, \varepsilon_h, \varepsilon_j, \varepsilon_k$ are the maximum acceptable errors defined by the user for g_3, pg_3, h_3, j_3, and k_3, respectively.

As an APFD is inherently relevant towards 'prediction', in the definition, we may observe, besides the 'approximation'-related part considering the acceptable errors, the specific requirement that relations s and w have the same set of 'predicted' values of attributes $\urcorner$.

APFDs are thus able to mine temporal data, to extract predictive data dependencies. With respect to other AI-based approaches toward prediction, the main differences of APFDs are:

Table 5
A *TF*-view, where $\overline{HR}^1, \overline{SpO_2}^2 \xrightarrow[\alpha,m]{\epsilon} \underline{AKI_1}$ holds with $\varepsilon_g = 0.5$, $\varepsilon_h = 0.2$, $\varepsilon_j = 0.6$ and with $\varepsilon_g = 0.5$, $\varepsilon_h = 0.4$, $\varepsilon_j = 0.4$.

#	Patient	$\overline{HR}^1$	$\overline{VT}^1$	$\overline{SpO_2}^2$	$\overline{VT}^2$	$\overline{Drug}^3$	$\overline{VT}^3$	$\underline{AKI_1}$	$\underline{VT_1}$
1	Daisy	High	19	Low	21	Aspirin	23	False	28
2	Daisy	Low	2	High	4	Aspirin	6	False	18
3	Daisy	Low	2	Medium	5	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	19
8	Luke	Medium	9	High	13	Sulindac	14	True	21
10	Stevie	High	1	Low	2	Aspirin	5	False	12
11	Stevie	High	1	Low	2	Aspirin	5	False	10
12	Stevie	High	1	Low	2	Indapamide	7	False	10
13	Stevie	High	1	Low	2	Indapamide	7	True	10
14	Stevie	High	1	Low	2	Indapamide	7	True	12
15	Stevie	High	1	Low	5	Indapamide	7	True	11
17	Stevie	High	1	Low	5	Indapamide	7	True	10
18	Stevie	Low	1	Medium	6	Indapamide	7	True	10
36	Stevie	Medium	4	Medium	7	Metolazone	8	True	14

- APFDs *do not learn from data*, as they only represent temporal dependencies existing in the data. Thus, partitioning the considered dataset into training, test, and validation data is not meaningful. Similarly, using precision, recall, AUC, and other error metrics for evaluating APFDs is out of scope. Indeed, in this case, there is no ground truth to use for some kind of comparison.
- As a complement to the previous comment, APFDs *are not able to provide any kind of prediction* for attribute values not appearing in the given dataset. The strength of APFDs is in discovering predictive temporal patterns in the existing data.

Example 8. In this example, we show the use of three error measures: g_3, h_3, j_3. Suppose that our final goal is to preserve at least the 50% of the tuples ($\varepsilon_g = 0.5$), the 80% of the patients ($\varepsilon_h = 0.2$), and the 40% of the tuples for each patient ($\varepsilon_j = 0.6$). In Table 5, the PFD $\overline{HR}^1, \overline{SpO_2}^2 \xrightarrow[\alpha,m]{\varepsilon} \underline{AKI_1}$ is satisfied by considering a (sub)instance s discarding tuples 1, 2, 3, 4, 10, 11, 12. Thus, in this case, $g_3 = 7/17$, $h_3 = 1/3$, as all tuples of Daisy are discarded, and $j_3 = 3/9$ as we delete tuples of Stevie, besides those 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 1, 2, 10, 11, 12, 18, and 36, we would observe that the PFD is still satisfied, while $g_3 = 7/17$, $h_3 = 0$, and $j_3 = max\{2/4, 5/9\} = 5/9$. In this case, all the errors are below or equal to the given thresholds. Thus, we can say that $w \models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \xrightarrow{\epsilon} \underline{AKI_1}$ with $\epsilon \equiv \langle 0.5, 0.2, 0.6 \rangle$.

If we set the error thresholds as $\varepsilon_g = 0.5$, $\varepsilon_h = 0.4$, and $\varepsilon_j = 0.4$ (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 $s \models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \rightarrow \underline{AKI_1}$, while $s' \not\models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \rightarrow \underline{AKI_1}$. Thus, $w \models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \xrightarrow{\epsilon} \underline{AKI_1}$ also with $\epsilon \equiv \langle 0.5, 0.4, 0.4 \rangle$.

As we are interested in finding the minimum predictive attribute set, we introduce the definition of a minimal APFD, described as:

Definition 17 *(Minimal APFD).* An APFD $\aleph \xrightarrow[\alpha,m]{\varepsilon}$ is minimal for w, if $w \models^E_{\alpha,m} \aleph \xrightarrow{\varepsilon} \beth$ and $\forall \overline{\aleph} \subset \aleph$ we have that $w \not\models^E_{\alpha,m} \overline{\aleph} \xrightarrow{\varepsilon} \beth$.

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

Example 9. Considering the mt-relation w depicted in Table 5, it is straightforward to observe that the following two APFDs hold for $\epsilon \equiv \langle 0.35, 0.4, 0.4 \rangle$ and are minimal.

$$w \models^E_{\alpha,m} \overline{HR}^1, \overline{SpO_2}^2 \xrightarrow{\varepsilon} \underline{AKI_1}$$
$$w \models^E_{\alpha,m} \overline{Drug}^3 \xrightarrow{\varepsilon} \underline{AKI_1}$$

As for the minimality of the first APFD, both $\overline{SpO_2}^2 \xrightarrow[\alpha,m]{\varepsilon} \underline{AKI_1}$ and $\overline{HR}^1 \xrightarrow[\alpha,m]{\varepsilon} \underline{AKI_1}$ cannot satisfy the first threshold, i.e., $g_3 \leq 0.35$.

As a final property of APFDs, it is straightforward to prove that
$w \models^E_{\alpha,m} \aleph \xrightarrow{\varepsilon} \beth_1$ and $w \models^E_{\alpha,m} \aleph \xrightarrow{\varepsilon} \beth_2$ does not imply $w \models^E_{\alpha,m} \aleph \xrightarrow{\varepsilon} \beth_1 \beth_2$.

6.1. Algorithmic issues of deriving an APFDs

It is well known that the complexity of deriving AFDs is exponential in the number of attributes [71,20]. As for data complexity, different approaches can be taken when considering algorithms that allow one to check whether a TF-view fulfills a given APFD. In general, as proved in [31], even considering only errors g_3 and h_3, the problem of checking an APFD with respect to a TF-view is NP-hard. More specifically, 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*.

Algorithm 1: DeterministicADC.

Input: a TF-view w, a given PFD $\aleph \to \beth$ 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. $s \models \aleph \to \beth$, $g_3(w, s) \leq \epsilon_{g_3}$, $h_3(w, s) \leq \epsilon_{h_3}$, $j_3(w, s) \leq \epsilon_{j_3}$

```
▷ Prepare data for an initial call according to epsilons
1  begin
2      del ← ⌊ε_{g3} |w|⌋
3      count ← ⌊ε_{h3} |{t[Z] | t ∈ w}|⌋
4      for z ∈ {t[Z] | t ∈ w}: do
5          thresholds[z] ← ⌊ε_{j3} |{t : t[Z] = z}|⌋
6      s ← RecADC(w, ℵ → ב, del, count, thresholds)
7      if {t[ב] | t ∈ w} = {t[ב] | t ∈ s} then
8          return s
9      return fail
10 Function RecADC(w, ℵ → ב, del, count, thresholds):
       ▷ This is the last recursive call before success
11     if w = ∅ then
12         return ∅
13     let v̄ ∈ {t[ℵ] | t ∈ w}
       ▷ For each value of ב
14     for v ∈ {t[ב] : t ∈ w} do
           ▷ del_tuples: tuples removed according to selection
15         del_tuples ← {t | t ∈ w ∧ t[ℵ] = v̄ ∧ t[ב] ≠ v}
16         s ← {t | t ∈ w ∧ t[ℵ] = v̄ ∧ t[ב] = v}
17         out ← {}
18         for z ∈ {t[Z] | t ∈ del_tuples} do
19             thresholds'[z] ← thresholds[z] − |{t | t[Z] = z ∧ t ∈ del_tuples}|
20             if thresholds'[z] < 0 then
21                 out ← out ∪ {z}
           ▷ out: the z groups that must disappear, since their tuples passed below the threshold ε_{j3} in the
           current state
22         if count − |out| ≥ 0 then
               ▷ count': represent the z groups still to be considered
23             count' ← count − |out|
24             del_tuples ← del_tuples ∪ {t : t ∈ w ∧ ∃z(t[Z] = z ∧ z ∈ out)}
25             if del − |del_tuples| ≥ 0 then
                   ▷ If the final test succeeds, we proceed with the recursive call on the updated values
26                 del' ← del − |del_tuples|
27                 w' ← w \ (del_tuples ∪ s)
28                 s' ← RecADC(w', del', count', thresholds')
29                 if s' ≠ fail then
30                     return s ∪ s'
31     return fail
```

Algorithm 1 provides the pseudo-code of the deterministic algorithm that stops the analysis of a relation, as soon as it verifies that the relation cannot satisfy the given APFD. It is a generalization and a refinement of the algorithm we proposed in [31], to manage possibly many and multivalues predicted attributes, by considering the refined definition of APFD. The general idea of this algorithm is to search for a solution considering one tuple at a time until it is possible to generate a solution that satisfies the selected thresholds. Throughout the code, w is the entire TF-view. *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 w, for each value $\overline{v}$ taken by the observation TAP $\aleph$, we try one value for TAP $\beth$ and verify the dependency; if it fails, we try the second value for $\beth$ and verify the dependency, and so on. If all the choices fail, then the algorithm fails. Inside the body of the loop starting at line 14 of Algorithm 1 we check a candidate value $\underline{v}$ for $\beth$ to be assigned as the consequent for the current value $\overline{v}$ of $\aleph$. This amounts of guessing that all the tuples $t \in w$ with $t[\aleph\beth] = \overline{v}\underline{v}$ will be kept in the final instance $s \subseteq w$. Such a guess unambiguously determines both a set s of tuples that

will be kept in the final solution (line 16) and a set *del_tuples* of tuples that will not belong to the final solution (lines 15 and 24), respectively. In order to reflect such a guess, new thresholds del', $count'$, and $thresholds'$ are computed according to their old values and the cardinalities $|s|$ and $|del\_tuples|$ (lines 19, 23, and 26). More importantly, all tuples in both s and *del_tuples* are removed from w (line 27), thus obtaining the new *TF*-view w', on which the recursive call (line 28) will be performed with the updated thresholds. It is worth noticing, that if the recursive call succeeds, it breaks the for-loop of line 14 without testing the remaining values of $\urcorner$. This is correct because, in the presence of a successful return from the recursive call, we have that the desired solution s has been found. If one of the $\urcorner$ values satisfies the thresholds, we update the counters, building at every step an intermediate relation s', as long as the thresholds are satisfied. After the end of all the recursive calls of `RecADC()`, the main algorithm has only to finally check that s still contains all the values that were in w for the predicted attribute(s), as in lines 7 and 8.

The other thresholds, we introduced in Section 5, can be easily embedded in the proposed algorithm, following the same approach we considered for g_3, h_3, and j_3, respectively. The proposed algorithm returns a possible set $s \subset w$ (if any), proving that w satisfies the considered APFD. No maximality for s has been defined, as the threshold is now a vector of different values, related to different kinds of errors, and a kind of maximality cannot be easily defined.

Another possible and less computationally expensive strategy may consist of leveraging well-known algorithms checking approximate dependencies only considering error g_3. As for the first experimental evaluations in [31], we adopted a sub-optimal solution, on top of the well-known TANE [71] algorithm, a popular approximate functional dependency detection algorithm, customizing it to mine only approximate functional dependencies with a fixed consequent, the predicted attribute $\urcorner$. Data complexity of TANE is linear in the number of tuples. Given *TF*-view w and the predicted attribute $\urcorner$, our approach was mainly based on the following steps:

- Derive the maximal 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 $j_3 \leq \varepsilon_j$.

This approach allows the extraction of APFDs that are satisfied by w according to the given thresholds. However, it could exclude other APFDs that are associated with another s, not maximal, but still satisfying $g_3 \leq \varepsilon_g$, which could also satisfy the other thresholds.

7. Evaluating the informative content of APFDs

Let us now consider a possible approach to evaluate the quality of the derived APFDs. Such approach will be based on well-known concepts in Information Theory [72], suitably adapted to functional dependencies.

Given an APFD $\aleph \xrightarrow[\alpha,m]{\varepsilon} \urcorner$ and a *TF*-view w which fulfills it according to Definition 16, we define the entropy of $\urcorner$ in w, denoted as $E_{\urcorner}(w)$, as:

$$E_{\urcorner}(w) = \sum_{\underline{v} \in \{t[\urcorner] \mid t \in w\}} \frac{|\{t \mid t \in w \wedge t[\urcorner] = \underline{v}\}|}{|w|} \log_2 \left(\frac{|w|}{|\{t \mid t \in w \wedge t[\urcorner] = \underline{v}\}|} \right)$$

Example 10. The entropy of $\underline{AKI}_1$ for w provided in Table 3 is $E_{\underline{AKI}_1}(w) = 0.918$.

Under the same premises, we may define the *Information Gain* of an APFD, denoted by $IG_{\aleph \rightarrow \urcorner}(w)$.

Definition 18 *(Information Gain of an APFD ($IG_{\aleph \rightarrow \urcorner}$)).* Given an APFD $\aleph \xrightarrow[\alpha,m]{\varepsilon} \urcorner$ and a *TF*-view w which fulfills it according to Definition 16, we define the Information Gain of such an APFD as

$$IG_{\aleph \rightarrow \urcorner}(w) = E_{\urcorner}(w) - \sum_{\overline{v} \in \{t[\aleph] \mid t \in w\}} \frac{|\{t \mid t[\aleph] = \overline{v} \wedge t \in w\}|}{|w|} E_{\urcorner}(\{t \mid t[\aleph] = \overline{v} \wedge t \in w\})$$

In the context of machine learning, Information Gain is a quite renowned measure, which is mainly used for finding the best split in a node of decision tree [73]. Here, we use Information Gain to measure the information content of approximate predictive functional dependencies.

It is easy to prove that $E_{\urcorner}(w) \geq IG_{\aleph \rightarrow \urcorner}(w)$ for every APFD $\aleph \xrightarrow[\alpha,m]{\varepsilon} \urcorner$ and every *TF*-view w.

Example 11. The Information Gain of $\overline{HR}^1\overline{SpO_2}^2\overline{Drug}^3 \rightarrow \underline{AKI}_1$ in Table 3 is $IG_{\overline{HR}^1\overline{SpO_2}^2\overline{Drug}^3 \rightarrow \underline{AKI}_1}(w) = 0.612$ while $IG_{\overline{HR}^1\overline{SpO_2}^2 \rightarrow \underline{AKI}_1}(w) = 0.167$.

In this context, Information Gain represents the drop in Entropy for the predicted attribute(s) ℸ in TF-view w, if we have information about the value of TAP ℵ. However, if the entropy $E_{\beth}(w)$ in the current *TF*-view w is very low, such a drop may not be carrying the real informative content made by the $\aleph \xrightarrow[\alpha,m]{\varepsilon} \beth$ under examination.

Example 12. Let us consider an APFD $\aleph \xrightarrow[\alpha,m]{\varepsilon} \beth$, where ℸ is a single boolean attribute, and a *TF*-view w for which $E_{\beth}(w) = 0.23$, and an Information Gain $IG_{\aleph \to \beth}(w) = 0.11$. We may interpret it as "knowing the values for the attributes in ℵ would provide us almost twice the information necessary for determining the value of ℸ, compared to having the information on the distribution of ℸ alone". However, this is not reflected in this case by the related small value of 0.11 for the Information Gain.

In the provided example we considered a cardinality $|\{t[\beth] \mid t \in w\}| = 2$, i.e., the binary case. When we consider an arbitrary cardinality of $\{t[\beth] \mid t \in w\}$, the Entropy value $E_{\beth}$ may range from 0 to $\log_2(|\{t[\beth] \mid t \in w\}|)$. It means that we lose the desirable property of having values of $IG_{\aleph \to \beth}(w)$ between 0 and 1 for cardinalities of ℸ greater than 2. For such reasons, we consider the *Information Gain Ratio* defined as follows:

$$IG\text{-}R_{\aleph \to \beth}(w) = \frac{IG_{\aleph \to \beth}(w)}{E_{\beth}(w)}$$

It is easy to see that for any cardinality of $\{t[\beth] \mid t \in w\}$, it holds $0 \leq IG\text{-}R_{\aleph \to \beth}(w) \leq 1$.

Example 13. With $E_{\beth}(w) = 0.23$ and $IG_{\aleph \to \beth}(w) = 0.11$, we have $IG\text{-}R_{\aleph \to \beth}(w) = 0.478$ which reflects better the (significant) drop in entropy for such small starting entropy.

Digging deeper into the predictivity features of APFDs, a low value for $E_{\beth}$ entails a class-unbalance problem. Let us now consider the set s returned by Algorithm 1 which satisfies $\aleph \to \beth$ on w provided, thresholds ε_* with $* \in \{g_3, h_3, j_3\}$ (assuming that such s exists.)

Since Algorithm 1 capitalizes on non-determinism to possibly achieve s and the provided thresholds are not aimed to preserve class distributions, we may end up with an $s \subseteq w$ for which $E_{\beth}(w)$ and $E_{\beth}(s)$ are very different, despite the fact that we have checked $|w| - |s|$ via threshold ε_{g_3}. Informally speaking, if we consider s from a prediction perspective, we may use the set of its induced association rules as an actual classifier: $AR_{\aleph \to \beth}(s) = \{t[\aleph], t[\beth] \mid t \in s\}$. If values $E_{\beth}(s)$ and $E_{\beth}(w)$ are "too much far away" we have two opposite but still undesirable situations: (i) most of the tuples t in w are removed, i.e., $t \in w \setminus s$, when $t[\beth]$ belong to a one or more among the *less represented classes*; (ii) a disproportion of tuples t in w is removed, when $t[\beth]$ belong to a one or more among the *most represented classes*.

Case (i) is a manifestation of the "tyranny of the masses" effect in which valuable informative outliers are crushed in order to meet the thresholds. Case (ii) is a manifestation of the "overfitting" effect in which noise is amplified in order to meet the thresholds since, in this case, even a significant drop in the cardinalities of the most represented classes is counterbalanced by the sheer determination to keep noisy data. Both these cases may be limited by suitably tuning some thresholds for error pg_3, or the different $pg_3(\underline{v})$ for any $\underline{v} \in \{t[\beth] \mid t \in w\}$, as discussed in Section 5. Since cases (i) and (ii) above are at opposite ends (is a sample of an underrepresented class a noisy or a valuable one?), at first, it may seem to be difficult to find a single measure whose value is able to detect simultaneously if we are in one of such cases. Information theory helps with a simple, elegant measure. Such a measure is the Kullback-Leibler divergence (KL), which can be declined in our context as follows:

$$KL_{\aleph \to \beth}(w, s) = \sum_{\underline{v} \in \{t[\beth] \mid t \in w\}} \frac{|\{t \mid t \in w \wedge t[\beth] = \underline{v}\}|}{|w|} \log_2 \left(\frac{|s|}{|w|} \cdot \frac{|\{t \mid t \in w \wedge t[\beth] = \underline{v}\}|}{|\{t \mid t \in s \wedge t[\beth] = \underline{v}\}|} \right)$$

In the context of APFDs, $KL_{\aleph \to \beth}(w, s)$ measures the difference in entropy between the original distribution of ℸ in w and the one in s, that is, the average number of additional bits of information we need if we are transmitting the value of ℸ according to its distribution in w using a message encoding built using the distribution of ℸ in s instead. A variant of KL, namely J-measure, has been introduced for Association Rule Mining (ARM) in [74] as an auxiliary interestingness measure for pruning rules extracted by the standard support/confidence thresholds. Analogously to what we have done for information gain, we can scale $KL_{\aleph \to \beth}(w, s)$ as follows:

$$KL\text{-}R_{\aleph \to \beth}(w, s) = \begin{cases} \frac{KL_{\aleph \to \beth}(w,s)}{E_{\beth}(w)} & \text{if } KL_{\aleph \to \beth}(w, s) \leq E_{\beth}(w); \\ 1 & \text{otherwise.} \end{cases}$$

The piece-wise function definition of $KL\text{-}R_{\aleph \to \beth}(w, s)$ is needed because, unlike the information gain, $KL_{\aleph \to \beth}(w, s)$ may in principle exceed the original entropy of $E_{\beth}(w)$. In this case, we are interested in amplifying the measure when the divergence is significant, even for a small value of $E_{\beth}(w)$. Unlike the information gain, here we are interested in obtaining a small value for $KL\text{-}R_{\aleph \to \beth}(w, s)$, which indicates that the distribution of ℸ in s resembles the one in w.

Example 14. Let us consider the example of Table 3 and the two APFDs $\overline{HR}^1 \rightarrow \underline{AKI}_1$ and $\overline{SpO_2}^2 \rightarrow \underline{AKI}_1$. Let us consider first $\overline{HR}^1 \rightarrow \underline{AKI}_1$, which has been extracted for some given values for the thresholds, producing a set s discarding tuples 4, 5, and 6. For such a scenario, we have $KL_{\overline{HR}^1 \rightarrow \underline{AKI}_1}(w, s) = 0.08$ and $KL\text{-}R_{\overline{HR}^1 \rightarrow \underline{AKI}_1}(w, s) = 0.13$.

As for the second APFD, $\overline{SpO_2}^2 \rightarrow \underline{AKI}_1$ has been extracted for some given value for the thresholds, producing a set s' discarding tuples 2 and 36. For such a scenario, we have $KL_{\overline{SpO_2}^2 \rightarrow \underline{AKI}_1}(w, s') = 0.005$ and $KL\text{-}R_{\overline{SpO_2}^2 \rightarrow \underline{AKI}_1}(w, s') = 0.008$.

As the two APFDs may have been extracted with similar threshold values, we may conclude that s' witnessing $\overline{SpO_2}^2 \rightarrow \underline{AKI}_1$ has a far better divergence ratio than s, which witnesses $\overline{HR}^1 \rightarrow \underline{AKI}_1$. Finally, if we consider that they have pretty similar Information Gain Ratios, i.e., $IG\text{-}R_{\overline{HR}^1 \rightarrow \underline{AKI}_1}(w) = 0.27$ and $IG\text{-}R_{\overline{SpO_2}^2 \rightarrow \underline{AKI}_1}(w) = 0.33$, the divergence ratio may be used for ranking APFDs with similar Information Gain Ratios.

8. Deriving APFDs: an experimental evaluation

In this Section, we provide some results from an experimental evaluation of real-world clinical data. Such clinical information is related to the discovery of predictive dependencies, allowing the exploration of temporal patterns of clinical data related to following AKI diagnosis, according to the criteria discussed in Section 3.

8.1. System configuration

To mine APFDs in the clinical dataset, we run the tests on a server with 16 cores, 12 GB of RAM, 1TB disk, equipped with Ubuntu 18.04, and Postgres 12. We mine the APFDs from the *TF*-views on a machine with a 2,3 GHz Intel Core i9 8 core, 16 GB of RAM, equipped with macOS Sonoma 14.0, and Python 3.

8.2. The clinical dataset: MIMIC III

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) 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 [39]. MIMIC III database is managed within Physionet, a broad initiative for sharing complex heterogenous clinical data and software tools, focusing on the management of biomedical physiological data [75].

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.

8.3. Data preprocessing and transformation

We employed seven tables from the original dataset, subjecting them to an ETL (Extract, Transform, Load) process. The reference tables, *D_ITEMS* and *D_LABITEMS*, were involved in labeling every measure associated with a patient. Information regarding ICU admission, discharge, and age was extracted from *PATIENTS* and *ICUSTAYS*. The *PRESCRIPTIONS* table provided details on administered medications, particularly focusing on four categories: diuretics, Non-steroidal anti-inflammatory drugs (NSAID), radiocontrast agents, and angiotensin. *LABEVENTS* contributed information on serum creatinine and urine, while *CHARTEVENTS* supplied data on heart rate, diastolic pressure, and oxygen saturation.

The preprocessing and transformation tasks may be summarized as follows:

- Categorizing numerical values. We stratified numerical variables into “low, medium, high” based on established clinical literature.
- Identifying AKI patients. The heaviest preprocessing task was related to identifying AKI patients. Indeed, such diagnosis is not stored in the MIMIC III dataset and has been derived by the clinical dataset, according to the clinical criteria discussed in Section 3. It is worth noting that such a task is computationally expensive as the criteria for identifying AKI patients are inherently temporal and require advanced temporal query specification.
- Building different *TF*-views, considering different attributes and different time frames. The considered attributes were selected according to the knowledge of the specific medical domain deriving from discussions with clinical experts [76].

For the analysis, we considered a time frame characterized by an OW lasting 72 hours, followed by a WW of 12 hours, and then a PW of 36 hours. The PW was specifically set to capture the (potential) onset of the illness in accordance with one of the KDIGO criteria [66].

Based on the literature [77], we focused on six measures: creatinine, administered drugs, respiratory rate, oxygen saturation, and diastolic blood pressure. Using a cohort of 50.711 patients, we considered three different *TF*-views:

Table 6
APFDs obtained from the three *TF*-views.

APFD	ε_g	ε_h	ε_j	*TF*-view	Algorithm
$\overline{Creat}^1, \overline{Creat}^3 \rightarrow \underline{AKI}_1$	27,45%	27%	50%	#1	TANE
$\overline{Creat}^1, \overline{Creat}^4 \rightarrow \underline{AKI}_1$	27,45%	27%	50%	#1	TANE
$\overline{Creat}^1, \overline{Creat}^3 \rightarrow \underline{AKI}_1$	27,45%	27%	50%	#1	Algorithm 1
$\overline{Creat}^1, \overline{Creat}^4 \rightarrow \underline{AKI}_1$	27,45%	27%	50%	#1	Algorithm 1
$\overline{Creat}^2, \overline{Creat}^3, \overline{Creat}^4 \rightarrow \underline{AKI}_1$	27,45%	27%	50%	#1	Algorithm 1
$\overline{Drug}^1, \overline{Drug}^2, \overline{Drug}^4 \rightarrow \underline{AKI}_1$	21%	30%	50%	#2	TANE
$\overline{Drug}^1, \overline{Drug}^2, \overline{Drug}^4 \rightarrow \underline{AKI}_1$	21%	30%	50%	#2	Algorithm 1
$\overline{AdministeredDrug}^1, \overline{RespiratoryRate}^3 \rightarrow \underline{AKI}_1$	10%	51%	75%	#3	TANE
$\overline{AdministeredDrug}^1 \rightarrow \underline{AKI}_1$	10%	51%	75%	#3	Algorithm 1
$\overline{RespiratoryRate}^3 \rightarrow \underline{AKI}_1$	10%	51%	75%	#3	Algorithm 1
$\overline{AdministeredDrug}^1, \overline{DiastolicPressure}^2 \rightarrow \underline{AKI}_1$	30%	50%	90%	#3	TANE
$\overline{DiastolicPressure}^2, \overline{RespiratoryRate}^3 \rightarrow \underline{AKI}_1$	30%	50%	90%	#3	TANE
$\overline{AdministeredDrug}^1 \rightarrow \underline{AKI}_1$	30%	50%	90%	#3	Algorithm 1
$\overline{RespiratoryRate}^3 \rightarrow \underline{AKI}_1$	30%	50%	90%	#3	Algorithm 1

- *TF*-view #1 involves serum creatinine and employs four observation-related valid times of the same measure. The sequence comprises four values of serum creatinine, where each value is the subsequent one (if any) within the specified time frame. We identified 2546 subjects (1878 patients without AKI, 668 patients with AKI) with 3839 rows;
- *TF*-view #2 includes administered drugs. We used four valid times of the same measure to build a sequence of four values of a measure, where any value is the next of the preceding one (if any) within the time frame. We identified 148 subjects (109 patients without AKI, 39 patients with AKI) with 1047 rows;
- *TF*-view #3 deals with four observation-related valid times, 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 ranging from 2 to 4 within the time frame. We obtained 413 subjects (305 patients without AKI, 108 patients with AKI) with 193.173 rows.

8.4. Experiments and results

As for the experiments, we applied the program we implemented, based on Algorithm 1, and focused on the discovery of minimal APFDs. Different thresholds for the different *TF*-views were set. Moreover, we compared our results with those obtained through a slightly modified version of TANE, the well-known algorithm for the discovery of (standard) approximate functional dependencies [71]. The version we used has been adapted as discussed in Section 6 to derive only data dependencies having the AKI attribute as consequent and checking the thresholds in a hierarchical way, starting from ϵ_g.

In Table 6, we reported some of the main APFDs obtained through the two different algorithms, TANE and Algorithm 1, with the corresponding error thresholds. The computational performances are comparable in the order of seconds. In general, TANE took a few seconds less than Algorithm 1 to produce the results. This difference corresponds to the fact that the complexity of TANE is linear with respect to the number of tuples, while our Algorithm 1 may require exponential time.

Considering the first *TF*-view, the first two APFDs in Table 6 are derived by both algorithms. The same happens for the only dependency derived from the second *TF*-view. From the first *TF*-view, our Algorithm 1 derives another APFD $\overline{Creat}^2, \overline{Creat}^3, \overline{Creat}^4 \rightarrow \underline{AKI}_1$. This is due to the fact that our algorithm checks different sets s to find one satisfying the given APFD. Such s has not to be maximal with respect to a single error threshold. On the other side, TANE finds the maximal s with respect to g_3. If such a set s does not satisfy the other error thresholds, the considered APFD candidate is discarded. Such a hierarchical approach by TANE also explains the results obtained for the third *TF*-view. In this case, our algorithm was able to find shorter minimal APFDs such as $\overline{RespiratoryRate}^3 \rightarrow \underline{AKI}_1$ compared to $\overline{DiastolicPressure}^2, \overline{RespiratoryRate}^3 \rightarrow \underline{AKI}_1$ derived by TANE.

The last part of the experimental evaluation is related to provide some example of the inherent explainability of our proposal. Indeed, after obtaining the relevant APFDs according to specific thresholds, it is possible to analyze the most common patterns for values associated with AKI patients (or without AKI) related to the attributes of the derived APFD.

Tables 7, 8 and 9, report the attribute values related to three APFDs, each one derived from a different *TF*-view, to show which are the values corresponding to AKI vs non AKI patients. As we can easily observe, the number of different patterns for attribute values is quite different for the three APFDs.

For the first two *TF*-views, there is only one value combination associated with $\underline{AKI}_1 = 1$. It could be not surprising as the sequence of values for creatinine and drugs may contain many patterns quite generic, appearing in most patients.

In the third *TF*-view, we have a prevailing value for $\overline{RespiratoryRate}^3$ that describes $\underline{AKI}_1 = 1$.

Table 7
Value combinations of $\overline{Creat}^2, \overline{Creat}^3\overline{Creat}^4 \rightarrow \underline{AKI}_1$ with thresholds $\varepsilon_g = 27.45\%$, $\varepsilon_h = 27\%$ and $\varepsilon_j = 50\%$.

$\overline{Creat}^2$	$\overline{Creat}^3$	$\overline{Creat}^4$	$\underline{AKI}_1$	Count
medium	medium	medium	0	1510
high	high	high	0	506
low	low	low	0	280
medium	medium	low	0	84
high	medium	medium	0	78
high	high	medium	0	63
medium	low	low	0	63
low	medium	medium	0	48
medium	low	medium	0	46
low	low	medium	0	45
low	medium	low	0	24
medium	high	medium	0	14
medium	medium	high	0	12
medium	high	high	1	9
high	medium	high	0	6

Table 8
Value combinations of $\overline{Drug}^1, \overline{Drug}^2\overline{Drug}^4 \rightarrow \underline{AKI}_1$ with thresholds $\varepsilon_g = 21\%$, $\varepsilon_h = 30\%$ and $\varepsilon_j = 50\%$.

$\overline{Drug}^1$	$\overline{Drug}^2$	$\overline{Drug}^4$	$\underline{AKI}_1$	Count
diuretics	diuretics	diuretics	0	270
diuretics	diuretics	angiotensin	0	74
diuretics	angiotensin	diuretics	0	66
nsaid	diuretics	diuretics	0	61
angiotensin	angiotensin	diuretics	0	60
angiotensin	diuretics	diuretics	0	57
diuretics	angiotensin	angiotensin	0	48
angiotensin	angiotensin	angiotensin	0	44
angiotensin	diuretics	angiotensin	0	38
nsaid	diuretics	angiotensin	0	29
angiotensin	angiotensin	nsaid	0	15
nsaid	angiotensin	angiotensin	0	13
nsaid	angiotensin	diuretics	0	12
nsaid	diuretics	nsaid	0	10
diuretics	diuretics	nsaid	0	8
angiotensin	nsaid	angiotensin	0	5
nsaid	nsaid	diuretics	0	5
angiotensin	nsaid	diuretics	0	4
diuretics	nsaid	diuretics	1	4
diuretics	angiotensin	nsaid	0	2
angiotensin	diuretics	nsaid	0	1
diuretics	nsaid	nsaid	0	1
nsaid	nsaid	nsaid	0	1

Table 9
Values of $\overline{RespiratoryRate}^3 \rightarrow \underline{AKI}_1$ with thresholds $\varepsilon_g = 10\%$, $\varepsilon_h = 51\%$ and $\varepsilon_j = 75\%$.

$\overline{RespiratoryRate}^3$	$\underline{AKI}_1$	Count
high	1	188462
medium	0	543
low	0	43

9. Discussion and 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. Approximate predictive functional dependencies have been proposed for a completely new kind of attribute timestamped temporal relation, named after *multi-temporal relation*, where attribute values holding at different timestamps allow the representation of histories of specific domain-related entities. We analyzed the approximation concept specifying different kinds of error, some already presented in [31], while pG_3 and K_3 are completely new. Such new error measures allow the user to focus on

(i) the error, i.e., the number of tuples violating the dependency, related to any specific attribute(s) value considered in the prediction (i.e., the consequent of the dependency, and (ii) on the number of pattern values of the antecedent associated with a predictive (consequent) attribute value that has to be deleted for satisfying the given APFD.

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 measures G_3 and H_3. We reduced the problem in hand to a general 3*SAT* problem, demonstrating that verifying an APFD with all three thresholds belongs to the *NP* complexity class.

We discussed a novel method for assessing the 'quality' of the resulting APFDs, using entropy-based concepts.

Finally, we implemented the Algorithm 1 and 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.

APFDs are completely agnostic with respect to the application domain. Indeed they can be used in any other domains where the primary challenge involves identifying temporal patterns from the past associated with subsequent (future) events in a prediction-oriented manner. In this regard, domain knowledge is required to build the right mt-relation and the most suitable time frames. Indeed, building the mt-relation is a task left to the user, that requires some specific effort and deep knowledge of the application domain. A further limitation of the proposed APFDs is that also time frames have to be specified by the user. While it is meaningful with respect to the requirements of the considered application domain, it would be interesting to have also the most suitable time frames discovered during the temporal data mining.

According to these two last highlighted limitations, and considering also the explainability features of APFDs, we plan to extend APFDs in different directions: (i) extending APFDs to discover suitable time windows, according to the considered dataset; (ii) coupling APFDs with some ML techniques applied to temporal data, to both have some indications about the most important attributes to consider in building mt-relations and to support explainability of results from black-box ML approaches.

CRediT authorship contribution statement

Beatrice Amico: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Carlo Combi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Romeo Rizzi:** Formal analysis. **Pietro Sala:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Carlo Combi reports financial support was provided by University of Verona Department of Computer Science. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank the Master's students, Simone Scaglia and Stefano Tanfoglio, for their help in implementing the algorithms for the experimental evaluation.

The research leading to these results has received funding from the European Union—NextGenerationEU through the Italian Ministry of University and Research under PNRR-M4C2-I1.3 Project PE_00000019 "HEAL ITALIA" to Carlo Combi CUP B33C22001030006. The views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

[1] O. Maimon, L. Rokach, Introduction to knowledge discovery and data mining, in: Data Mining and Knowledge Discovery Handbook, Springer, 2009, pp. 1–15.
[2] J. Han, Data mining, in: L. Liu, M.T. Özsu (Eds.), Encyclopedia of Database Systems, second edition, Springer, 2018, pp. 797–802.
[3] R. Moskovitch, Y. Shahar, Fast time intervals mining using the transitivity of temporal relations, Knowl. Inf. Syst. 42 (1) (2015) 21–48.
[4] P. Sala, C. Combi, M. Mantovani, R. Rizzi, Discovering evolving temporal information: theory and application to clinical databases, SN Comput. Sci. 1 (3) (2020) 153, https://doi.org/10.1007/s42979-020-00160-9.
[5] M. Arora, U. Kanjilal, D. Varshney, Evaluation of information retrieval: precision and recall, Int. J. Indian Cult. Bus. Manag. 12 (2) (2016) 224–236.
[6] M. Buckland, F. Gey, The relationship between recall and precision, J. Am. Soc. Inf. Sci. 45 (1) (1994) 12–19.
[7] N.R. Cook, Comments on 'evaluating the added predictive ability of a new marker: from area under the roc curve to reclassification and beyond' by M.J. Pencina et al., statistics in medicine, Stat. Med. 27 (2) (2008) 191–195.
[8] D. Castelvecchi, Can we open the black box of AI?, Nat. News 538 (7623) (2016) 20.
[9] A. Jazayeri, C.C. Yang, M. Capan, Frequent temporal patterns of physiological and biological biomarkers and their evolution in sepsis, Artif. Intell. Med. 143 (2023) 102576, https://doi.org/10.1016/J.ARTMED.2023.102576.

[10] R. AlSaad, Q.M. Malluhi, A.A. Abd-Alrazaq, S. Boughorbel, Temporal self-attention for risk prediction from electronic health records using non-stationary kernel approximation, Artif. Intell. Med. 149 (2024) 102802, https://doi.org/10.1016/J.ARTMED.2024.102802.
[11] G. Ifrim, R. Tavenard, A.J. Bagnall, P. Schäfer, S. Malinowski, T. Guyet, V. Lemaire (Eds.), Advanced Analytics and Learning on Temporal Data - 8th ECML PKDD Workshop, Revised Selected Papers, AALTD 2023, Turin, Italy, September 18–22, 2023, Lecture Notes in Computer Science, vol. 14343, Springer, 2023.
[12] F. Alhaek, W. Liang, T.M. Rajeh, M.H. Javed, T. Li, Learning spatial patterns and temporal dependencies for traffic accident severity prediction: a deep learning approach, Knowl.-Based Syst. 286 (2024) 111406, https://doi.org/10.1016/J.KNOSYS.2024.111406.
[13] D. Ma, B. Zhou, X. Song, H. Dai, A deep reinforcement learning approach to traffic signal control with temporal traffic pattern mining, IEEE Trans. Intell. Transp. Syst. 23 (8) (2022) 11789–11800, https://doi.org/10.1109/TITS.2021.3107258.
[14] C. Combi, B. Amico, R. Bellazzi, A. Holzinger, J.H. Moore, M. Zitnik, J.H. Holmes, A manifesto on explainability for artificial intelligence in medicine, Artif. Intell. Med. 133 (2022) 102423, https://doi.org/10.1016/J.ARTMED.2022.102423.
[15] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2019) 93, https://doi.org/10.1145/3236009.
[16] M. Langer, D. Oster, T. Speith, H. Hermanns, L. Kästner, E. Schmidt, A. Sesing, K. Baum, What do we want from explainable artificial intelligence (XAI)? - a stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research, Artif. Intell. 296 (2021) 103473, https://doi.org/10.1016/j.artint.2021.103473.
[17] L. Caruccio, V. Deufemia, F. Naumann, G. Polese, Discovering relaxed functional dependencies based on multi-attribute dominance, IEEE Trans. Knowl. Data Eng. 33 (9) (2021) 3212–3228, https://doi.org/10.1109/TKDE.2020.2967722.
[18] C. Combi, M. Mantovani, A. Sabaini, P. Sala, F. Amaddeo, U. Moretti, G. Pozzi, Mining approximate temporal functional dependencies with pure temporal grouping in clinical databases, Comput. Biol. Med. 62 (2015) 306–324, https://doi.org/10.1016/j.compbiomed.2014.08.004.
[19] L. Berti-Équille, H. Harmouch, F. Naumann, N. Novelli, S. Thirumuruganathan, Discovery of genuine functional dependencies from relational data with missing values, Proc. VLDB Endow. 11 (8) (2018) 880–892, https://doi.org/10.14778/3204028.3204032, http://www.vldb.org/pvldb/vol11/p880-berti-equille.pdf.
[20] S. Kruse, F. Naumann, Efficient discovery of approximate dependencies, Proc. VLDB Endow. 11 (7) (2018) 759–772, https://doi.org/10.14778/3192965.3192968.
[21] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995.
[22] M. Mazuran, E. Quintarelli, L. Tanca, S. Ugolini, Semi-automatic support for evolving functional dependencies, in: E. Pitoura, S. Maabout, G. Koutrika, A. Marian, L. Tanca, I. Manolescu, K. Stefanidis (Eds.), Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15–16, 2016, OpenProceedings.org, 2016, pp. 293–304.
[23] P. Schirmer, T. Papenbrock, S. Kruse, F. Naumann, D. Hempfing, T. Mayer, D. Neuschäfer-Rube, Dynfd: functional dependency discovery in dynamic datasets, in: M. Herschel, H. Galhardas, B. Reinwald, I. Fundulaki, C. Binnig, Z. Kaoudi (Eds.), Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26–29, 2019, OpenProceedings.org, 2019, pp. 253–264.
[24] W. Fan, F. Geerts, X. Jia, Semandaq: a data quality system based on conditional functional dependencies, Proc. VLDB Endow. 1 (2) (2008) 1460–1463, https://doi.org/10.14778/1454159.1454200, http://www.vldb.org/pvldb/vol1/1454200.pdf.
[25] A.A. Qahtan, N. Tang, M. Ouzzani, Y. Cao, M. Stonebraker, Pattern functional dependencies for data cleaning, Proc. VLDB Endow. 13 (5) (2020) 684–697, https://doi.org/10.14778/3377369.3377377, http://www.vldb.org/pvldb/vol13/p684-qahtan.pdf.
[26] L. Caruccio, V. Deufemia, G. Polese, Relaxed functional dependencies - a survey of approaches, IEEE Trans. Knowl. Data Eng. 28 (1) (2016) 147–165, https://doi.org/10.1109/TKDE.2015.2472010.
[27] J. Kivinen, H. Mannila, Approximate inference of functional dependencies from relations, Theor. Comput. Sci. 149 (1) (1995) 129–149, https://doi.org/10.1016/0304-3975(95)00028-U.
[28] C. Giannella, E. Robertson, On approximation measures for functional dependencies, Inf. Syst. 29 (6) (2004) 483–507, https://doi.org/10.1016/j.is.2003.10.006.
[29] C. Combi, P. Sala, Mining approximate interval-based temporal dependencies, Acta Inform. 53 (6–8) (2016) 547–585, https://doi.org/10.1007/s00236-015-0246-x.
[30] B. Amico, C. Combi, A 3-window framework for the discovery and interpretation of predictive temporal functional dependencies, in: M. Michalowski, S.S.R. Abidi, S. Abidi (Eds.), Artificial Intelligence in Medicine - 20th International Conference on Artificial Intelligence in Medicine, Proceedings, AIME 2022, Halifax, NS, Canada, June 14–17, 2022, in: Lecture Notes in Computer Science, vol. 13263, Springer, 2022, pp. 299–309.
[31] B. Amico, C. Combi, R. Rizzi, P. Sala, Discovering predictive dependencies on multi-temporal relations, in: A. Artikis, F. Bruse, L. Hunsberger (Eds.), 30th International Symposium on Temporal Representation and Reasoning, TIME 2023, September 25–26, 2023, NCSR Demokritos, Athens, Greece, in: LIPIcs, vol. 278, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 4.
[32] R. Moskovitch, F. Polubriaginof, A. Weiss, P. Ryan, N. Tatonetti, Procedure prediction from symbolic electronic health records via time intervals analytics, J. Biomed. Inform. 75 (2017) 70–82.
[33] R. Bellazzi, B. Zupan, Predictive data mining in clinical medicine: current issues and guidelines, Int. J. Med. Inform. 77 (2) (2008) 81–97.
[34] R. Bellazzi, C. Larizza, P. Magni, R. Bellazzi, Temporal data mining for the quality assessment of hemodialysis services, Artif. Intell. Med. 34 (1) (2005) 25–39.
[35] N. Itzhak, S. Jaroszewicz, R. Moskovitch, Event prediction by estimating continuously the completion of a single temporal pattern's instances, J. Biomed. Inform. 156 (2024) 104665, https://doi.org/10.1016/J.JBI.2024.104665.
[36] N.S.B. Ari, R. Moskovitch, Predictive temporal patterns discovery, Expert Syst. Appl. 226 (2023) 119974, https://doi.org/10.1016/J.ESWA.2023.119974.
[37] C.D. Francescomarino, I. Donadello, C. Ghidini, F.M. Maggi, W. Rizzi, S. Tessaris, Making sense of temporal event data: a framework for comparing techniques for the discovery of discriminative temporal patterns, in: G. Guizzardi, F.M. Santoro, H. Mouratidis, P. Soffer (Eds.), Advanced Information Systems Engineering - 36th International Conference, Proceedings, CAiSE 2024, Limassol, Cyprus, June 3–7, 2024, in: Lecture Notes in Computer Science, vol. 14663, Springer, 2024, pp. 423–439.
[38] C.S. Jensen, R.T. Snodgrass, Temporal homogeneity, in: L. Liu, M.T. Özsu (Eds.), Encyclopedia of Database Systems, second edition, Springer, 2018.
[39] A.E. Johnson, T.J. Pollard, L. Shen, H.L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, MIMIC-III, a freely accessible critical care database, Sci. Data 3 (1) (2016) 1–9, https://doi.org/10.1038/sdata.2016.35.
[40] C. Combi, A. Sabaini, Extraction, analysis, and visualization of temporal association rules from interval-based clinical data, in: Conference on Artificial Intelligence in Medicine in Europe, Springer, 2013, pp. 238–247.
[41] L. Sacchi, C. Larizza, C. Combi, R. Bellazzi, Data mining with temporal abstractions: learning rules from time series, Data Min. Knowl. Discov. 15 (2) (2007) 217–247.
[42] S. Concaro, L. Sacchi, C. Cerra, P. Fratino, R. Bellazzi, Mining healthcare data with temporal association rules: improvements and assessment for a practical use, in: Conference on Artificial Intelligence in Medicine in Europe, Springer, 2009, pp. 16–25.
[43] E. Winarko, J.F. Roddick, ARMADA - an algorithm for discovering richer relative temporal association rules from interval-based data, Data Knowl. Eng. 63 (1) (2007) 76–90, https://doi.org/10.1016/j.datak.2006.10.009.
[44] M. Mantovani, B. Amico, C. Combi, Discovering predictive trend-event patterns in temporal clinical data, in: Proceedings of the 36th Annual ACM Symposium on Applied Computing, 2021, pp. 570–579.

[45] M. Mantovani, C. Combi, M. Zeggiotti, Discovering and analyzing trend-event patterns on clinical data, in: 2019 IEEE International Conference on Healthcare Informatics (ICHI), IEEE, 2019, pp. 1–10.
[46] O.D. Harel, R. Moskovitch, Complete closed time intervals-related patterns mining, Proc. AAAI Conf. Artif. Intell. 35 (5) (2021) 4098–4105.
[47] C. Bettini, S. Jajodia, S. Wang, Time Granularities in Databases, Data Mining, and Temporal Reasoning, Springer Science & Business Media, 2000.
[48] C.S. Jensen, R.T. Snodgrass, M.D. Soo, Extending existing dependency theory to temporal databases, IEEE Trans. Knowl. Data Eng. 8 (4) (1996) 563–582, https://doi.org/10.1109/69.536250.
[49] V. Vianu, Dynamic functional dependencies and database aging, J. ACM 34 (1) (1987) 28–59, https://doi.org/10.1145/7531.7918.
[50] J. Wijsen, Design of temporal relational databases based on dynamic and temporal functional dependencies, in: J. Clifford, A. Tuzhilin (Eds.), Recent Advances in Temporal Databases, Workshops in Computing, Zürich, Switzerland, 17–18 September 1995, in: Proceedings of the International Workshop on Temporal Databases, Springer, 1995, pp. 61–76.
[51] J. Wijsen, Temporal fds on complex objects, ACM Trans. Database Syst. 24 (1) (1999) 127–176, https://doi.org/10.1145/310701.310715.
[52] C. Combi, A. Montanari, P. Sala, A uniform framework for temporal functional dependencies with multiple granularities, in: International Symposium on Spatial and Temporal Databases, Springer, 2011, pp. 404–421.
[53] A.A. Qahtan, N. Tang, M. Ouzzani, Y. Cao, M. Stonebraker, Pattern functional dependencies for data cleaning, Proc. VLDB Endow. 13 (5) (2020) 684–697, https://doi.org/10.14778/3377369.3377377, http://www.vldb.org/pvldb/vol13/p684-qahtan.pdf.
[54] Z. Abedjan, C.G. Akcora, M. Ouzzani, P. Papotti, M. Stonebraker, Temporal rules discovery for web data cleaning, Proc. VLDB Endow. 9 (4) (2015) 336–347, https://doi.org/10.14778/2856318.2856328.
[55] O. Kwon, J.M. Sim, Effects of data set features on the performances of classification algorithms, Expert Syst. Appl. 40 (5) (2013) 1847–1857, https://doi.org/10.1016/j.eswa.2012.09.017.
[56] S. Phdikar, J. Sil, A. Das, Feature selection by attribute clustering of infected rice plant images, Int. J. Mach. Intell. 3 (2) (2011) 74–88.
[57] M. Le Guilly, J.-M. Petit, V.-M. Scuturici, Evaluating classification feasibility using functional dependencies, in: Transactions on Large-Scale Data- and Knowledge-Centered Systems XLIV, Springer, 2020, pp. 132–159.
[58] Q. Lin, Y. Gu, J. Sai, J. Liu, K. Ren, L. Xiong, T. Wang, Y. Pang, S. Wang, F. Li, Eulerfd: an efficient double-cycle approximation of functional dependencies, in: 39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, CA, USA, April 3–7, 2023, IEEE, 2023, pp. 2878–2891.
[59] T. Bleifuß, T. Papenbrock, T. Bläsius, M. Schirneck, F. Naumann, Discovering functional dependencies through hitting set enumeration, Proc. ACM Manag. Data 2 (1) (2024) 43, https://doi.org/10.1145/3639298.
[60] X. Wan, X. Han, J. Wang, J. Li, Efficient discovery of functional dependencies on massive data, IEEE Trans. Knowl. Data Eng. 36 (1) (2024) 107–121, https://doi.org/10.1109/TKDE.2023.3288209.
[61] F. Pennerath, P. Mandros, J. Vreeken, Discovering approximate functional dependencies using smoothed mutual information, in: R. Gupta, Y. Liu, J. Tang, B.A. Prakash (Eds.), KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23–27, 2020, ACM, 2020, pp. 1254–1264.
[62] M. Alatar, A. Sali, Approximate keys and functional dependencies in incomplete databases with limited domains, in: I. Varzinczak (Ed.), Foundations of Information and Knowledge Systems - 12th International Symposium, Proceedings, FoIKS 2022, Helsinki, Finland, June 20–23, 2022, in: Lecture Notes in Computer Science, vol. 13388, Springer, 2022, pp. 147–167.
[63] M. Parciak, S. Weytjens, N. Hens, F. Neven, L.M. Peeters, S. Vansummeren, Measuring approximate functional dependencies: a comparative study, CoRR, arXiv:2312.06296 [abs], 2023, arXiv:2312.06296, https://doi.org/10.48550/ARXIV.2312.06296.
[64] S. Uchino, R. Bellomo, D. Goldsmith, S. Bates, C. Ronco, An assessment of the rifle criteria for acute renal failure in hospitalized patients, Crit. Care Med. 34 (7) (2006) 1913–1917, https://doi.org/10.1097/01.CCM.0000224227.70642.4F.
[65] R.W. Schrier, W. Wang, B. Poole, A. Mitra, et al., Acute renal failure: definitions, diagnosis, pathogenesis, and therapy, J. Clin. Invest. 114 (1) (2004) 5–14, https://doi.org/10.1172/JCI22353.
[66] A. Khwaja, Kdigo clinical practice guidelines for acute kidney injury, Nephron, Clin. Pract. 120 (4) (2012) c179–c184.
[67] C.S. Jensen, R.T. Snodgrass, Valid time, in: L. Liu, M.T. Özsu (Eds.), Encyclopedia of Database Systems, second edition, Springer, 2018, pp. 4359–4360.
[68] C.S. Jensen, R.T. Snodgrass, Temporal data models, in: L. Liu, M.T. Özsu (Eds.), Encyclopedia of Database Systems, second edition, Springer, 2018.
[69] A.R.M. Forkan, I. Khalil, A clinical decision-making mechanism for context-aware and patient-specific remote monitoring systems using the correlations of multiple vital signs, Comput. Methods Programs Biomed. 139 (2017) 1–16, https://doi.org/10.1016/j.cmpb.2016.10.018.
[70] P. Pirasteh, S. Nowaczyk, S. Pashami, M. Löwenadler, K. Thunberg, H. Ydreskog, P. 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, 2019, pp. 1–10.
[71] Y. Huhtala, J. Kärkkäinen, P. Porkka, H. Toivonen, Tane: an efficient algorithm for discovering functional and approximate dependencies, Comput. J. 42 (2) (1999) 100–111, https://doi.org/10.1093/comjnl/42.2.100.
[72] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423.
[73] D.J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.
[74] P. Smyth, R.M. Goodman, An information theoretic approach to rule induction from databases, IEEE Trans. Knowl. Data Eng. 4 (4) (1992) 301–316.
[75] G.B. Moody, R.G. Mark, A.L. Goldberger, Physionet: physiologic signals, time series and related open source software for basic, clinical, and applied research, in: 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2011, Boston, MA, USA, August 30 – Sept. 3, 2011, IEEE, 2011, pp. 8327–8330.
[76] B. Amico, C. Combi, G. Gambaro, Discovering predictive temporal patterns for acute kidney injury from critical care data, AMIA Annual Symp. Proc. 2023 (2023) 261–269.
[77] Z. Xu, J. Chou, X.S. Zhang, Y. Luo, T. Isakova, P. Adekkanattu, J.S. Ancker, G. Jiang, R.C. Kiefer, J.A. Pacheco, et al., Identifying sub-phenotypes of acute kidney injury using structured and unstructured electronic health record data with memory networks, J. Biomed. Inform. 102 (2020) 103361, https://doi.org/10.1016/j.jbi.2019.103361.